Deep Business Optimization: Making Business Process Optimization Theory Work in Practice

Florian Niedermann and Holger Schwarz

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

Abstract. The success of most of today's businesses is tied to the efficiency and effectiveness of their core processes. This importance has been recognized in research, leading to a wealth of sophisticated process optimization and analysis techniques. Their use in practice is, however, often limited as both the selection and the application of the appropriate techniques are challenging tasks. Hence, many techniques are not considered causing potentially significant opportunities of improvement not to be implemented. This paper proposes an approach to addressing this challenge using our deep Business Optimization Platform. By integrating a catalogue of formalized optimization techniques with data analysis and integration capabilities, it assists analysts both with the selection and the application of the most fitting optimization techniques for their specific situation. The paper presents both the concepts underlying this platform as well as its prototypical implementation.

Keywords: Business Process Optimization, Optimization Techniques, Business Process Analytics, Data Mining, Tool Support.

1 Introduction

In this section, we first present the approach that is usually employed during *Business Process Optimization (BPO)*. Then, using the introduced approach, we discuss the challenges that are typically encountered when transferring *BPO* research results into practice. Finally, we show how the *deep Business Optimization Platform (dBOP)* can help to close the gap between *BPO* research and practice.

1.1 Business Process Optimization

In the past decade, businesses have moved from tweaking individual business functions towards optimizing entire business processes. Originally, this trend - then called Business Process Reengineering [7] - was triggered by the growing significance of Information Technology and the trend towards globalization. The increasing volatility of the economic environment and competition amongst businesses has further increased its significance over the past years. Hence, the ability

T. Halpin et al. (Eds.): BPMDS 2011 and EMMSAD 2011, LNBIP 81, pp. 88–102, 2011.

BPO Step	How BPO research helps	Challenges w.r.t. application in practice
Data Integration	▪ General schema integration methods from databases ▪ Process Data Warehousing	▪ Schema integration methods not well-suited for process data; significant manual efforts ▪ Process DWH does not integrate operational data
Analysis	▪ Metrics and KPI systems ▪ Data and Process Mining tools	▪ Metrics either simple or domain-specific ▪ Application requires considerable know-how; algorithms not tied to optimization techniques
Detection & Implemen-tation	▪ "Computer Science" techniques allow for automation ▪ Broad range of "Business" techniques, derived typically from "real life" projects	▪ Automation works only possible in a very narrow domain and a specific formalism ▪ Descriptions often vague with little guidance ▪ Techniques not comparable with each other, interdependencies not described

Fig. 1. Challenges for Applying BPO Research in Practice

of a business to achieve superior process performance is nowadays one of the main sources of competitive advantage (or, in many cases, survival of the business).

To achieve this, nearly all companies have dedicated *Business Process Optimization (BPO)* staff and frequently run large-scale optimization projects. Technically speaking, the ultimate goal of *BPO* is the selection of the right process designs and the application of the most appropriate optimization techniques. To achieve this goal, *BPO* efforts ideally include the following three steps:

1. **Data integration:** As processes are cross-functional, data pertaining to the process can be spread over many different data sources. Hence, as a first step, all possibly relevant data needs to be collected and integrated.
2. **Data analysis:** After the raw data has been collected, both the process model and the process data need to be analyzed. This analysis can range from the calculation of basic metrics (such as duration, cost or frequency) to the application of data mining techniques to discover "hidden" insights.
3. **Detection and implementation of improvements:** Based on the analysis results, deficiencies within the process are detected. These can, e.g., relate to the process flow, the composition of activities or the resources used during the process execution. After assessing the deficiencies, appropriate techniques for addressing are selected and applied to the process or its context (e.g., the resources executing the process).

1.2 BPO Challenges

Recognizing the importance of *BPO*, considerable research efforts both in Computer Science and Business have been made to address the challenges and complexities of each of the individual *BPO* steps listed in the previous section. While they have been successful in many ways of advancing the broad field of *BPO*, several factors limit their usability in a *BPO* project.

