Automated Process Decision Making Based on Integrated Source Data

Florian Niedermann, Bernhard Maier, Sylvia Radeschütz,
Holger Schwarz, and Bernhard Mitschang

Institute of Parallel and Distributed Systems, University of Stuttgart,
Universitätsstraße 38, 70569 Stuttgart, Germany
{florian.niedermann,bernhard.maier,sylvia.radeschuetz,holger.schwarz,
bernhard.mitschang}@ipvs.uni-stuttgart.de
http://www.ipvs.uni-stuttgart.de

Abstract. Decision activities are frequently responsible for a major part of a process's duration and resource consumption. The automation of these activities hence holds the promise of significant cost and time savings, however, only if the decision quality does not suffer. To achieve this, it is required to consider data from diverse sources that go beyond the process audit log, which is why approaches relying solely on it are likely to yield sub-optimal results. We therefore present in this paper an approach to process decision automation that incorporates data integration techniques, enabling significant improvements in decision quality.

Keywords: Data Mining, Decision Automation, Data Integration, Business Process Management, Data-driven Processes.

1 Introduction

In this section, we first discuss the role and complexities of decisions in business processes. Then, we demonstrate the importance of decision automation by considering the various effects that automation can have on a business. Finally, we briefly introduce our *deep Business Automation Platform (dBOP)* that provides an integrated environment for *Business Process Optimization (BPO)*, including the automation of decisions.

1.1 Decisions and Influence Factors in Business Processes

A decision fundamentally consists of a set of one or several nodes in a business process that determines, in the presence of several alternatives, which process path to take. A typical (manual or non-automated) is made by a human actor who weighs several factors against each other to arrive at the decision results. Some of these factors include:

- **Process data**: As most processes are today executed using some kind of *Business Process Management System (BPMS)* that supports handing data over between different activities, one decision factor is the process data. This is also the data that is typically written into the audit log of the *BPMS*.

W. Abramowicz (Ed.): BIS 2011, LNBIP 87, pp. 160–171, 2011.

- **Application systems**: In many processes, the decision maker utilizes one or several application systems - either to gain additional information or for decision support.
- **External services**: In some instances, data provided by an external source can be instrumental to the decision - a popular example being the role of credit rating agencies in many retail processes.
- **Decision maker attributes**: Whenever the decision maker does not follow strict formal rules, her or his characteristics, experiences and other attributes influence the decision. These attributes can be either explicit (e.g., formal education, years of experience) or implicit (e.g., cultural values, biases etc.).
- **Other implicit and explicit knowledge**: A wide range of other factors can influence decisions. This can include non-codified knowledge about the work item, the business process or external influences. For instance, major events or a casual conversation with a supplier can have substantial impact on a decision.

1.2 Motivation for Decision Automation

Depending on the nature of the process, decisions and their preparation can easily account for a significant proportion of a processes' duration, cost and resource requirements. Hence, there are strong reasons for business to automate decisions as much as feasible:

- **Faster processes**: While a human actor often can take minutes to decide upon a complex matter, a decision model can arrive at a conclusion within the fraction of a second.
- **Dealing with resource bottlenecks**: If a decision involves human actors, an organization invariably needs to have a sufficient number of people with the according qualifications. This is especially challenging if the demand is fluctuating, as this either leads to idle workers or to long waiting times.
- **Reducing process cost**: As a manual decision involves a (costly) human actor, its cost is typically reduced through automation.
- **Enabling new business models**: Through the combination of the factors mentioned above, automated decisions enable new business models that would not be economical or even possible with a manual decision maker.

Despite this importance, the multitude of influence factors discussed in the previous section suggests that it is nearly impossible to create a decision model that exactly replicates the decision of a human actor in any given situation. This is frequently, however, neither necessary nor desired (e.g., human bias can negatively influence decision quality). Further, research in machine learning has shown that the behavior of complex systems (such as decisions, see Section 3.2) can in various scenarios be reasonably well predicted using a sufficiently large subset of the system's attributes [4], such as the one that can be provided by integrating process and operational data.

