Ein Annotationsansatz zur Unterstützung einer ganzheitlichen Geschäftsanalyse

Sylvia Radeschütz, Florian Niedermann, Bernhard Mitschang

Institut für Parallele und Verteilte Systeme
Universität Stuttgart
Universitätsstr. 38
70569 Stuttgart
{sylvia.radeschuetz | bernhard.mitschang}@ipvs.uni-stuttgart.de
florian.niedermann@gmx.de

Abstract: Die Verbesserung der Geschäftsprozesse in einem Unternehmen spielt eine immer wichtigere Rolle, um Wettbewerbsvorteile gegenüber der Konkurrenz zu erlangen. Dafür ist eine umfassende Analyse nötig über alle verfügbaren Informationen in diesem Unternehmen. Aktuelle Verfahren konzentrieren sich entweder auf die Analyse von Prozessdaten oder die Analyse von operativen Anwendungsdaten, die typischerweise in einem Data Warehouse vorliegen. Für die Ausführung einer tiefergehenden Analyse ist es jedoch notwendig, Prozessdaten und operative Daten zu verknüpfen. Dieser Beitrag stellt zwei Ansätze vor, welche es ermöglichen, diese Daten effektiv und flexibel zusammenzuführen. Der erste Ansatz stellt eine direkte Verknüpfung von Entitäten aus den Prozessdaten mit Entitäten aus den operativen Daten her. Die Verknüpfung im zweiten Ansatz beruht hingegen auf der semantischen Beschreibung der Daten. Beide Methoden sind in einem Werkzeug realisiert.

1 Einleitung

Vor dem Hintergrund der Globalisierung und der Verbreitung des E-Business setzen Unternehmen vermehrt auf die Etablierung hochflexibler Geschäftsprozesse. Workflowsysteme automatisieren die Geschäftsprozesse und verbessern ihre Qualität. Dabei entscheidet eine schnelle Anpassung und Optimierung der Geschäftsprozesse an neue Marktsituationen oftmals über den Erfolg eines Unternehmens [We05]. Heutige Analyseverfahren im Unternehmen fokussieren sich entweder auf die Prozessanalyse (Monitoring und Workflow (WF) Mining) oder auf Analysen über operative Anwendungsdaten (Business Intelligence mit OLAP-Analysen und Data Mining). Zur weiteren Optimierung der Prozesse sind jedoch ganzheitliche Analysen notwendig, die Daten aus Prozessen und operativen Systemen gemeinsam auswerten. Diese sind bisher nur mit Hilfe aufwändiger manueller Aktivitäten zur Analysezeit möglich.

Somit verzichten viele Unternehmen bei der Geschäftssteuerung auf das Potential solcher umfassenden Analysen und führen die Analyse der Prozesse und der operativen Daten mit Hilfe getrennter Systemumgebungen durch. Die Prozessanalyse allein zeigt zwar auf, wie z.B. die Bearbeitungszeit der Aktivitäten durch bestimmte Angestellte variiert, aber sie gibt keinen Aufschluss darüber, welche Faktoren (z.B. Arbeitserfahrung oder Fortbildung) dabei eine Rolle spielen. Die Daten für diese Faktoren liegen in einem operativen Data Warehouse (DWH). Auch eine rein operative Datenanalyse führt nicht zu einer Prozessoptimierung, da im DWH wiederum wichtige Prozessdaten für die Analyse fehlen.

Werden Prozessdaten und operative Daten jedoch während des ETL (Extraktion, Transformation, Laden) zusammengeführt (Matching in Abb. 1) und in einem DWH integriert, können ganzheitliche, unternehmensweite Analysen einen wertvollen Beitrag zu weiteren Prozessoptimierungen leisten, indem Faktoren aus beiden Datenquellen berücksichtigt werden. Prozesseigenschaften wie die Prozesslaufzeit können nun bezüglich ihrer Kausalitäten organisiert werden. So kann z.B. geprüft werden, welchen Einfluss Arbeitserfahrung und Fortbildungen auf die Bearbeitungszeit bestimmter Prozessaktivitäten haben.