While the details of the limiting factors are shown in Fig.1, they all can be traced to three common root causes. First, the data integration and analysis steps are often neglected. The optimization techniques assume that all relevant data is contained in a single data source. Further, typically only basic analysis techniques (such as metrics) are employed. Second, while there are plenty of

R1: Analytics	The approach needs to be able to integrate a broad range of heterogeneous data. Its analytics capabilities need to allow for the discovery of complex insights, even if the analyst is not highly familiar with data mining techniques.
R2: Optimization	To ensure that the most appropriate optimization techniques are used in any given scenario, the approach needs to be able to (semi-)automatically select and apply those techniques that match the description of the optimization goals given by the analyst. As it is likely that the analyst has additional knowledge about the process context, his input needs to be considered during the optimization.
R3: Integration	To facilitate adoption as well as minimize both the error rate and the need for human intervention, the various *BPO* steps need to be seamlessly integrated.

Fig. 2. *BPO* Usability Requirements

optimization techniques, each one is defined in isolation using a different formalism and/or with a different application domain in mind. This makes the combination of different techniques challenging, as their interdependencies are not defined and their results are sometimes not comparable. Finally, there is little to no integration between the different *BPO* steps, which makes especially the application of complex optimization techniques difficult.

1.3 The Deep Business Optimization Platform

To make *BPO* research work in practice, we need an approach that overcomes the challenges discussed in the previous section. Hence, we derive the requirements listed in Fig.2 for such an approach to be feasible.

Our *deep Business Optimization Platform (dBOP)* was developed with exactly these requirements in mind, which is why we use it throughout this paper as an example of how to achieve the paper's goal. The platform, as shown in Fig.3 consists of three integrated layers. The data integration layer handles the integration of heterogeneous data sources, with a specific focus on the integration of process and operational data. The integration layer is why we use the prefix "deep" as a qualifier for the platform name. It indicates that the optimization is based on an integrated, rather than an isolated view on the relevant data.

Based on the results of a graph analysis and knowledge about the employed optimization patterns, the analytics layer automatically processes and analyses the process data, both using standardized metrics and data mining techniques. Finally, the optimization layer utilizes an optimizer engine and the optimization patterns stored in the pattern catalogue to (semi-)automatically detect and apply process improvements.

Many of the *dBOP*'s technical details are extensively discussed in our previous work (see for instance [15], [12] and [13]). Hence, we the focus of this paper is to show how the different components work together to increase the usability of

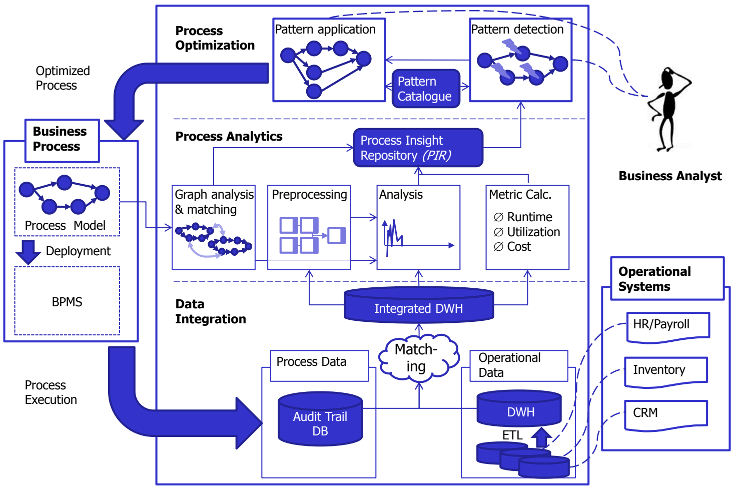

Fig. 3. dBOP Platform Overview

BPO research results by meeting the requirements laid out in Fig.2. To that extent, this paper aims to give a balanced account of the platform's benefits from a user's and practitioner's perspective and the most salient concepts underlying it.

In Section 2, we discuss the integration and analytics capabilities and how they help with meeting requirement **R1**. Building on these results, Section 3 introduces both the process model and the pattern catalogue necessary for meeting requirement **R2**. In Section 4, we see how all the components come together in the prototypical implementation of the *dBOP* to fulfill requirement **R3**. After a brief evaluation and case study in Section 5, we discuss related work in Section 6 before concluding the paper in Section 7.

2 Data Integration and Analytics

As we have seen in the previous section, the first requirement for making *BPO* research work in practice is assisting the analyst with both the analysis of the process itself as well as the associated data. Therefore, we first take a look at how the *dBOP* uses data integration techniques to make sure that all relevant data is included in the analysis. Then, we present some of the data mining techniques employed and how they are matched to optimization techniques.