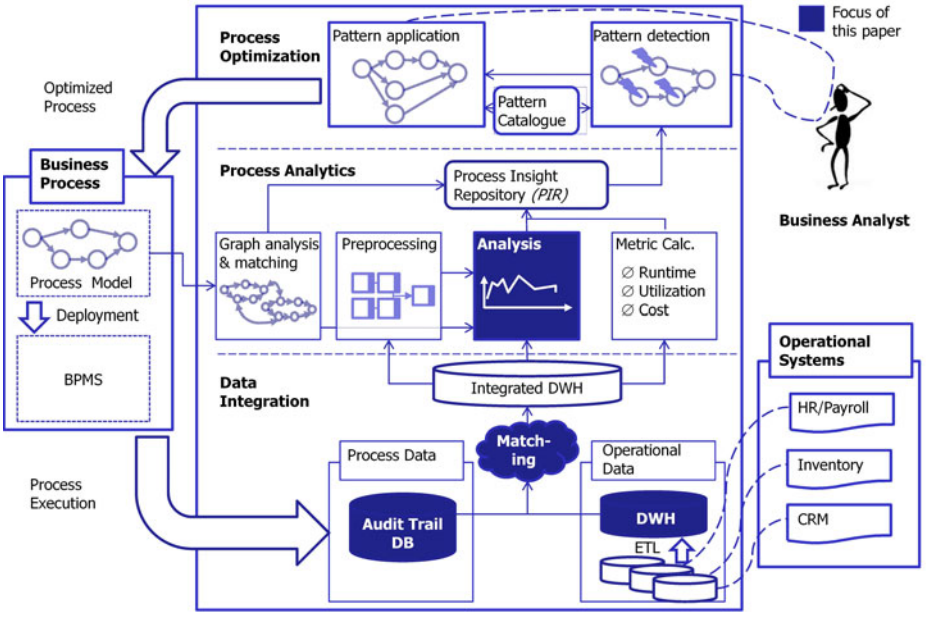

Fig. 1. Deep Business Optimization Platform overview

1.3 The Deep Business Optimization Platform

In order to make decision automation viable, we need a platform that provides the required data, the appropriate analysis techniques as well as a means to use the decision model in process execution. Our *deep Business Automation Platform (dBOP)* shown in Fig.1 provides these and other facilities for (semi-)automated process optimization spread over three architectural layers:

1. **Data Integration**: As data relevant to the process and its decisions can be spread over a variety of heterogeneous data sources, the first platform layer provides the means to match and integrate process data with other data sources. We discuss some of the aspects of this layer in Section 2.
2. **Process Analytics**: In order to achieve meaningful optimization results, process specific "insights" need to be extracted from the integrated data layer. One of these analysis techniques is the learning of a decision classifier that is the subject of Section 3. This layer also includes process matching capabilities for design-time optimization [10] and static process graph analysis methods.
3. **Process Optimization**: Next to customized analysis techniques, the platform also contains a broad set of formalized best practice process optimization patterns. These patterns (such as parallelization, task elimination or decision automation) utilize the analysis results to determine which modification of the process are most beneficial under a certain goal function given

by the process analyst. We only briefly visit this layer in Section 3.3, more details can be found in [11].

As our previous work (see for instance [13], [12] or [9]) has discussed general aspects of the platform extensively, this paper focuses on the specific components required for decision automation. In Section 2, we show how our integration approach enables us to cover a large subset of the decision influence factors listed in Section 1.1. Next, Section 3 shows how the integrated data is used to build and provide an automated decision classifier. In Section 4, we evaluate our approach using the decision classifier implemented in the *dBOP* and demonstrate how integrated data can improve the classifier quality. Finally, we take a look at related work in Section 5 before providing a brief discussion of our future research plans on the subject and the conclusion of the paper in Section 6.

2 Data Integration

In this section, we illustrate the Data Integration capabilities of the *dBOP*. First, we discuss the properties of process data vs. other operational data sources. Then, we explain how our matcher helps to integrate these heterogeneous data sources and how they are consolidated to provide the input for the decision classifier.

2.1 Process and Operational Data

To achieve integration, process activities pass data between each other. As most processes are today executed on some kind of *BPMS*, this process data is recorded in their audit log. The nature of process data is *flow-oriented*, i.e., while it contains information about the process flow, activity durations etc., the amount of information about the process subjects (e.g., work items, customers, resources) is often reduced to a minimum.

Operational data, stored in some application system or a a *Data Warehouse (DWH)*, on the other hand is *subject-oriented*. It contains comprehensive information about the process subjects, but little information about the process flow (e.g., it might not contain information about process paths or cancelled instances).

