

BIAEditor - Matching Process and Operational Data for a Business Impact Analysis

Sylvia Radeschütz
IPVS
University of Stuttgart
Germany
sylvia.radeschuetz@ipvs.
uni-stuttgart.de

Florian Niedermann
IPVS
University of Stuttgart
Germany
florian.niedermann@gmx.net

Wolfgang Bischoff
IPVS
University of Stuttgart
Germany
wfb2@gmx.de

ABSTRACT

A profound analysis of all relevant business data in the company is necessary for optimizing business processes effectively. Current analyses typically exclusively run on business process execution data or on operational business data stored in a data warehouse. However, to achieve a more informative analysis and to fully optimize a company's business, a consolidation of all major business data sources is indispensable. Recent matching algorithms are insufficient for this task, since they are restricted either to schema or to process matching. Our demonstration presents *BIAEditor* that allows to annotate and match process variables and operational data models in order to perform such a global business impact analysis.

1. INTRODUCTION

Increasing competition and significantly shortened product lifecycles have led to a situation where fast adaptation and continuous optimization of processes are critical factors in determining the success of a company[17]. This is especially true for process-driven industries, as in the insurance or point-of-sales driven business. Process engines have been widely adopted that allow for quick changes to business while ensuring high process execution quality and flexibility. In order to identify optimization potential, companies typically rely on either a process analysis (e.g. monitoring and process mining) or on analysis of data (e.g. OLAP (Online Analytical Processing) and data mining) in a Data Warehouse (DWH) generated by operational business applications such as ERP systems (Enterprise Resource Planning). However, these methods usually fall short (or require significant manual labor[9]) when it comes to dealing with questions requiring an integrated view on both process and operational data. In the example of a car rental company looking to optimize its rental processes, a relevant question to a business analyst would be how trainings and work experience affect the execution time as well as the success of the

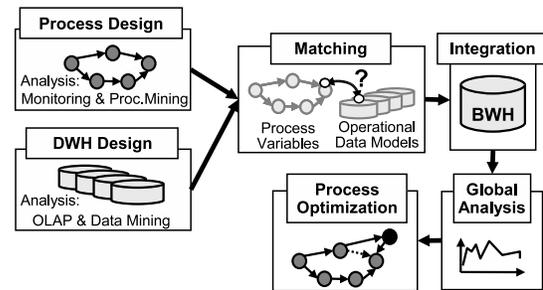


Figure 1: Business Impact Analysis

process. Answering this question requires both process data (execution data, paths taken) as well as additional operational data relating to the employee executing the process (work experience, trainings, demographics). In such a situation, an integrated global analysis tool can make a valuable contribution by ensuring that all relevant data is taken into account. Fig.1 shows the different phases of our approach to a so-called Business Impact Analysis (BIA). During ETL (Extraction-Transformation-Load), process and operational data are matched and integrated into a Business Warehouse (BWH) after data cleansing. A global analysis is now able to discover additional optimization issues, because aspects from both kinds of data sources are considered.

BIA builds on the high research attention that information integration has drawn [8]. Most approaches, e.g.[11, 16, 10], focus on a specific domain. Our approach is novel, because it goes beyond mere schema matching or mere process matching. Instead, operational data models are matched with process variables. Content and structure of audit trails that record process data vary significantly from operational data models which adds complexity to the matching task. Moreover, BIA profits from the trend towards semantic web services and reuses their semantic variable annotations.

In this demonstration, we introduce the *BIAEditor* for matching process data and operational data to enable the analyzing tasks for BIA. We briefly sketch our Matching Framework in the following section. Section 3 shows the system architecture of the *BIAEditor* and Section 4 provides the outline of our demo.

2. MATCHING FRAMEWORK

As shown in Section 1, it is important for an overall optimization of business processes to match all relevant process

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

EDBT 2010, March 22–26, 2010, Lausanne, Switzerland.