2.1 Data Integration

Broadly speaking, two different kinds of data are relevant for analyzing processes. To achieve integration, process activities pass data between each other. This *process data* is then typically stored in the audit trail database (see Fig.3). This data is flow-oriented, i.e., it contains a complete picture of the dynamic

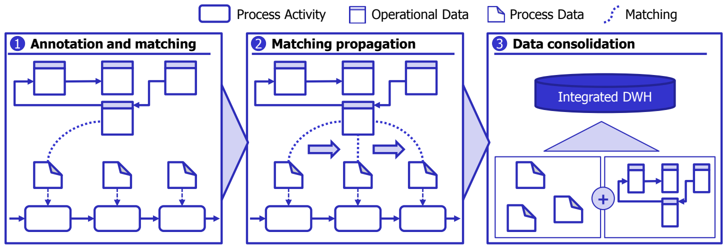

Fig. 4. Data Integration Approach

properties of the process (e.g., the process paths taken, the duration of the individual activities) but little information about the process subjects (such as the people and the items involved in it).

During their execution, processes additionally invoke one or several application systems which contain and generate data relevant to the process. This *operational data* is frequently consolidated in a Data Warehouse. It is subject-oriented, i.e., it contains a broad range of information about the process subjects, but little to no information about the process flow.

For a thorough process analysis, we need to integrate both kinds of data. During the design of the integration layer, we have found that two factors are imperative for this effort to succeed. First, the user needs to be assisted as much as possible with finding the most appropriate matches for integration. Second, the matching of the schema data needs to be combined with an according Data Warehouse structure and ETL (Extraction, Transformation, Load) [9] process.

Due to the different paradigms of process and operational data, classical schema matching approaches like [2] are not successful at meeting the requirements outlined above. This is why we have developed a specific approach for integrating operational and process data. As Fig.4 illustrates, it consists of three steps: In the matching step, the analyst matches the attributes that he knows to be the same, optionally (if both the process and operational data is semantically annotated) supported by a semantic reasoner. These basic matches are then taken in the next step and spread to other process elements according to a set of processing rules. One of these rules, for instance, propagates the matchings along data flow mappings (such as the one realized by `<assign>` statements in BPEL). Finally, the data is consolidated into an integrated DWH, similar to the one discussed in [4] which is used as the main data source for analysis techniques [15].

2.2 Analytics

After the process and operational data have been successfully consolidated, it can now be analyzed together with the process model to discover insights that are useful for the optimization. The analysis is made up by four components, as can be seen in the analysis layer in Fig.3. The first component analyses the process model and its execution behavior as well as matching it to other available process models. Together with the metrics calculation component, this indicates

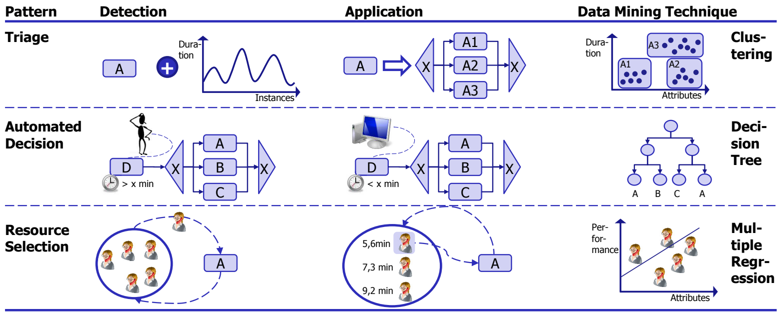

Fig. 5. Data Mining Techniques matched to Optimization Patterns

which areas of the process model might be interesting for further analysis. For these areas that have been selected as potentially interesting, the data is preprocessed and an in-depth analysis is performed using a suite of customized data mining algorithms.

As the spectrum of available data mining algorithms is huge [6], selecting the "right" algorithm is a challenging task that requires considerable analyst skills. To address this issue, each optimization pattern is internally matched to the suitable algorithm(s). This is exemplary shown in Fig.5 for three sample patterns.

The first pattern shown is the "Triage" pattern [16]. It is based on the notion that sometimes, during the execution of a certain activity, distinct variants of this activity emerge. In the interest of transparency and the ability to optimize the respective variants, the "Triage" pattern splits the activity. To determine if this pattern is applicable, the analyzer first checks if the activity's behavior (e.g., activity duration variance) suggests its presence. Then, a clustering algorithm, applied to different subsets of the activity's input data, tries to determine if there are "sufficiently different" [3] identifiable variants. If this is the case, the pattern can be applied.

