Overview of Recommendation Techniques in Business Process Modeling*

Krzysztof Kluza, Mateusz Baran, Szymon Bobek, Grzegorz J. Nalepa

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

Abstract Modeling business processes is an important issue in Business Process Management. As model repositories often contain similar or related models, they should be used when modeling new processes. The goal of this paper is to provide an overview of recommendation possibilities for business process models. We introduce a categorization and give examples of recommendation approaches. For these approaches, we present several machine learning methods which can be used for recommending features of business process models.

1 Introduction

Business Process (BP) models are visual representations of processes in an organization. Such models can help to manage process complexity and are also easy to understand for non-business user. Although there are many new tools and methodologies which support process modeling, especially using Business Process Model and Notation (BPMN) [1], they do not support recommendation mechanisms for BP modelers.

As BPMN specifies only a notation, there can be several ways of using it. There are style directions how to model BPs [2], or guidelines for analysts based on BPs understandability (e.g. [3]). However, a proper business process modeling is still a challenging task, especially for inexperienced users.

Recommendation methods in BP modeling can address this problem. Based on current progress or additional pieces of information, various features can be recommended to a modeler, and he/she can be assisted during designing models. Such assistance can provide autocompleting mechanisms with capabilities of choosing next process fragments from suggested ones. Names of model elements or attachments can be recommended as well. Such approaches can reduce number of errors during process design as well as speed up modeling process. It also supports reusing of existing process models, especially when a process repository is provided.

The rest of this paper is organized as follows: In Section 2, we provide a categorization of recommendation methods used in business process modeling. Section 3 describes the current state of the art in this research area. Selected machine learning methods that can be used for recommending features of process models are presented in Section 4. Section 5 presents an example which can be considered as a suitable case study for recommendation purposes. The paper is summarized in Section 6.

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

2 Types of recommendations

Basically, recommendation methods in BPs modeling can be classified as one of two types: *subject-based* and *position-based* classification. The first one concentrates on what is actually suggested, while the second one focuses on the place where the suggestion is to be placed. However, they are suited for different purposes and therefore are complementary. A hierarchy of the identified types of recommendation methods is presented in Figure 1.

2.1 Subject-based classification

In subject-based classification we focus on what is actually suggested. The suggestion itself is not directly dependent on the context it is placed in. The recommendation algorithms may actually inspect the context to be able to deliver more accurate results but it is not an inherent feature of recommended item.

1. **Attachment recommendations** – as the name suggests, these recommendations suggest how to link a business process (or, more precisely, a selected element of it) with an external entity like a decision table or another process. Attachment recommendations appear naturally where user should link two already existing items.
   (a) **Decision tables** – recommendations for a decision table describing conditions in a gate. See an example in Figure 2.