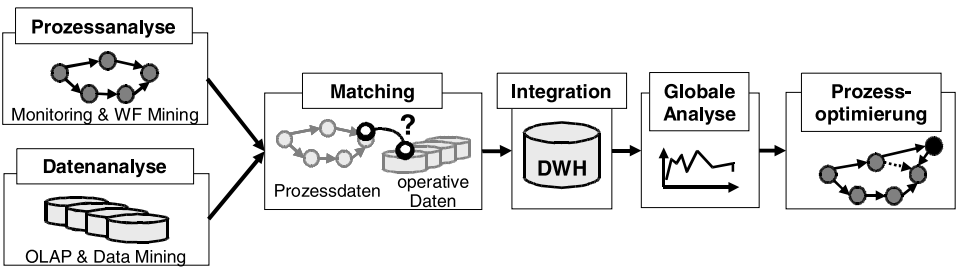

Abbildung 1: Unternehmensweite Geschäftsanalyse

In diesem Beitrag werden zwei Match-Methoden vorgestellt, die es ermöglichen, diese Daten sowohl direkt als auch auf semantischer Ebene zu verknüpfen. Diese Methoden sind Teil eines größeren Frameworks, das zunächst in Kapitel 2 eingeführt wird. Basierend auf den Methoden wird ein Editor entwickelt. Die Match-Methoden selbst, wie auch die Konzepte und die Realisierung des Editors, werden in Kapitel 3 und 4 beschrieben. In Kapitel 5 folgen weiterführende Betrachtungen, wie eine ganzheitliche Analyse mit Hilfe der daraus resultierenden Verknüpfung der Daten durchgeführt werden kann. Die Methoden und der Editor werden in Kapitel 6 anschließend evaluiert. Verwandte Arbeiten (Kapitel 7) und eine Zusammenfassung (Kapitel 8) bilden den Abschluss.

2 Matching Framework

Wie in Kapitel 1 dargelegt, ist es für eine umfassende Optimierung der Geschäftsprozesse notwendig, dass alle relevanten Prozessdaten und operativen Daten zusammengeführt werden. Die Verknüpfung der Daten kann manuell oder (halb)automatisch erfolgen. Abb. 2 gibt eine Übersicht über unser Framework, das Match-Methoden über verschie-

dene Workflow-Management-Systeme, operative Datenquellen und semantische Beschreibungssprachen hinweg integriert und anschließend den Match in einer Annotationstabelle ablegt.

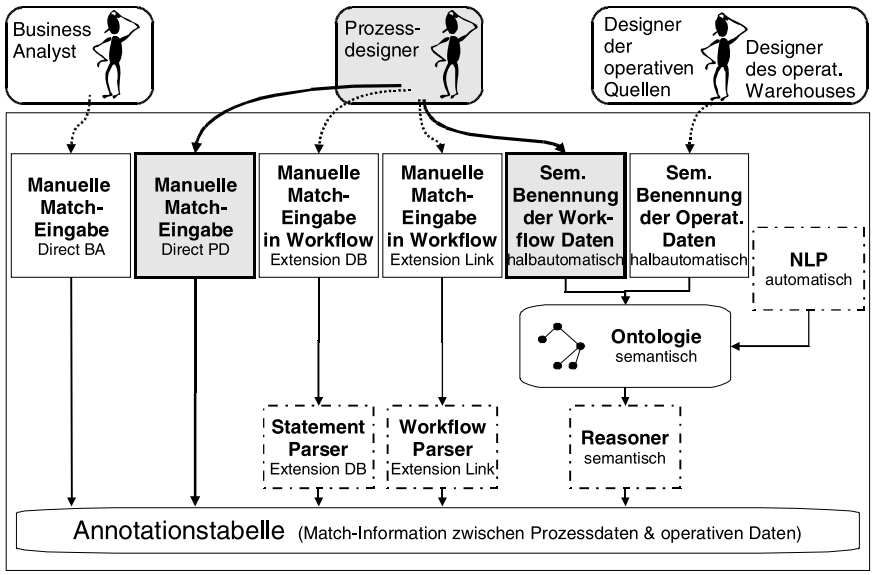

Abbildung 2: Architektur des Matching Frameworks

Das Framework bietet mehrere Alternativen für eine passende Match-Methode je nach Situation, z.B. abhängig davon, ob die genaue Verknüpfung dem Prozessdesigner bekannt ist oder er nur die semantische Bedeutung der Daten kennt. Die oberen Rechtecke im Framework in Abb. 2 zeigen den Namen der Methode und die verschiedenen Eingaben der Personen, die am Match beteiligt sind und auf die im Folgenden näher eingegangen wird. Grau hinterlegt sind die beiden Match-Methoden des Prozessdesigners (Direct PD und halbautomatisch), die durch den Annotationseditor, der in diesem Beitrag vorgestellt wird, realisiert und unterstützt werden. Eine allgemeine Übersicht über das gesamte Framework mit allen Methoden findet man in [RML08]. Gegebenenfalls sind für die Ausführung der Methoden im Framework neben den Benutzereingaben zusätzliche Tools nötig. Diese sind in Abb. 2 in gestrichelten Boxen dargestellt.