In the middle of Fig.5 we see the "Automated Decision" pattern [11]. This pattern is based on the idea that in highly repetitive processes, a common pattern for manual decisions can be identified and the decision can hence be automated (or supported) by a classifier. For the application of this pattern, the analyzer first checks for all decisions in the process, if they exceed a certain duration (or cost, depending on the goal function). If that is the case, a decision tree (or alternatively, a multilayer perceptron [8]) for the decision is built and tested. If it is sufficiently good (as determined by the business analyst), it can be applied.

The third pattern displayed is the "Resource Selection" pattern. Its goal is to select, from a pool of resources, the one resource that execute the current activity "best" (e.g., fastest, with the highest quality, lowest cost - see Section 3.1), which is in contrast to the usual procedure, i.e., select any available resource randomly. For that purpose, a multiple (linear) regression model is used to predict the likely performance of any available resource. The one that shows the best likely performance for the given setup is selected (assuming that other defined constraints, such as utilization goals, are met).

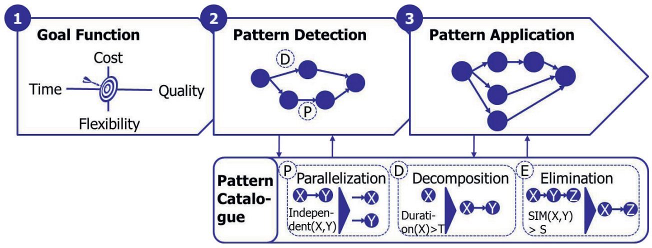

Fig. 6. *dBOP* Optimization Methodology

3 Optimization

In this section, we show how the *dBOP* uses the analysis results together with a set of formalized process optimization techniques for the semi-automated detection and implementation of improvement opportunities . First, we introduce the three-staged optimization approach and explain how it helps an analyst who is using it to achieve the desired results. Then, we present the criteria that are used to classify optimization patterns and give an excerpt of the pattern catalogue. Finally, we provide the detailed version of a sample pattern.

3.1 Optimization Overview

Each optimization pattern changes a process in a different way and with different effects. Further, many pattern have interdependencies that have an influence on, e.g., their order of execution. As there are currently more than 30 patterns to choose from, getting this right can be a daunting task for any analyst. This is why to make the application of these patterns practical (and to meet requirement **R2** of Fig.2), we have developed the three step methodology shown in Fig.6.

In the first step, the analyst is asked to select the goal function(s) he wants to achieve and define any constraints to apply to the optimization. In the next step, the optimizer tries to find instances of the patterns selected based on the previous step. This is achieved using a combination of metrics, graph analysis and data mining techniques, as we have seen in Section 2.2. In the final step, the analyst is presented with each pattern instance in an order that is likely to yield the optimal results. For those that he confirms, the process is modified accordingly.

3.2 Pattern Catalogue

At the core of the optimization methodology explained in the previous section is the pattern catalogue. In accordance with requirement **R2**, the pattern catalogue - an excerpt of which we see in Fig.7 - is a consistent collection of patterns which are mapped to a common formalism. To enable the optimizer to perform a qualified selection, the patterns are classified according to a wide range of criteria. First, each pattern is classified according to the type of changes it implements,