Copyright 2010 ACM 978-1-60558-945-9/10/0003 ...\$10.00

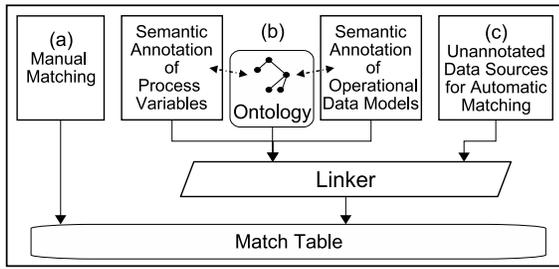


Figure 2: Matching Framework

data and operational data. In [14] we introduced a generic Matching Framework that allows for determining a match result and thus enables a global analysis based on an integrated view of both process-related data and operational data. The matching can be done manually or (semi-)automatically. Fig.2 shows those parts of the overall framework that are covered by the *BIAEditor*.

In the manual matching (a), the process variables are combined with a corresponding operational data model and stored into a match table. As soon as the user doesn't know the combining sources, semi-automatic matching is applied. For semi-automatic matching (b), each variable of the process as well as all operational data models have to be described semantically by adding an appropriate URI (Uniform Resource Identifier) reference to a concept in an ontology. Based on these annotations, the linker of the framework infers matches between process data and operational data from the rules in the used ontology. For non-annotated process and operational data, the linker also finds matches in the automatic matching (c). In [15], we proposed an approach for this manual matching and for the semantic annotation of process variables, while in [13] we concentrated on the operational data model annotation.

3. SYSTEM ARCHITECTURE

In this section, we discuss the main *BIAEditor* components (see Fig.3) and provide details of its implementation.

3.1 Model Transformation Engine

The Model Transformation Engine (MTE) gets the annotatable models, process variables or operational DWH schemas, and maps them to the corresponding representation of the *BIAEditor* structure. MTE allows to load BPEL [7] process variables directly from a process file or from the audit trail after deploying the process to a BPEL process engine. It applies a vendor-specific BPEL process engine mapping to a given BPEL process description to transform the variables to the internal *BIAEditor* structure. *BIAEditor* usually annotates variables of the audit trail. However, an annotation of the process file might be advantageous when the variables can't be loaded from the BPEL process engine because it is disconnected. In this case the variables are annotated within the file. But later on the process must be redeployed and its annotations must be processable by the process engine for subsequent analysis tasks. Operational DWH schemas may be available in different formats. MTE maps both relational and XML DWH schemas from the leading vendors to the *BIAEditor* structure and supports also the standard CWM [12] (Common Warehouse Metamodel).

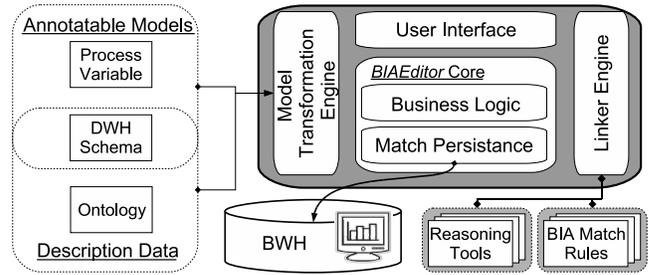


Figure 3: *BIAEditor* System Architecture

Description data that is used for annotation is transformed by MTE into the *BIAEditor* structure as well. The operational DWH schemas in Fig.3 can be loaded not only as annotatable models but also as description data for a manual matching with process variables. Furthermore, the *BIAEditor* allows to visualize both OWL[6] and WSML[5] ontology structures.

3.2 *BIAEditor* Core

The editor core provides all facilities that are required for serializing, traversing, navigating and de-serializing of description and annotatable data. In this capacity, it controls the Model Transformation Engine.

It further ensures persistence of matched data models. Process variables are stored with their matches in the audit trail of the associated process engine using a native XML type. The match information is used in the BWH for BIA.

3.3 Linker Engine

The task of the Linker Engine is to find matching operational data models for all relevant variables in the given process. For semantically annotated data it applies a suitable reasoning tool to infer logical consequences from the loaded WSML or OWL ontologies. For this purpose, the linker holds a selection of appropriate inference rules. We are especially interested in rules that discover synonyms, subclasses, equality or union relationships as well as user defined axioms between the matching data. For non-annotated data, the linker employs BIA Match Rules that consider useful process features as activity and operation names or the data flow in the process in addition to common schema matching algorithms.