The relationship between process and operational data and the decision influence factors is conceptually illustrated in Fig.2. As we can see, both process and operational data cover different areas. Further, three observations can be made: First, even by combining process and operational data, we are unlikely to capture all relevant information. This is, however, quite often not necessary, as we have discussed in Section 1.2. Second, process and operational data have some overlapping attributes (such as ID values). These attributes can be used to match and integrate the data, as we show in the next section. Finally, the information gain of operational data varies with the richness of the subject oriented information contained in the process data (e.g., because it is newly captured during the process). The last observation is revisited and quantitatively evaluated in Section 4.

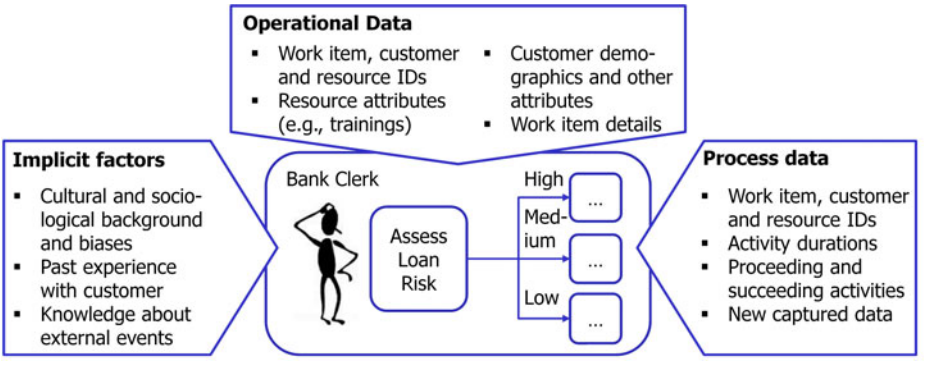

Fig. 2. DecisionAttributes

2.2 Integration Approach

As prior sections have shown, decision automation requires the integration of heterogeneous data sources. Due to the different paradigms of process and operational data, classical schema matching approaches like [1] struggle with their integration (e.g., because they have no specific methods to propagate matchings). This is why we have developed a specific approach for integrating operational and process data. As Fig.3 illustrates, it consists of three steps:

1. **Annotation and matching:** First, the matches between the process and operational data models need to be determined. This can be done using a variety of techniques, including direct matches pointed out by an analyst, matches found out using a semantic reasoner or through natural language processing (e.g., by matching synonyms).
2. **Matching propagation:** After the initial matches have been determined, the matches are propagated based on a set of matching rules that utilize the specific properties of process data. One of these rules, for instance,

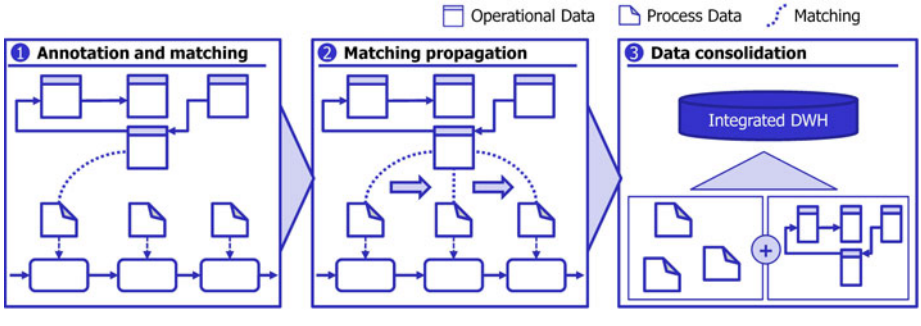

Fig. 3. *dBOP* Data Integration Approach

propagates the matchings along data flow mappings (such as the one realized by `<assign>` statements in BPEL).
3. **Data consolidation:** Finally, the data is consolidated into an integrated DWH, similar to the one discussed in [2]. This DWH is then used as the data source, e.g., for the construction of the decision classifier.

More details on the integration approach can be found in [13] and [12].

3 Decision Automation

After the previous section has discussed how we can get a large share of the decision influence factors through the use of data integration, this section will show how this data can be used to build a concrete decision classifier. First, we will show how decisions can be identified in a graph-based process model. Then, we will discuss how these decisions can be transformed into a classification problem. Finally, we show how the classifier is used by the *dBOP* to implement decision automation during process modeling and execution.