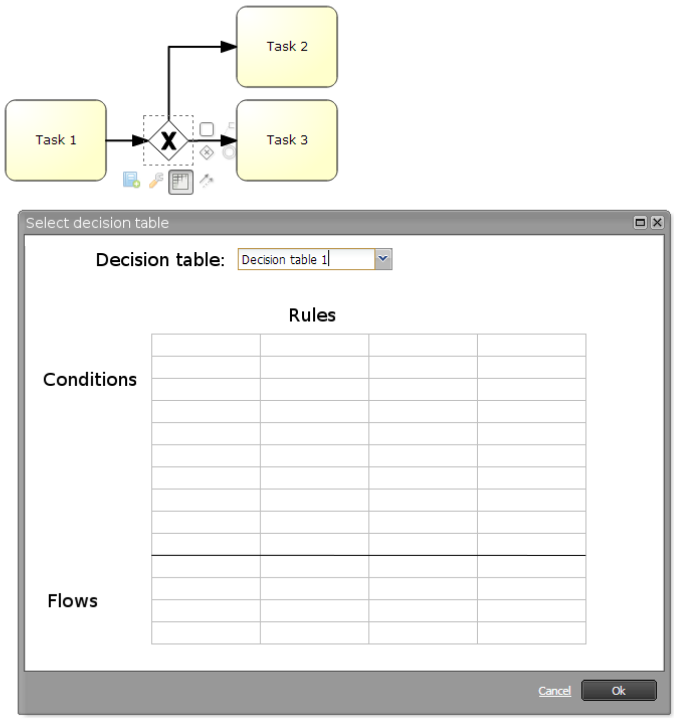

Figure 2. Decision table suggestion

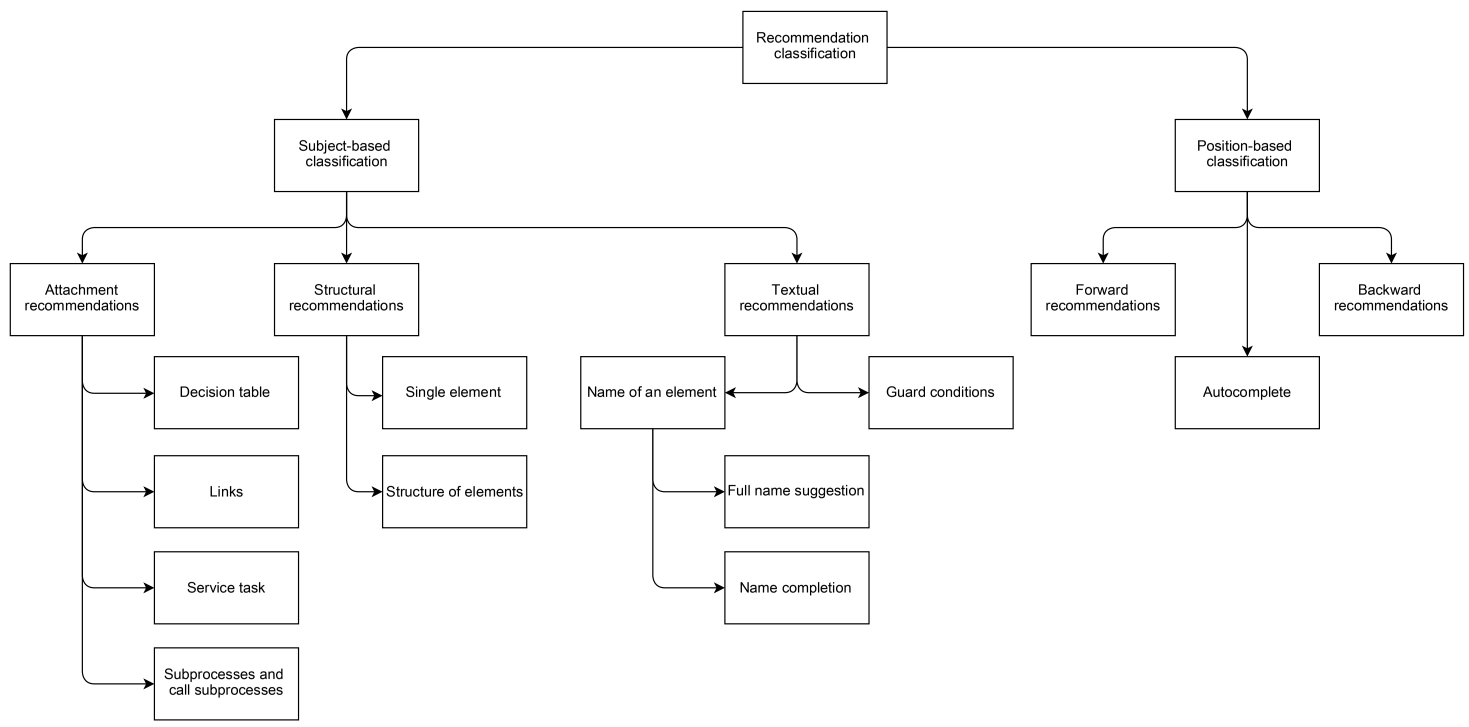

Figure 1. Identified types of recommendations

(b) **Links** – recommendations for a catching event that should be connected with the selected throwing Intermediate Link Event. See an example on Figure 3.