3.4 User Interface

The user interface controls the whole matching process. It offers the following functionality: *Load Data*: A connection to a BPEL engine or an operational data source is set up or a process file, a CWM or XML file is referenced. The data is loaded into the artifact explorer (see (1) in Fig.4) and its structure visualized in the annotation explorer (2). *Annotate Data*: In the Annotation View of the editor (Fig.4) a node's annotation can be entered textually in the attribute table (4) or provided by clicks with the description data explorer (3) where either an ontology or operational DWH schemas are illustrated. *Match Data*: The Linker Engine is called for matching. Its results are shown in the Match View (Fig.6). Both a match overview and detailed views of the matched DWH schemas are provided. Manual matching has been done already one step before. *Save Match*: The match is saved by the *BIAEditor* core.

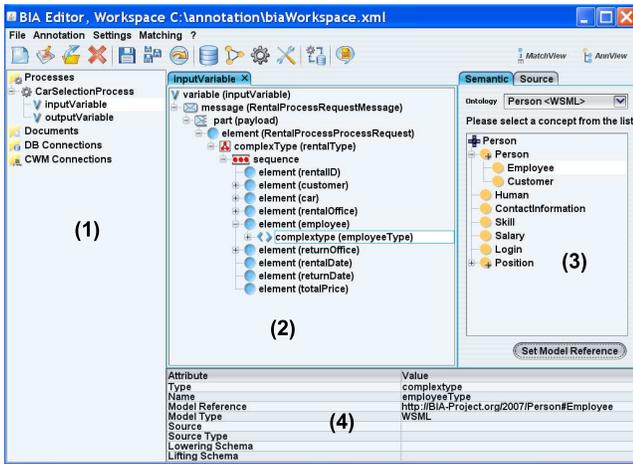


Figure 4: *BIAEditor* (Semantic Annotation View)

3.5 Implementation

At the core of the editor, a lean business logic has been implemented in Java 1.5 that provides the conceptual model that is used to load, annotate, match and save a number of annotatable artifacts (e.g. WSDL 1.1 and 2.0 descriptions, XML schema definitions). For that purpose, it utilizes a number of standards like SAWSDL[4] and of open source APIs, e.g. OWL and WSMML reasoners (e.g. Pellet[2]) or JDOM (CWM XMI and XML handling). At the moment, the editor supports workflows executed on either IBM WebSphere Process Server or Oracle BPEL Process Server. Since the editor core abstracts over the actual annotation artifacts, it can be extended, e.g. by adding another process engine.

3.6 Generality Issues

A key advantage of the *BIAEditor* is that it makes the annotation and matching independent from a specific operational data source by using CWM. Most data can be imported into our editor, as CWM is supported by many leading DWH vendors. MTE is also allowed to be adapted to import further operational data sources directly. So the editor can be used as general tool in other ontology-based information integration scenarios as well.

The editor can also easily be extended, as mentioned in Sec. 3.5. That provides the flexibility to adjust it to future extensions of BPEL as well as to variable constructs of other process languages or to further ontology enhancements.

4. DEMONSTRATION

The goal of this demo is to show the complete BIA optimization cycle (cf. Fig.1). The demo will present the annotation of a BPEL process description and of an operational CWM. It will then move on to the matching of the two models and shows how a subsequent global process optimization will be performed using commercially available tools.

1. *Process Design*: Our demo starts with a BPEL sample process that is modeled and visualized using Oracle JDeveloper [1] (see parts of it in Fig.5). The process is part of a car rental service and describes the selection of a rental car. If no car is available during the ordered rental period, an employee must execute a human task