3.1 Identifying Decisions

To explain how decisions are identified, we first need some meta-model to represent processes. For the scope of this paper, we use a greatly simplified version of the process model graph presented in [7], please see there for further details. We use this meta-model, as it presents a good mix of usability (due to its proximity to prominent modeling languages, such as BPMN), implementability and formal reasoning powers. The concepts presented can, of course, also be applied with minor modification to other meta-models like Petri nets.

Definition. *Simplified PM Graph: A simplified PM Graph G is a tuple $(V, O, N, C, E, \iota, o)$, μ and χ are functions, in which*

1. *V is the finite set of process data elements (also called variables).*
2. *O is the finite set of operational data relevant to the process.*
3. *N is the finite set of process nodes.*
4. *C is the finite set of conditions.*
5. *$E \subseteq N \cup N \cup C$ is the set of (control) connectors.*
6. *$\iota : N \cup C \cup \{G\} \rightarrow \wp(V)$ is the input data map, with $\wp(V)$ being the power set over V.*
7. *$o : N \cup \{G\} \rightarrow \wp(V)$ is the output data map.*
8. *$\mu : \wp(V) \rightarrow \wp(O)$ performs the matching, i.e., integrates operational data with variables.*
9. *$\chi : E \rightarrow C \cup \{\emptyset\}$ determines, which condition has been assigned to a control connector.*

Let further be $\vec{N}$: $N \rightarrow \wp(N)$ denote the immediate successor nodes and $\vec{E}$: $N \rightarrow \wp(E)$ the immediate successor control connectors of a process node

and let the predicates $SUCC(n_1, n_2)$ and $PRED(n_1, n_2)$ denote that n_2 is the successor/predecessor of n_1 respectively.

Based on this definition, we now define a decision as a node with several conditional outgoing connectors and exactly one outgoing connector without a condition (the *default* connector). Further, we define the set of associated nodes as the nodes that provide the input for the outgoing control connectors. More precisely, we define a decision as follows:

Definition. *Decision: A Decision D in a simplified PM Graph G is a tuple (n_D, V_E, A, R) in which*

1. *n_D is the decision node for which holds true: $|\vec{E}(n_D)| \geq 2$ and $\exists e_{Default} \in \vec{E}(n_D) : \chi(e_{Default}) = \{\emptyset\} \Rightarrow \forall e \in \vec{E} \setminus \{e_{Default}\} : \chi(e) \neq \{\emptyset\}$.*
2. *V_E is the set of data attributes used to select a path with $V_E = \bigcup_{\forall e \in \vec{E}(n_D)} \iota(\chi(e))$.*
3. *A is the minimal set of associated nodes, with $\bigcup_{\forall a \in A} o(a) \supseteq V_E$ and $\forall a \in A : \exists v \in V_E : v \in o(a) \wedge \neg\exists n \in N : v \in o(n) \wedge SUCC(a, n) \wedge PRED(n, n_D)$*
4. *R is the set of possible decision results, with $R = \vec{N}(n_D)$.*

For the sake of simplicity, we will focus on decisions, where $A = \{n_D\}$, i.e., the only node that is relevant for the decision is the decision node itself. While the principle method for automating decisions with multiple associated nodes is the same, there are some added complexities (such as resequencing of activities or effects of the automation on the process context) whose discussion goes beyond the scope of this paper.

3.2 Decision Automation as a Classification Problem

In machine learning, a classification problem typically has three components [4]: A set of class labels that should be "learned", a set of input data to use for the learning and a classification algorithm that processes the data to learn the classification rules. Using the definition of a decision introduced in the previous section, we can define a decision classifier as follows:

Definition. *Decision classifier: A decision classifier D_{CL} for a decision D is a tuple (D, I, L, Alg) in which*

1. *D is the decision the classifier seeks to learn.*
2. *I is the set of classification input data. The input data is made up by the matched input data of all associated activities, i.e., $I = \bigcup_{\forall a \in A} \mu(\iota(a))$.*
3. *L is the set of class labels to be learned from the input data, which is made up by the different decision results, i.e., $L = R$.*
4. *Alg is some classification algorithm that is used to learn the decision (e.g., a decision tree or a multilayer perceptron).*

Decision automation can therefore be successfully transformed into a "standard" classification problem.