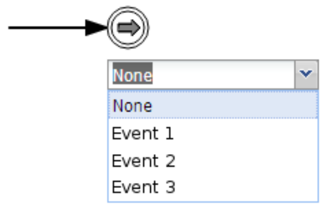

Figure 3. Throwing Intermediate Link Event suggestion

(c) **Service task** – recommendation for a service task performed in the given task item. See an example in Figure 4.

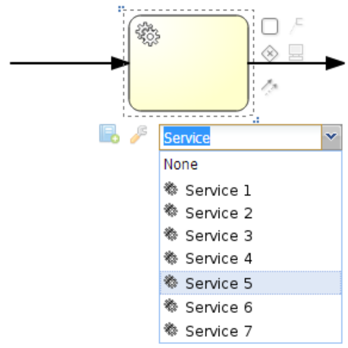

Figure 4. Service task selection with recommendation

(d) **Subprocess and call subprocess** – recommendation for a subprocess or call subprocess that should be linked with the given activity (see Figure 5).

2. **Structural recommendations** – a new part of the diagram is suggested. One or more elements with, for example, missing incoming or outgoing flows are selected. The suggested structure is connected with old chosen elements.
   (a) **Single element** – a single item (activity, gate, swimlane, artifact, data object or event) is suggested. This is a more straightforward extension of editors like Oryx/Signavio that can already insert single elements quite easily.
   (b) **Structure of elements** – two or more items are suggested. A more sophisticated solution where an entire part of the process is inserted into existing, unfinished structure.
3. **Textual recommendations** are suggestions of names of elements or guard conditions. Either the full text can be suggested or suggestions may show while the text is being typed.
   (a) **Name of an element** – a name of activity, swimlane or event may be suggested.
      i. **Name completion** happens when user is typing the name. Several possible completions of partially entered name are suggested to the user.

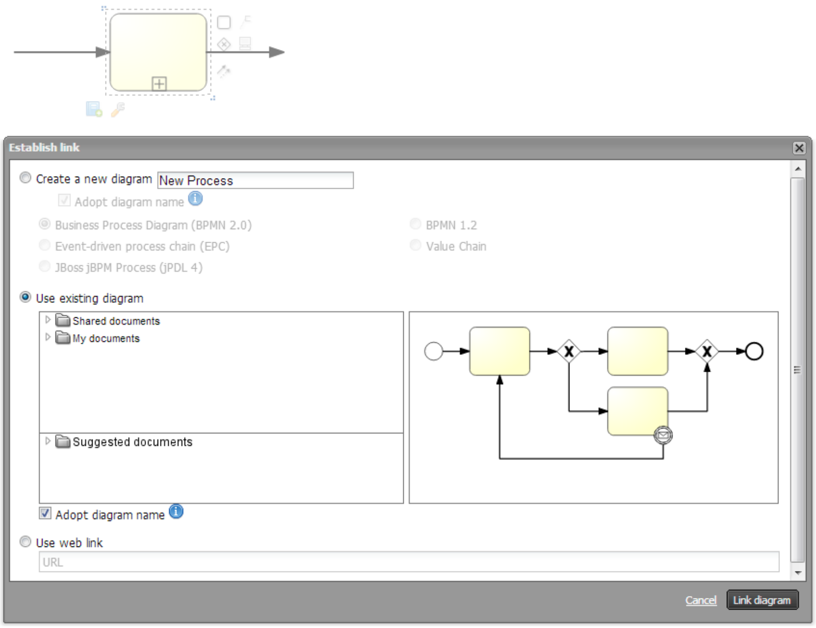

Figure 5. Subprocess selection with recommendation

ii. **Full name suggestion** happens when the user wants the name to be suggested by the system based on the context in which the element is placed.