2.1 Business Analyst

Die aufwändigste Möglichkeit für eine direkte Verknüpfung ist ein manueller Match durch den Business Analysten (Direct BA). Hier müssen passende Matches zur Analysezeit gefunden werden, was aufgrund der riesigen Datenmengen im operativen DWH und im Audit Trail, in dem die Prozessdaten protokolliert werden, sehr zeitintensiv und fehleranfällig ist. Des Weiteren muss ein Matching der Daten gegebenenfalls bei jeder Analyse neu durchgeführt werden. Daher sind andere Methoden (siehe Abb. 2) zu bevorzugen.

2.2 Prozessdesigner

Dem Prozessdesigner stehen im Framework vier Alternativen zur Verfügung, die ihn beim Matching der Daten unterstützen. Er kann die konkrete Verknüpfung direkt angeben und abspeichern, wenn der Prozess fertig gestellt ist (Direct PD). Dazu gibt er zu jeder Prozessvariablen die passende operative Datenquelle an. In den nächsten Kapiteln wird diese Methode als Teil des Editors im Detail beschrieben.

Es kann aber auch sein, dass der Match schon Teil des Workflows ist. Dies ist der Fall, wenn der Workflow selbst um Anfragen auf operative Datenquellen erweitert wird, auf die sich der Workflow bezieht (Extension DB), weil die Anfrageergebnisse im weiteren Workflow benötigt und weiter verarbeitet werden müssen. Der Prozessdesigner muss also schon während des Prozessdesigns Datenbank-Statements entweder direkt im Workflowcode (z.B. IBM BPEL/SQL PlugIn [IBM07]) oder in einen Datenbankadapter (z.B. Oracle Database Adapter [VD07]) im Workflow einbetten, wodurch eine Verbindung zur jeweiligen operativen Datenquelle hergestellt wird. Der Match wird später durch einen Parser extrahiert.

Während des Prozessdesigns wird auch die nächste Match-Methode angewandt. Hier ergänzt der Prozessdesigner jede Prozessvariable im Prozess durch ein Attribut, das auf die betreffende operative Datenquelle verweist (Extension Link). Diese Methode verlangt jedoch eine Erweiterung der Workflowengine, die diese Attributserweiterung bei dem Deployment (Kompilieren des Workflows) erkennt und im Audit Trail abspeichert.

Die nächste Match-Alternative (halbautomatisch) eignet sich für die Annotation von Prozessvariablen, zu denen dem Prozessdesigner die entsprechende operative Datenquelle nicht bekannt ist. Er gibt zu jeder Variable die Referenz zu der passenden semantischen Beschreibung in einer Ontologie an. Damit die dazugehörige Datenquelle später automatisch gefunden werden kann, muss jedoch auch diese entsprechend semantisch annotiert sein. Das ist die Aufgabe des DWH-Designers und wird in Kapitel 2.3 behandelt.

2.3 Designer des Operativen Data Warehouses

Für eine halbautomatische Zusammenführung annotiert der Designer der operativen Datenquellen oder der DWH-Designer die Daten mit Hilfe einer Ontologie semantisch. Ein Reasoning Tool kann anschließend auf Basis der Annotationen und der Ontologie die Prozessdaten mit den operativen Daten matchen.

Für eine komplett automatische Verknüpfung wird ein Natural Language Processing Tool (NLP) benötigt, das in einer eingeschränkten Domäne zu jeder Variable und operativen Datenquelle abhängig von deren Namen das passende semantische Konzept findet. Die Annotation läuft so voll automatisch ab ohne Prozessdesigner und DWH-Designer.

3 Annotationsmechanismus

Um den manuellen Ansatz „Direct PD“ und den halbautomatischen semantischen Ansatz unseres Frameworks umzusetzen, ist ein Mechanismus zur Annotation der Workflowmodelle notwendig.