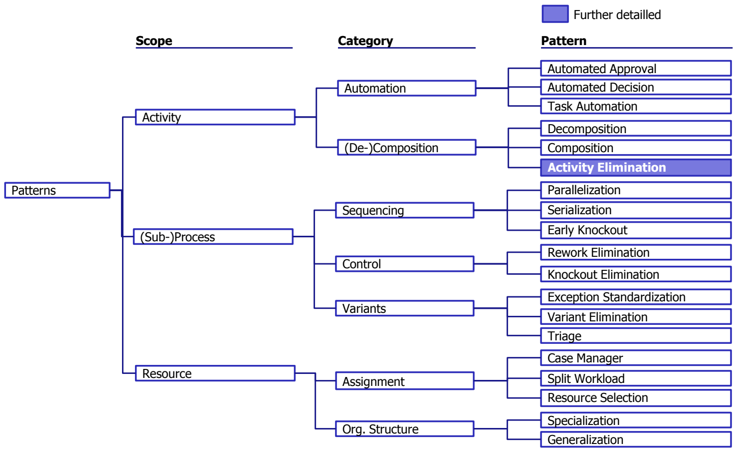

Fig. 7. Pattern Catalogue Excerpt

as shown in Fig.7. Second, it lists whether a pattern contributes, prevents or is indifferent towards the fulfillment of each of the optimization goals. Third, it describes the process stage (design, execution, analysis) during which each pattern can be applied. Fourth, it lists constraints, especially w.r.t. the execution order that should be considered when combining this pattern with other patterns. Finally, it indicates which data (just the model, process or integrated data) is required to run the pattern.

More details on the classification of the patterns can be found in [14] and the next section, where we present a simplified version of the process meta-model as well as the description of a sample pattern.

3.3 Meta-model and Pattern Example

One of the key ingredients for defining the optimization patterns is a common process meta-model. In line with requirement **R2** our main goal with regards to the meta-model is to have a good mix of usability, proximity to common modeling languages such as BPMN and formal reasoning power. Hence, we have selected a graph based meta-model. The concepts presented can, of course, also be applied with minor modification to other meta-models like Petri nets.

The meta-model is based on the process model graph presented in [10]. In this paper we only introduce a greatly simplified version, so please see [10] and [14] for more details.

Definition. *Simplified PM Graph: A simplified PM Graph G is a tuple $(V, O, N, C, E, \iota, o)$ 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.*

Pattern 1. Activity Elimination

Specification

Goals:	Time	↓	Cost	↓	Quality	→	Flexibility	→
Stage:	Design	✓	Execution	✗	Analysis	✓		
Data req.:	Model	✓	Process	(✓)	Operational	(✓)		

Detection

Require: Similarity threshold $T \in [0,1]$, Activity nodes N_A of process graph G
Ensure: All found pattern *instances*

```
instances = {}
for all act ∈ N_A do
   for all succ ∈ Successors(act, E_C) do
      if SIM(act, succ) ≥ T then
         instances = instances ∪ newInstance(act, succ, SIM(act, succ))
      end if
   end for
end for
return instances
```

Application

Require: Graph G, all *instances*, selected *instance*, *analyst* input
Ensure: The optimized (improved) process graph G_{opt}

```
G_opt = G
if confirms(instance, analyst) then
   DeleteNode(inst.succ)
   RemoveDependents(instances, inst)
end if
return G_opt
```

3. *N is the finite set of process nodes which includes the set of activities N_A, the start and the end node, the termination nodes N_T and the control nodes (XOR and AND Fork/Join) N_C.*
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.*

Using this meta-model, we can now define basic optimization patterns, such as the "Activity Elimination" pattern shown below in Pattern 1. As we can tell from the pattern specification, the elimination of redundant activities both lowers the process duration and cost. The pattern can be applied during the design and the analysis stage. It is flexible w.r.t. the employed data - the process model is sufficient, but if process and operational data is available, it can be used to further improve the required similarity measurement.

The idea of this pattern is that a high similarity of process activities can be an indicator for redundancy. To find similar activities, the pattern utilizes a

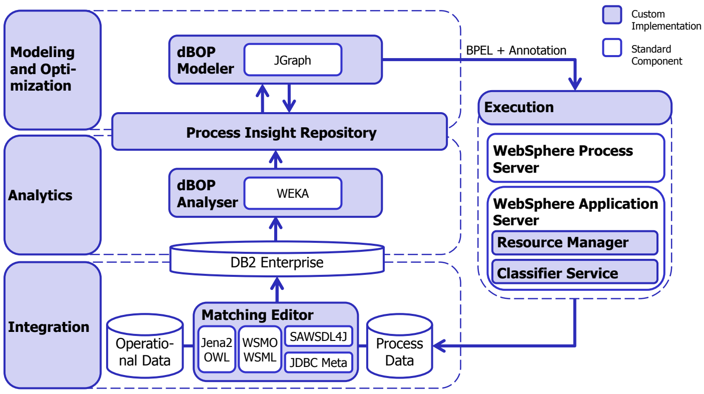

Fig. 8. dBOP Implementation Overview

similarity function $sim_A : N_A \times N_A \rightarrow [0, 1]$ [13], which measures the similarity of two activities on a scale of 0 (not similar at all) to 1 (identical) - in our experience, a threshold of 0.8 has shown a good correlation with human judgment. For those activities that are found highly similar, the analyst is asked to confirm whether they are redundant. If that is the case, the redundant activity is removed from the process.