(b) **Guard condition suggestions** are different from name suggestions because more than one text (condition) may be suggested at once and these conditions must satisfy the requirements of the gateway. The latter requirement implies that semantic analysis of conditions is necessary to give meaningful suggestions. See example in Figure 6.

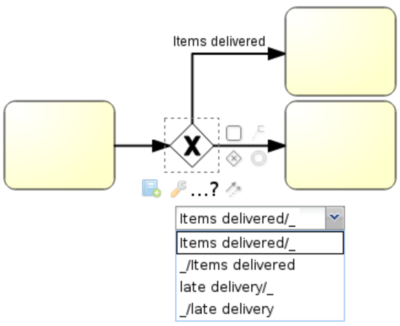

Figure 6. Guard condition suggestion

2.2 Position-based classification

1. **Forward completion** – a part of the process is known and the rest of the process, starting with one selected activity, is to be suggested. See Figure 7.

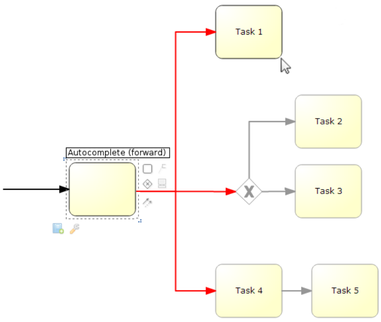

Figure 7. Forward completion

2. **Backward completion** – a part of the process is known and the rest of the process, ending with one selected activity, is to be suggested. See Figure 8.

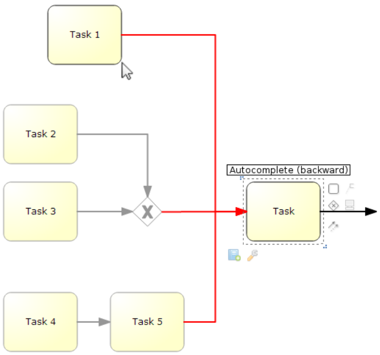

Figure 8. Backward completion

3. **Autocomplete** – a part of the process is known and the rest of the process is to be suggested. A number of items with no outgoing or incoming flows is selected – missing flows will lead to or from the suggested structure. See Figure 9.

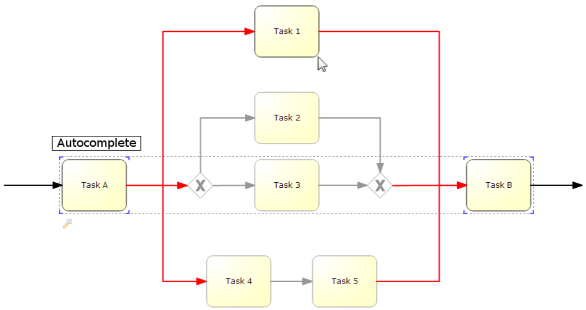

Figure 9. Autocompletion

3 Recommendation Techniques for Business Process Models

Empirical studies have proven that modelers prefered to receive and use recommendation suggestions during design [4]. Recommendations can be based on many factors, including labels of elements, current progress of modeling process, or some additional pieces of information, such as process description. There are several existing approaches which can be assigned to the following subject-based categories:

1. **Attachment recommendations**: Born et al. [5] presented an approach that supports modelers during modeling tasks by finding appropriate services, meaningful to the modeler. More complex approach which helps process designers facilitate modeling by providing them a list of related services to the current designed model was proposed by Nguyen et al. [6]. They capture the requested service's composition context specified by the process fragment and recommend the services that best match the given context. The authors also described an architecture of a recommender system which bases on historical usage data for web service discovery [7].
2. **Structural recommendations**: Mazanek et al. [8] proposed a syntax-based assistance in diagram editor which takes advantage of graph grammars for process models. Based on this research they proposed also a sketch-based diagram editor with user assistance based on graph transformation and graph drawing techniques [9].