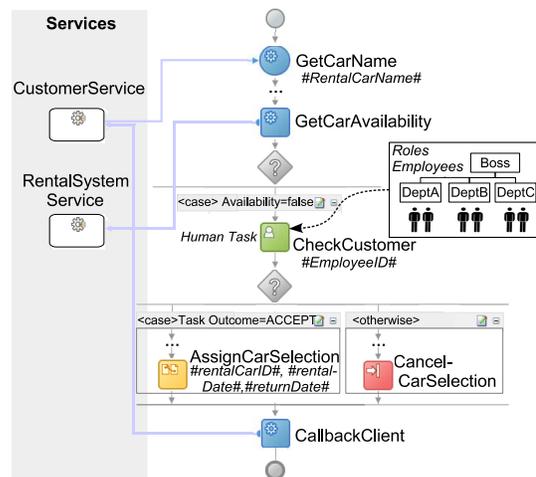


Figure 5: CarSelectionProcess in CarRentalAgency

activity CheckCustomer to prove if the customer would also accept another car class. The task isn't assigned directly to one employee, but to one of the available roles. Thus, CheckCustomer can be claimed and executed by all agents from departments A, B or C. For our demo, we want to investigate on what terms another car is accepted or the process is canceled. So we focus on the outcome of this human task and its employeeID variable. All process variables are marked in Fig.5 by hash marks # for clarity.

2. *DWH Design*: All operational car rental data sources are loaded into a DWH. That includes also data about rental agents, their address, but also their trainings etc. In our demo, we use IBM Infosphere DWH [3].
3. *Matching*: The demonstration shows both annotation and matching capabilities of the *BIAEditor*.
 - (a) *Semantic Annotation & Automatic Matching*: After deploying the process, it is loaded in the *BIAEditor* (Fig.4) that illustrates its variables. A WSMML ontology about a person taxonomy within the company is used for semantic annotation of the employeeID. Additionally, the DWH schema is also annotated semantically. We show how a sample CWM in XMI format is loaded and annotated. Afterwards, the linker is called. It considers annotated parts, but also finds matches for non-annotated data models. Its matching results are displayed (Fig.6). In this scenario, employeeID is matched with the Empno column of the HumanResource schema. RentalID hasn't been matched yet and is thus marked in red.
 - (b) *Manual Matching*: As an alternative, we can directly match employee IDs in the process data and the DWH. Therefore, the DWH source of step 2 is loaded and visualized instead of the ontology. In this matching scenario, no linker is required.
4. *BWH Integration*: Both executed car rental processes from the audit trail and DWH data are integrated in a BWH for what we use IBM Infosphere DWH again.

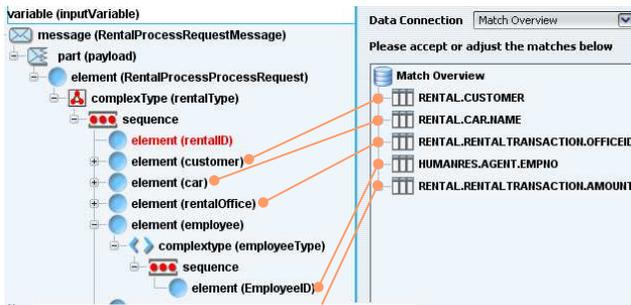


Figure 6: BIAEditor (Match View)

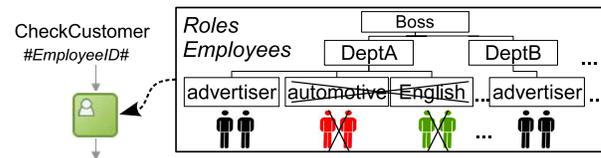


Figure 8: Optimized Process

5. FUTURE WORK

Planned extensions of BIA deal with the analysis side. We will keep exploring the adaption of common data and process mining algorithms to further enhance the insights gained from a global analysis.