Für die Auswahl eines Mechanismus legen wir zu Grunde, dass die Workflowmodelle in der Sprache BPEL [Jo07] beschrieben und auf sogenannten BPEL-Engines ausgeführt werden. Diese Annahme ist plausibel, da BPEL sich über die letzten Jahre sowohl auf Anwender- als auch auf Herstellerseite als dominierender Standard für die Beschreibung von Workflows etabliert hat.

3.1 Annotierbare Elemente

Für die Auswahl eines Annotationsmechanismus muss zunächst geklärt werden, welche Daten überhaupt annotiert werden sollen. Ausgangsbasis sind dabei die Daten, die während der Ausführung des Workflows in die sogenannte Auditdatenbank geschrieben werden. Die Standardisierung der Datenformate in diesem Bereich (siehe [ZM04, Wm98]) ist noch nicht sehr weit fortgeschritten bzw. umgesetzt. Dennoch enthalten sämtliche Auditdatenbanken der von uns untersuchten BPEL-Engines für jedes Ereignis im Prozess (z.B. Aufruf eines Web Services, Aktualisierung einer Variable) zumindest die folgenden Daten:

(a) ID des Workflowmodells

(b) ID der Workflowinstanz

(c) Typ und die ID des Ereignisses

(d) Ein Zeitstempel für den Zeitpunkt des Ereignisses

(e) Name und Typ der BPEL-Aktivität, welche das Ereignis ausgelöst hat: Wenn die Aktivität einen Web Service aufgerufen hat, wird zusätzlich eine ID des Services angegeben.

(f) Name sowie der (neue) Wert der Variablen, wenn das Ereignis eine Variable verändert hat.

Die folgenden zwei Kriterien sind für die Auswahl der zu annotierenden Elemente zu beachten: Das erste Kriterium ist die *Verknüpfbarkeit*. Ziel unseres Frameworks ist die Verknüpfung von operativen Datenquellen und Prozessdaten. Eine Annotation ist daher nur für die Elemente sinnvoll, welche in irgendeiner Form sowohl in den operativen Daten als auch auf Prozessseite vorkommen. Das zweite Kriterium ist die *Nützlichkeit*. Ein Element ist im Sinne unserer Analyse nützlich, wenn es zusätzliche Einsichten und Erkenntnisse aus Geschäftssicht ermöglicht. Rein technische Steuerungsinformationen, wie sie z.B. für das Routing von Nachrichten zwischen Services verwendet werden, sind daher weniger interessant.

Die Anwendung beider Kriterien ergibt, dass sämtliche Elemente bis auf die Variablen von der Annotation ausgeschlossen werden können. Im Wesentlichen geht dies auf das Verknüpfbarkeitskriterium zurück, da alle anderen Elemente auf Seite der operativen Datenquellen kein entsprechendes Gegenstück aufweisen.

Variablen sind die Datencontainer eines BPEL-Prozesses. Sie bilden Input- und Outputdaten des Prozesses sowie mögliche Zwischenergebnisse ab. Ihre Struktur entspricht entweder einem XML Schema Type oder einer WSDL Message. XML Schema ist ein Standard zur Beschreibung der Datenstruktur von XML-Dokumenten. WSDL Messages sind in WSDL 1.1 definierte Nachrichten, die von und mit Web Services ausgetauscht werden. Die Annotation der Variablen kann daher auf drei verschiedenen Ebenen erfolgen
(siehe Abb 3):

Man kann die Variable „InputVariable“ in (1) selbst annotieren oder Teile einer Message („parameters“ in (2)). Falls die Variable komplex mit XML Schema Types (3) aufgebaut ist, ist ebenso eine Annotierung der einzelnen Elemente, z.B. der „EmployeeID“ möglich. Wir haben uns dafür entschieden, es dem Benutzer zu ermöglichen, die Variablen bis auf die Ebene der XML-Schemastruktur zu annotieren. Die Annotation der Variable selbst ist zwar einfach umzusetzen, bringt aber wenig neue Erkenntnisse. Ähnliches gilt für die Message Parts. Hier kommt das Problem hinzu, dass das Message-Konzept mit der Einführung von WSDL 2.0 entfernt wurde [Ch07].