   Hornung et al. [10] presented the idea of interpreting process descriptions as tags and based on them provide a search interface to process models stored in a repository. Koschmider and Oberweis extended this idea in [11] and presented their recommendation-based editor for business process modeling in [4]. The editor assists users by providing search functionality via a query interface for business process models or process model parts and using automatic tagging mechanism in order to unveil the modeling intention of a user at process modeling time. An approach proposed by Wieloch et al. [12] delivers a list of suggestions for possible successor tasks or process fragments based on analysis of context and annotations of process tasks. Case based reasoning for workflow adaptation was discussed in [13]. It allows for structural adaptations of workflow instances at build time or at run time. The approach supports the designer in performing such adaptations by an automated method based on the adaptation episodes from the past. The recorded changes can be automatically transferred to a new workflow that is in a similar situation of change.
3. **Textual recommendations**: Naming strategies for individual model fragments and whole process models was investigated in [14] They proposed an automatic naming approach that builds on the linguistic analysis of process models from industry. This allows for refactoring of activity labels in business process models [15].

   According to Kopp et al. [16] it is not to automatically deduct concrete conditions on the sequence flows going out from the new root activity as we cannot guess the intention of the fragment designer. However, they presented how a single BPMN fragment can be completed to a BPMN process using autocompletion of model fragments, where the types of the joins are AND, OR, and XOR.

4 Machine Learning Approach for Recommendation

The idea of recommender systems was evolving along with a rapid evolution of the Internet in mid-nineties. Methods such as collaborative filtering, content-based and knowledge-based recommendation [17] gained huge popularity in the area of web services [18] and recently most often in context-aware systems [19]. The principal rule that most of the recommendation methods are based on, exploits an idea of similarities measures. This measures can be easily applied to items that features can be extracted (eg. book genre, price, author) and ranked according to some metrics (customer liked the book or not). However, when applied to BPMN diagrams, common recommender systems face a big problem of non existence of standard metrics that will allow for comparison of models. What is more, feature extraction of the BPMN diagrams that will allow for precise and unambiguous description of models is very challenging and, to our knowledge, still unsolved issue.

Therefore, other machine learning methods should be investigated according to an objective aiming at providing recommendation mechanisms for a designer. The following Section contains an analysis of possible application of machine learning methods to recommendations described in Section 2. A comprehensive summary is also provided in Table 1. The black circle denotes full support of particular machine learning method to recommendation; half-circle denoted partial support of particular machine learning method to recommendation, and empty circle means no, or very limited support.

	Clustering algorithms[a]	**Decision trees**[b]	**Bayesian networks**[c]	**Markov chains**
Attachment recommendations	❍	●	❍	❍
Structural recommendations	❍	●	●	●
Textual recommendations	❍	❍	◗	●
Position based classification	❍	●	●	●

Table 1. Comparison of different machine learning methods for recommending features denoted in Section 2

[a] Useless as an individual recommendation mechanism, but can boost recommendation when combined with other methods

[b] No cycles in diagram

[c] No cycles in diagram

4.1 Classification

Clustering methods Clustering methods [20] are based on optimization task that can be described as an minimization of a cost function that are given by the equation 1. K denotes number of clusters that the data set should be divided into.

$$\sum_{n=1}^{N} \sum_{k=1}^{K} \|X_n - \mu_n\|^2 \tag{1}$$

This cost function assume existence of a function f that allows for mapping element's features into an M dimensional space of $X \in R^m$. This however requires developing methods for feature extraction from BPMN diagrams, which is not trivial and still unsolved task. Nevertheless, clustering methods can not be used directly for recommendation, but can be very useful with combination with other methods.