6. REFERENCES

- [1] Oracle JDeveloper 11g. Available: <http://oracle.com/technology/software/products/jdev/>.
- [2] Pellet reasoner. Avail.: <http://clarkparsia.com/pellet/>.
- [3] IBM Infosphere Warehouse. Available: <http://www-01.ibm.com/software/data/infosphere/warehouse/>.
- [4] W3C Recommendation: Semantic Annotations for WSDL and XML Schema. Available: <http://www.w3.org/TR/sawSDL/>.
- [5] W3C Submission: Web Service Modeling Language. Available: <http://www.w3.org/Submission/WSML/>.
- [6] W3C Recommendation: Web Ontology Language. Available: <http://www.w3.org/TR/owl-ref/>, 2004.
- [7] OASIS Standard: Web Services Business Process Execution Language Version 2.0. Available: <http://docs.oasis-open.org/wsbpel/2007/>.
- [8] P. A. Bernstein and L. M. Haas. Information integration in the enterprise. *Commun. ACM*, 51(9):72–79, 2008.
- [9] F. Casati, M. Castellanos, U. Dayal, and N. Salazar. A generic solution for warehousing business process data. In *Proc. VLDB Conf.*, 2007.
- [10] J. C. Corrales, D. Grigori, M. Bouzeghoub, and J. E. Burbano. Bematch: a platform for matchmaking service behavior models. In *Proc. EDBT Conf.*, 2008.
- [11] D. Dou and P. LePendu. Ontology-based integration for relational databases. In *Proc. ACM SAC*, 2006.
- [12] J. Poole, D. Chang, D. Tolbert, and D. Mellor. *Common Warehouse Metamodel Developer's Guide*. Wiley, January 2003.
- [13] S. Radeschütz and B. Mitschang. An annotation approach for the matching of process variables and operational business data models. In *Proc. CAINE Conference*, 2008.
- [14] S. Radeschütz, B. Mitschang, and F. Leymann. Matching of process data and operational data for a deep business analysis. In *Proc. I-ESA Conf.*, 2008.
- [15] S. Radeschütz, F. Niedermann, and B. Mitschang. Ein Annotationsansatz zur Unterstützung einer ganzheitlichen Geschäftsanalyse. In *Proc. DW Conf.*, 2008.
- [16] E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *VLDB Journal*, 2001.
- [17] S. Weerawarana, F. Curbera, F. Leymann, T. Storey, and D. F. Ferguson. *Web Services Platform Architecture*. Prentice Hall PTR, 2005.

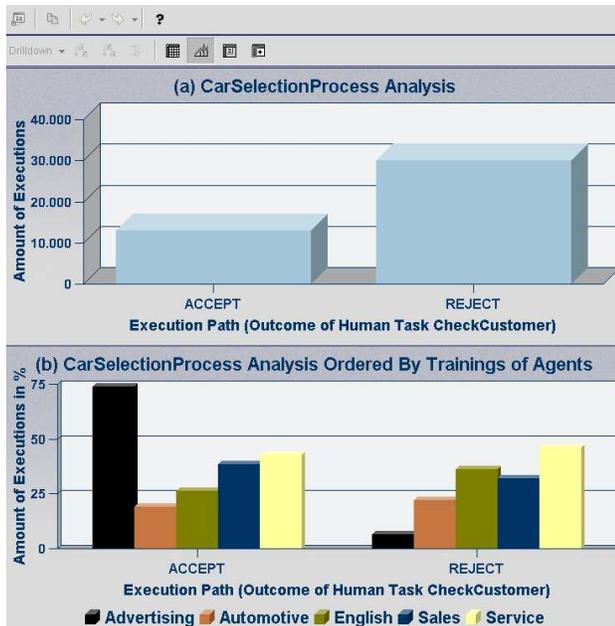


Figure 7: BIA Results

5. *Global Analysis*: Fig.7 demonstrates sample analyses performed by IBM Alphablox [3] in the Infosphere DWH. A query about process data (a) shows that 70% of the executions of task CheckCustomer are canceled ("reject" outcome), but no reasons for that result. The analysis of processes and DWH together (b) however reveals that those employees execute this task well ("accept" outcome) if they are trained in advertising.
6. *Process Optimization*: Using the BIA result, the process can be optimized. We reduce the group of employees that are allowed to execute CheckCustomer (Fig.8). Only agents that are fit in advertising are allowed to execute the task now instead of all Dept A+B+C. Trainings for advertising are offered to the other agents. This leads to a better process performance and company profit, because now only 20% of the processes might be canceled. Besides role restrictions there are further ways using BIA results to optimize the process. The syntax of the flow could be changed, e.g. by adding another case activity and deploying agents with necessary skills for specific customer groups.