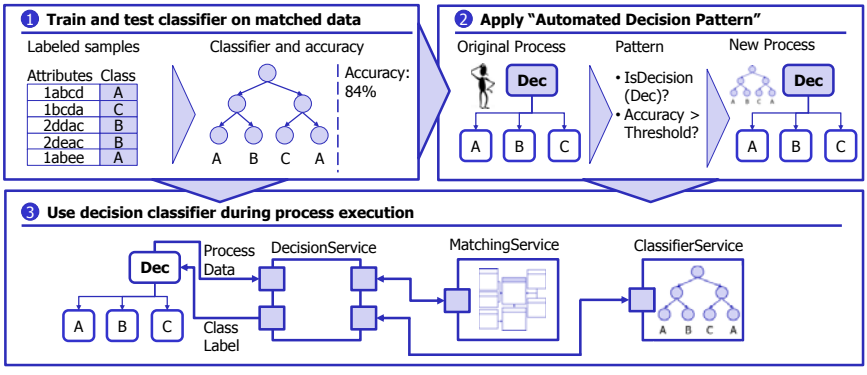

Fig. 4. *dBOP* Decision Automation Implementation

3.3 Implementation

As we have shown that decision automation can be treated like a classification problem, we can now use one of the many established classification algorithms to solve this challenge. The *dBOP* offers for this purpose a broad spectrum of classifiers which are taken from the WEKA library [5], with the default recommendation being the use of classification trees (see Section 4 for details on this choice).

As Fig.4 shows, the implementation of decision automation within the *dBOP* consists of three steps. First, the classifier for each decision in the process is trained and tested. The classifier along with the testing results are then forwarded to the *dBOP* optimizer. For these classifiers that the optimizer deems (based on user preferences) to be sufficiently good, it employs the "Automated Decision" process optimization pattern (see [11] for details on the pattern mechanism) to rewrite the process to include the automated decision. Finally, during process execution, the decision is handled by an automated decision service. It takes the original decision's input, enriches it per the defined mappings with operational data and feeds it to the decision classifier who in turn determines the class label.

4 Evaluation

After the decision automation approach has been explained thoroughly in the previous sections, this section will now quantitatively evaluate how well the approach performs in a sample application scenario. First, we will explain the evaluation scenario as well as the evaluation design. Then, we will discuss the evaluation results and their implications on the approach presented in this paper.

4.1 Evaluation Design

The evaluation is based on a simplified loan approval scenario shown in Fig.5. In it, a bank clerk classifies loan requests into high, middle and low risk loans. As

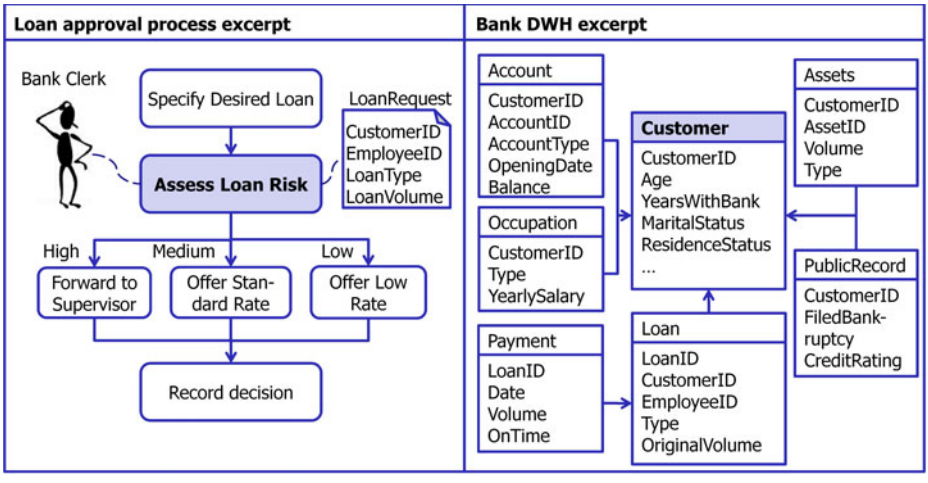

Fig. 5. Loan Approval Scenario

only existing customers can get a new loan, most of the information important to the decision is stored in operational data sources as shown in Fig.5. We chose this scenario despite its typically already high degree of automation, as the factors involved are fairly complex and both the implicit and explicit decision factors are well publicized [17]. This allows us both to generate realistic test data and makes the scenario representative for other, less complex, decisions.