Decision trees Decision trees [21] provide a powerful classification tool that exploits the tree data structure to represent data. The most common approach for building a tree, assumes possibility of calculation entropy (or based on it, so-called *information gain*) that is given by the equation 2.

$$E(X) = -\sum_{i=1}^{n} p(x_i)log(p(x_i))) \tag{2}$$

To calculate the entropy, and thus to build a decision tree, only a probability p of presence of some features in a given element is required. For the BPMN diagram, those features could be diagram nodes (gateways, tasks, etc) represented by a distinct real numbers. Having a great number of learning examples (diagrams previously build by the user), it is possible to build a tree that can be used for predicting next possible element in the BPMN diagram. However, the nature of the tree structure requires from BPMN diagram to not have cycles, which not always can be guaranteed.

4.2 Probabilistic Graphical Models

Probabilistic Graphical Models use a graph-based representation as the basis for compactly encoding a complex probability distribution over a high dimensional space [22]. The most important advantage of probabilistic graphical models over methods described in Section 4.1 is that it is possible to directly exploit the graphical representation of BP diagrams, which can be almost immediately translated into such model.

Bayesian networks Bayesian network (BN) [23] is an acyclic graph that represents dependencies between random variables, and provide graphical representation of the probabilistic model. The example of a Bayesian network is presented in Figure 10.

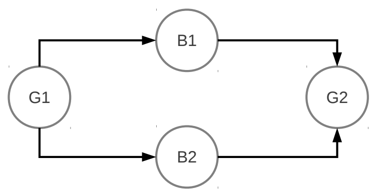

Figure 10. Bayesian network

The advantage of BN is that the output of a recommendation is a set of probabilities, allowing for ranking the suggestion from the most probable to the least probable. For example to calculate the probability of the value of the random variable *B1* from the Figure 10, the equation 3 can be used. The G1,2 can be denoted as BPMN gateways, and B1,2 as other blocks, e.g. Tasks or Events. Thus, having any of these blocks given, we can calculate a probability of a particular block being a missing part.

$$P(B1) = \sum_{G1}\sum_{G2}\sum_{B2} P(G1)P(B1|G1)P(B2|G1)P(G2|B1, B2) \tag{3}$$

This method however, will not be efficient for large diagrams, since exact inference in Bayesian networks is NP-hard problem. To solve this problem either the small chunks of BPMN diagram can be selected for the inference, or approximate inference applied.

Markov Chains Markov chain [24] is defined in terms of graph of state space $Val(X)$ and a transition model τ that defines, for every state $x \in Val(X)$ a next-state distribution over $Val(X)$. These models are widely used for text auto-completion and text correction, but can be easily extended to cope with other problems such as Structural recommendations, or position-based classification.

We can assume different BPMN block types as states x , and connections between them as a transition model τ.

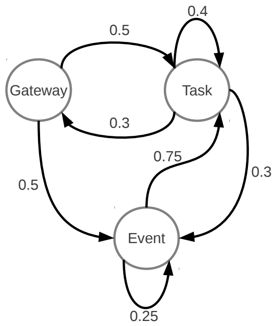

Figure 11. Markov Chain

An example Markov chain model is presented in Figure 11. The number above the arrows denotes transition probability from one state to another. The characteristic of the Markov chains allows for cycles.

5 Case Study

The presented recommendation approaches can be applied to process models, especially modeled on the basis of the existing processes in a model repository. For the purpose of evaluation, we prepared 3 different BPMN models of bug tracking systems (Django and JIRA) and the model of the issue tracking approach in VersionOne. A bug tracking system is a software application that helps in tracking and documenting the reported software bugs (or other software issues in a more general case). Such a system is often integrated with other software project management applications.