4 Implementation and Prototype

After the previous sections have focused on explaining the concepts underlying the *dBOP*, we now take a closer look at its implementation. First, we give a brief overview over the different component implementations. Then, we present the *dBOP Modeler* in detail, as it is the core.

4.1 Implementation Overview

The implementation of the *dBOP* shown in Fig.8 achieves to fulfill the integration requirement **R3** by ensuring that the interfaces between the different components fit seamlessly into each other. Hence, while it is possible to use each component individually, transferring results is possible with no or minimal manual intervention.

The applications stack itself employs a mix of standard software and custom-developed applications

The *dBOP* **execution environment** is built on top of IBM WebSphere. The process execution itself occurs in IBM WebSphere Process Server, while run-time services such as the "Classifier" service for decision automation or the "Resource Manager" for resource allocation are built as enterprise applications and run on WebSphere Application Server. Note that instead of Process Server, we can run the process on any other BPEL (or for that matter YAWL-)compliant execution engine by providing a custom implementation of the mapping interfaces.

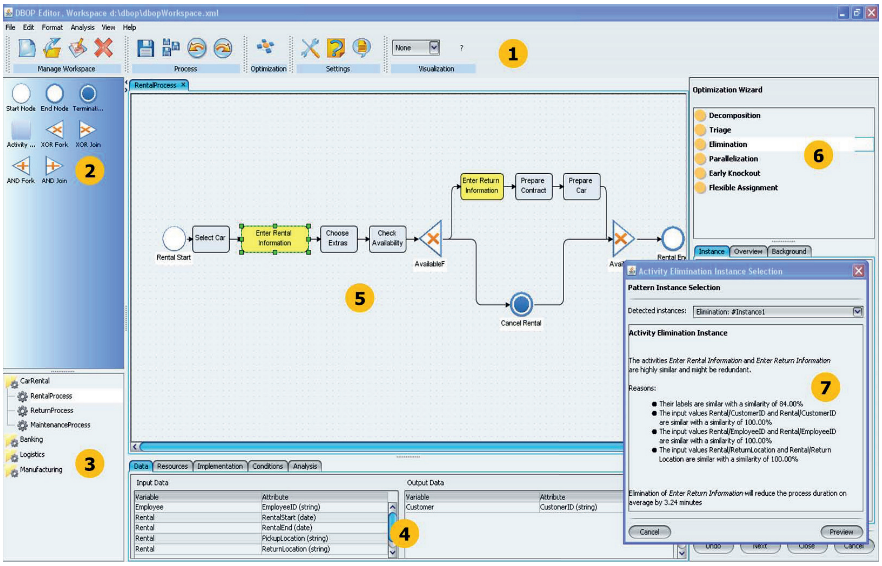

Fig. 9. *dBOP Modeler* Screenshot

The main component of the **integration layer** is the *Matching Editor* [15] which implements the matching approach described in section 2.1. For storing the integrated process and operational data, basically any sufficiently powerful relational database is feasible - our prototype uses DB2 Enterprise.

At the heart of the **analytics layer** is the *dBOP Analyzer*. It incorporates data preprocessing, metric calculation and data mining techniques to provide the necessary inputs for the optimization layer. Its implementation is based on a customized and extended version of the WEKA library [6].

The central component for the **modeling and the optimization** of the processes is the *dBOP Modeler* that we take a closer look at in the next section. For graph visualization and basic reasoning purposes, it uses the JGraph library [1] - beyond that, it is a custom implementation.

4.2 dBOP Modeler

After the overview of the whole platform's implementation in the previous section, we now take a closer look at one of the components - the *dBOP Modeler*. It is the central tool for process analysts, as it is used both for the modeling and the optimization of business processes. To allow for intuitive usage and a quick progression on the learning curve, the user interface, as shown in the screenshot in Fig.9, has been designed to be as close to standard modeling tools as possible.