To assess the feasibility of our approach for automating the "Assess Loan Risk" activity using a decision classifier D_{CL}, the evaluation is set up as follows:

- **Input data**: To assess the impact of data integration on the quality of D_{CL}, we conduct the measurements using different input data sets: A minimal process data set containing only the loan type and volume, a rich process data set containing additionally the customer's account balance and credit rating and the integrated data set, containing the complete set of attributes shown in Fig.5.
- **Classification algorithms**: To assess the suitability of different types of algorithms, we employ three different ones: The WEKA implementations (in brackets: the WEKA names) of the C4.5 decision tree (J48), the naive bayesian classifier (NaiveBayes) and a multilayer perceptron (MultilayerPerceptron).
- **Sample size**: In total, we use a set of 27.000 sample processes. The set is split into 18.000 training and 9.000 testing samples (without cross validation).
- **Quality measurement**: The quality of the classifier is measured by the classification accuracy, defined as the share of correctly classified samples compared to the total number of classified samples.

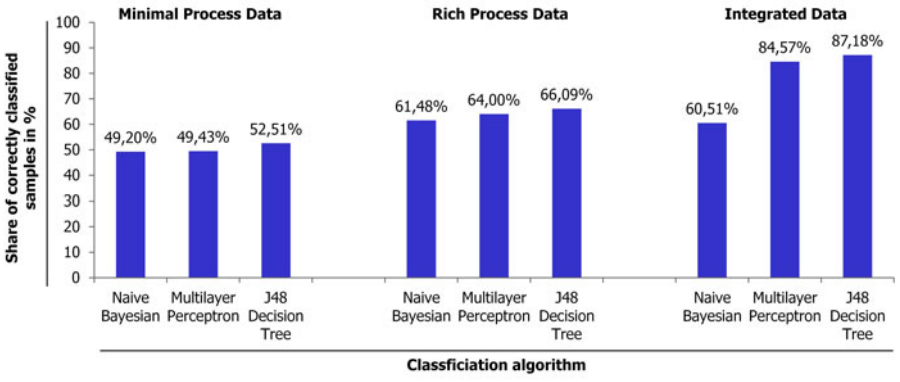

Fig. 6. Evaluation Results

4.2 Results

The results of the evaluation are shown in Fig.6. The following observations can be made: First, non-suprisingly, using integrated data (87,18% accuracy with the best classifier) creates clearly superior results over both the minimal (52,51%) and the rich process data (66,09%). Second, while the benefit of using integrated data is somewhat reduced by the richer process data, it still remains significant. Third, the choice of classifier makes a great difference: The naive bayesian classifier performs notably worse than the other two classifiers. This is largely because the classifier assumes that attributes are independent [6], which is not the case in this scenario. The best performing algorithm is the decision tree, with the multilayer perceptron trailing behind. In this scenario, this could be attributed to the greater flexibility of decision trees with regards to the partitioning of training data. An alternative explanation can be found in [8].

Overall, the evaluation has demonstrated the feasibility of our approach for the given application scenario. As we on purpose selected a scenario with a multitude of influence factors and a complex decision logic, it is reasonable to assume that the approach also works under other circumstances. Further, it has shown that the inclusion of operational data significantly improves the quality of the decision classifier - to an extent that largely depends on the initial information richness of the process data.

5 Related Work

This paper is part of our work on the *dBOP* platform [9]. The data integration layer is discussed in [13]. Our integrated warehouse is similar to the process warehouse presented in [2], however, it offers better support for connecting process and operational data. The methods employed in the analysis layer are adapted from standard data mining and machine learning literature [6] [4]. Examples for their application can be found in [12] and [10]. The optimization layer builds heavily on existing research into business process optimization techniques, such

as [14]. Its role within the *dBOP* is the subject of [11]. Overall, the approach of a system that automatically adapts according to a set of rules and feedback from its execution can be conceptually seen as an application of cybernetics [18] to *BPO*. The workflow controlling framework discussed in [19] and the process analysis approach of [3] are somewhat similar to our platform in that they use custom analysis tools to gain process insights. However, their integration and analysis capabilities are limited and they lack an optimization layer.