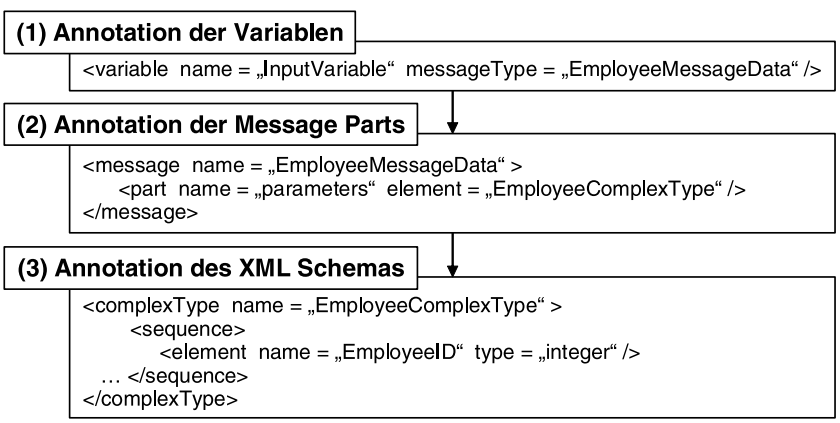

Abbildung 3: Annotationstiefe bei Variablen

3.2 SAWSDL

Für die Annotation von Web Services und ihrer Daten haben sich in den letzten Jahren im Rahmen des gestiegenen Interesses am Semantic Web eine Reihe von Standards etabliert. Auf der einen Seite stehen „heavy-weight“ Ansätze, die eine komplett neue konzeptuelle semantische Repräsentation eines Web Service bereitstellen. Die prominentesten Vertreter dafür sind WSML (Web Service Modeling Language) [Br05] und OWL-S (Web Ontology Language for Web Services) [Ma04].

Auf der anderen Seite stehen Standards wie SAWSDL (Semantic Annotations for WSDL) [FL07], die lediglich die Beschreibungssprache WSDL um zusätzliche Elemente erweitern. Dieser Ansatz ist für unser Framework besser geeignet als z.B. WSML oder OWL-S. Wir wollen fokussiert bestimmte Elemente einer WSDL- oder XML-Schema-Datei beschreiben und nicht die Semantik eines gesamten Services erfassen.

Daher werden wir im Folgenden auf die verschiedenen Elemente von SAWSDL und ihrer Verwendung für die Umsetzung der direkten und semantischen Datenverknüpfung eingehen. Attribute, die nicht Teil des SAWSDL-Standards sind und von uns eingeführt werden, sind mit einem „*“ markiert.

Semantische Verknüpfung

- modelReference: Das „modelReference“-Attribut wird verwendet, um XML-Elemente mit Konzepten einer Ontologie zu verknüpfen. SAWSDL schreibt keine spezielle Struktur für diese Referenz vor, es hat sich aber die folgende Konvention eingebürgert, um das Konzept in der Ontologie zu adressieren:

  <Ontologie ID> “#” <Konzept ID>

- liftingSchemaMapping und loweringSchemaMapping: Diese Attribute ermöglichen es, eine Datei zu referenzieren, die die Transformation von XML in die Ontologiesprache (liftingSchemaMapping) bzw. in die andere Richtung (loweringSchemaMapping) realisiert.

- modelType*: Zur Unterscheidung von verschiedenen Ontologiesprachen haben wir dieses Attribut eingeführt. Es vereinfacht die Arbeit eines Parsers bei dem automatischen Matching und verbessert die Lesbarkeit.