The basic features and options (1) of the *dBOP Modeler* closely resemble those of standard modeling tools. Further, it allows the analyst to visualize analysis results and explore the effects of different process setups. New elements can be inserted into the model by dragging them from the element palette (2). Process

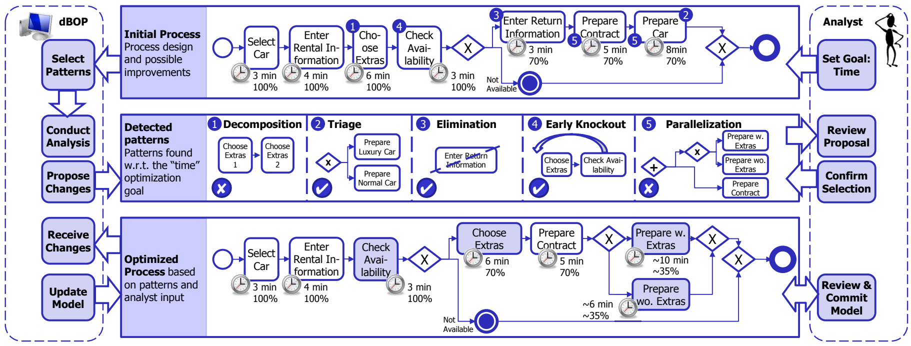

Fig. 10. Car Rental Case Study

models can be stored and loaded from the process repository (3), the contents of which can also be used for process model matching [13]. For enhancing the process model with data-, resource- and implementation information, additional facilities are provided (4).

The processes themselves are displayed in the modeling pane (5). As the *dBOP Modeler* is currently running in optimization mode, the modeling pane is visualizing the currently detected "Activity Elimination" pattern instance. This pattern was detected by the optimization wizard shown in (6) based on the optimization goal "time". It guides the analyst step-by-step through the optimization approach discussed above. Its current state shows for which patterns instances were found. Finally, before applying a pattern, the analyst gets detailed information about the expected effects (7) and, if applicable, possible conflicts that might arise from the application (e.g., when the "Activity Elimination" pattern would delete an activity whose outputs are used by a later activity).

5 Case Study and Evaluation

In this section, we briefly evaluate the *dBOP* in the light of the *BPO* requirements introduced in Fig.2. As the basis for our evaluation, we use the small "Car Rental" case study depicted in Fig.10 - the same process that can be seen in the screenshot in Fig.9.

In the case study, the analyst wants to optimize the process time. The *dBOP* hence analyses the process and the integrated DWH to find any patterns that either reduce process time or enhance general process attributes like transparency and manageability.

To improve transparency, the "Decomposition" pattern is proposed to split up the long running *Select Extras* activity. Further, a clustering of the *Prepare Car* execution results indicates that there are two distinct variants - one for cars with and one for cars without extras. To reduce the process duration, it is proposed to eliminate the possibly redundant "Enter Return Information" activity, move the knockout condition w.r.t the car availability [18] to the earliest possible position and parallelize the "Prepare Contract" and "Prepare Car" activities, as the one is done by a mechanic and the other is done the rental clerk.

In this case, the analyst confirms all but the "Decomposition" and "Parallelization" patterns, reducing both the processing time by on average 3.9 minutes and improving the process transparency through the "Triage" pattern. The "Decomposition" pattern is not applied as the analyst does not see a possibility of sensibly splitting it. The "Parallelization" pattern remains unused as sometimes, the customer sometimes cancels the process during the contract phase and hence does not need the prepared car.

The case study illustrates, that the *dBOP* is a suitable approach for fulfilling the requirements of Fig.2 and hence also suitable for transferring *BPO* research results into practice:

R1: Analytics	The *dBOP* automates most analytics and data integration steps. The analyst can hence focus on the reviewing the analysis results and the process optimization itself.
R2: Optimization	The patterns selection reflects the goal(s) and constraints defined by the user. The optimizer considers interdependencies between patterns. The analyst's superior knowledge of the process context is taken into account by leaving the decision about which pattern is applied with him.
R3: Integration	The *dBOP* ensures that all components work seamlessly together and that all data and instructions flow as require. Hence, the need for manual intervention is reduced only to those steps that can't be automated (such as giving "sensible" names to activities created by a split).

6 Related Work

As the *dBOP* main's goal is to improve the accessibility of existing research findings through integration, standardization and systematization, it is sensible to distinguish in the discussion of related work between the platform as a whole and the separate platform layers.