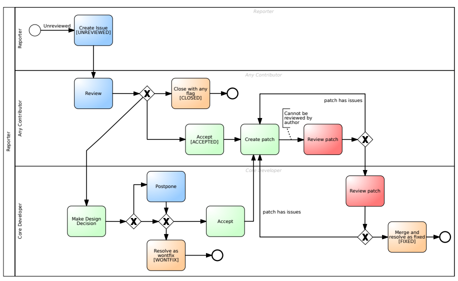

Figure 12. Django

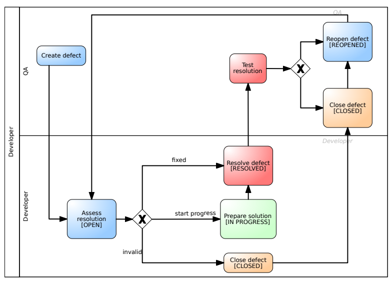

Figure 13. Jira

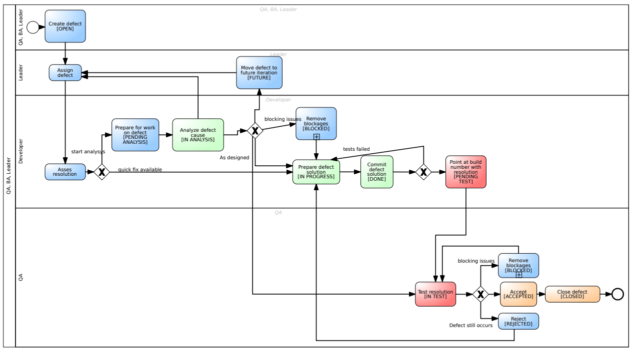

Figure 14. VersionOne

We selected such models as a case study because of their similarity. As the processes of different bug trackers present the existing variability, such example can be easily used to present for recommendation purposes when modeling a new bug tracking flow for a bucktracking system.

6 Conclusion and future work

This paper focuses on a problem of recommendation methods in BP modeling. Such methods help in speeding up modeling process and producing less error prone models than modeling from scratch. The original contribution of the paper is introducing a categorization of recommendation approaches in BP modeling and short overview of machine learning methods corresponding to the presented recommendations.

Our future work will focus on specifying recommendation approach for company management systems in order to enhance modeling process and evaluation of the selected recommendation methods. We plan to carry out a set of experiments aiming at testing recommendation approaches on various model sets.

References

1. OMG: Business Process Model and Notation (BPMN): Version 2.0 specification. Technical Report formal/2011-01-03, Object Management Group (2011)
2. Silver, B.: BPMN Method and Style. Cody-Cassidy Press (2009)
3. Mendling, J., Reijers, H.A., van der Aalst, W.M.P.: Seven process modeling guidelines (7pmg). Information & Software Technology **52** (2010) 127–136
4. Koschmider, A., Hornung, T., Oberweis, A.: Recommendation-based editor for business process modeling. Data & Knowledge Engineering **70** (2011) 483 – 503
5. 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
6. 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

7. 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
8. 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
9. 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)
10. 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
11. 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
12. 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
13. 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
14. 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
15. Leopold, H., Smirnov, S., Mendling, J.: On the refactoring of activity labels in business process models. Information Systems **37** (2012) 443–459
16. 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
17. Jannach, D., Zanker, M., Felfernig, A., Friedrich, G.: Recommender Systems An Introduction. Cambridge University Press (2011)
18. Wei, K., Huang, J., Fu, S.: A survey of e-commerce recommender systems. In: Service Systems and Service Management, 2007 International Conference on. (2007) 1–5
19. Bobek, S., Nalepa, G.J.: Overview of context-aware reasoning solutions for mobile devices. proposal of a rule-based approach. Computer Science and Information Systems (2014) accepted.
20. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)
21. Mitchell, T.M.: Machine Learning. MIT Press and The McGraw-Hill companies, Inc. (1997)
22. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. MIT Press (2009)
23. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Machine Learning **29** (1997) 131–163
24. Rabiner, L., Juang, B.H.: An introduction to hidden markov models. ASSP Magazine, IEEE **3** (1986) 4–16