Various other papers deal with the application of machine learning and data mining techniques to process data under the umbrella term of *Process Mining*. Closest related to this paper is the decision mining approach presented in [15] and [16]. The focus of the presented approach seems to be, however, more on process model validation (i.e., verification of whether a certain process execution instance conforms to a given process model) and less on actual decision automation. It hence only considers process data and does not provide a classifier for process execution, which restricts its application to decisions with limited complexity.

6 Conclusion and Outlook

In this paper, we have presented an approach for the automation of decisions in business processes. We have shown how decision automation can be transformed into a standard classification problem and demonstrated both qualitatively and quantitatively, that data integration can greatly increase the quality of a decision classifier. Further, we have presented our *dBOP* platform, which allows for a direct application of the analysis results through process rewriting and an integrated decision classifier service.

Future research plans on the topic include the application of machine learning techniques to a broader set of process optimization scenarios, such as the selection of process resources or the retrieval of process variants. Further, we are planning to do an empirical study based on business expert interviews to determine which level of accuracy would constitute, in a practical environment, a "good" classifier and what the ramifications of its use would be. Additionally, we are exploring the application of our data-driven approach to different areas of Business Process Management, such as the construction of simulation models and the definition and verification of business rules.

References

1. Aumüller, D., Do, H.H., Massmann, S., Rahm, E.: Schema and ontology matching with COMA++. In: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data (2005)
2. Casati, F., Castellanos, M., Dayal, U., Salazar, N.: A generic solution for warehousing business process data. In: Proceedings of the 33rd International Conference on Very Large Data Bases, pp. 1128–1137 (2007)
3. Castellanos, M., Casati, F., Dayal, U., Shan, M.C.: A comprehensive and automated approach to intelligent business processes execution analysis. Distributed and Parallel Databases 16(3), 239–273 (2004)

4. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern classification. Wiley Interscience, Hoboken (2000)
5. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: An update. ACM SIGKDD Explorations Newsletter 11(1), 10–18 (2009)
6. Han, J., Kamber, M.: Data mining: concepts and techniques. Morgan Kaufmann, San Francisco (2006)
7. Leyman, F., Roller, D.: Production Workflow. Prentice-Hall, Englewood Cliffs (2000)
8. Liu, X., Bowyer, K.W., Hall, L.O.: Decision trees work better than feed-forward back-prop neural nets for a specific class of problems. In: 2004 IEEE International Conference on Systems, Man and Cybernetics, vol. 6, pp. 5969–5974. IEEE, Los Alamitos (2005)
9. Niedermann, F., Radeschütz, S., Mitschang, B.: Deep business optimization: A platform for automated process optimization. In: Proceedings of the 3rd International Conference on Business Process and Services Computing (2010)
10. Niedermann, F., Radeschütz, S., Mitschang, B.: Design-time process optimization through optimization patterns and process model matching. In: Proceedings of the 12th IEEE Conference on Commerce and Enterprise Computing (2010)
11. Niedermann, F., Radeschüz, S., Mitschang, B.: Business process optimization using formalized patterns. In: Proceedings BIS 2011 (2011)
12. Radeschütz, S., Mitschang, B.: Extended analysis techniques for a comprehensive business process optimization. In: Proceedings KMIS (2009)
13. Radeschütz, S., Niedermann, F., Bischoff, W.: Biaeditor - matching process and operational data for a business impact analysis. In: Proceedings EDBT (2010)
14. Reijers, H.A., Mansar, S.L.: Best practices in business process redesign: an overview and qualitative evaluation of successful redesign heuristics. Omega 33(4), 283–306 (2005)
15. Rozinat, A., van der Aalst, W.M.P.: Decision mining in proM. In: Dustdar, S., Fiadeiro, J.L., Sheth, A.P. (eds.) BPM 2006. LNCS, vol. 4102, pp. 420–425. Springer, Heidelberg (2006)
16. Rozinat, A., van der Aalst, W.M.P.: Decision mining in business processes (2006)
17. Thomas, L.C.: A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers. International Journal of Forecasting 16(2), 149–172 (2000)
18. Wiener, N.: Cybernetics: Control and communication in the animal and the machine. MIT Press, Cambridge (1948)
19. zur Mühlen, M.: Workflow-based process controlling: foundation, design, and application of workflow-driven process information systems. Logos Verlag (2004)