Looking at the *dBOP* as a whole, it can be seen as an application of cybernetics [19] to *BPO*. The workflow controlling framework discussed in [21] and the process analysis approach of [5] are somewhat similar to our platform in that they use custom analysis tools to gain process insights. However, their data integration capabilities are limited and they lack an integrated optimization layer.

Considering the integration of heterogeneous data sources as required for the integration layer, this is the classical domain of schema matching approaches such as [2] which, however, struggle with the specifics of process data. The integrated DWH is similar to the one presented in [4], however, with a stronger focus on including both process and operational data.

The analytics layer can be seen as a specific implementation of Business Process Analytics [20]. As this relatively new field has so far not yielded too many readily applicable algorithms, we however by and large rely on "classical" data mining and preprocessing techniques [8]. Implementation-wise, the WEKA framework [6] is the core component used for fulfilling this layer's tasks.

The techniques used in the optimization layer rely heavily on existing *BPO* literature such as [7] and [17]. Particular important sources for our work are existing surveys on *BPO* techniques such as [16] and research into particular optimization techniques like [18]. We have used these papers as both a source of additional patterns and leveraged some of their findings, e.g., on the different optimization goals. Please note that the mentioned sources are only representative and by no means comprehensive, as the literature survey considered more than hundred different computer science, manufacturing engineering and business publications on the subject.

7 Conclusion and Future Work

This paper has argued that for making techniques derived from *BPO* research findings more readily applicable in practice, an approach is needed that assists analysts both in the selection and the application of the proper techniques. For this purpose, we have introduced our *deep Business Optimization Platform (dBOP)* that implements such an approach by mapping optimization techniques to a common formalism and combines them with specialized data integration and analytics capabilities. We have demonstrated how the *dBOP* hence empowers even analysts with only basic knowledge to apply sophisticated process analysis and optimization techniques.

As our prototypical implementation of the *dBOP* platform is by and large complete, our current focus is on showing its usefulness in "real world" application scenarios - both for providing a realistic assessment of its strenghts and a perspective on its limitations. For that purpose, we are using two different evaluation modes. For showing its benefits from a *BPO* effectiveness perspective, we are collaborating with several service and manufacturing companies on the application of the *dBOP* to their processes. To demonstrate its benefits with regards to analyst productivity, we are currently conducting a user study where we analyze the effect of the various *dBOP* components on the outcome of the optimization of selected sample scenarios.

References

1. Alder, G.: Design and implementation of the JGraph swing component. Techical Report, (6) (2003)
2. 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)
3. Berkhin, P.: A survey of clustering data mining techniques. Grouping Multidimensional Data, 25–71 (2006)

4. 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)
5. 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)
6. 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)
7. Hammer, M., Champy, J.: Reengineering the corporation: a manifesto for business revolution. Brealey, London (1993)
8. Han, J., Kamber, M.: Data mining: concepts and techniques. Morgan Kaufmann, San Francisco (2006)
9. Inmon, W.H.: Building the data warehouse. Wiley, Chichester (2009)
10. Leyman, F., Roller, D.: Production Workflow. Prentice-Hall, Englewood Cliffs (2000)
11. Niedermann, F., Maier, B., Radeschütz, S., Schwarz, H., Mitschang, B.: Automated process decision making based on integrated source data. In: Proceedings BIS 2011 (2011)
12. 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)
13. 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)
14. Niedermann, F., Radeschütz, S., Mitschang, B.: Business process optimization using formalized patterns. In: Proceedings BIS 2011 (2011)
15. Radeschütz, S., Niedermann, F., Bischoff, W.: Biaeditor - matching process and operational data for a business impact analysis. In: Proceedings EDBT (2010)
16. 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)
17. Sharp, A., McDermott, P.: Workflow Modeling: Tools for Process Improvement and Applications Development. Artech House Publishers, Boston (2008)
18. Van der Aalst, W.M.P.: Re-engineering knock-out processes. Decision Support Systems 30(4), 451–468 (2001)
19. Wiener, N.: Cybernetics: Control and communication in the animal and the machine. MIT Press, Cambridge (1948)
20. zur Mühlen, M., Shapiro, R.: Business process analytics. Handbook on Business Process Management 2 (2009)
21. zur Mühlen, M.: Workflow-based process controlling: foundation, design, and application of workflow-driven process information systems. Logos Verlag (2004)