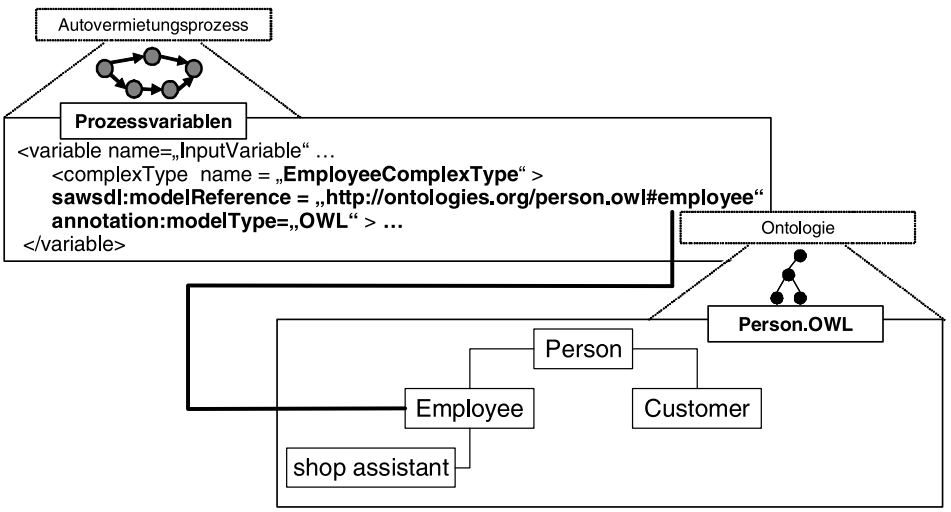

Abbildung 4: SAWSDL für semantische Verknüpfung

Abb. 4 zeigt beispielhaft, wie ein XML-Element „EmployeeComplexType“ der „InputVariable“ mit dem Konzept „Employee“ der OWL-Ontologie „person.owl“ verbunden wird.

Direkte Verknüpfung

– source*: Das neu eingeführte „source“-Attribut ermöglicht es, ein XML-Element direkt mit einer Entität in einer operativen Datenquelle zu verknüpfen und so den „Direct PD“-Ansatz zu realisieren. Es hat die folgende Syntax:

  <ID der Datenquelle> “#” <ID der Entität>

  Für den Fall, in dem die Datenquelle eine relationale Datenbank ist, nimmt das Attribut die folgende Form an:

  <JDBC Verbindungs-URL> “#” <Schemaname> “.”
  <Tabellenname> [“.” <Spaltenname>]

– sourceType*: Analog zu modelType wird sourceType verwendet, um zwischen verschiedenen Arten von Datenquellen zu unterscheiden, z.B. relationale Spalten und Tabellen, XML-Daten oder Spalten in einer CSV-Textdatei (Character Separated Values).

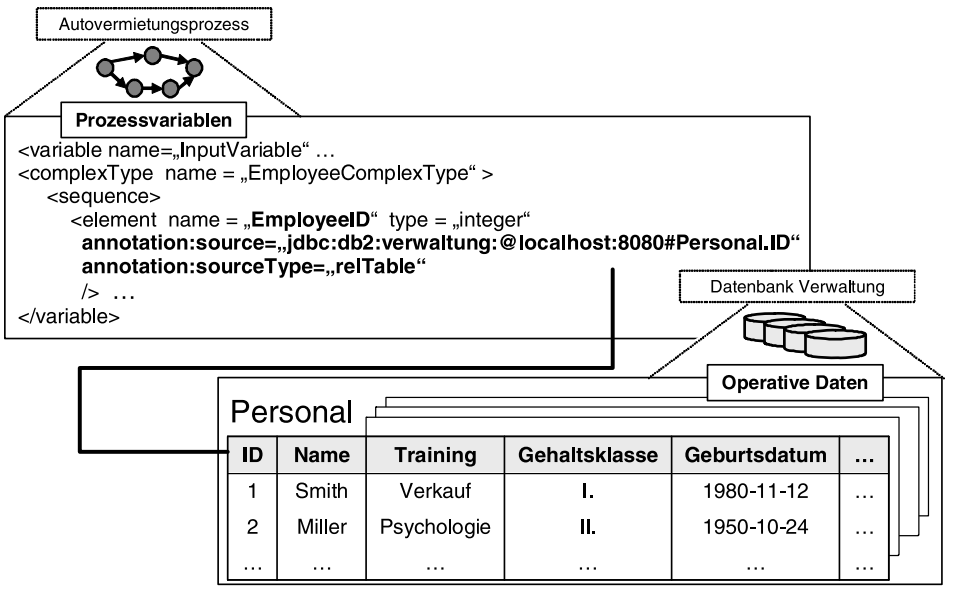

Abbildung 5: SAWSDL-Erweiterung für direkte Verknüpfung

In Abb. 5 wird die SAWSDL-Erweiterung „source“ aus unserem neuen Namespace „annotation“ verwendet, um das Element „EmployeeID“ der Variable „InputVariable“ mit der Spalte „ID“ aus der Tabelle „Personal“ in einer relationalen Datenbank namens „Verwaltung“ zu verknüpfen.

4 Annotationseditor

Zur Umsetzung der Annotationsmethode stellen wir einen Editor bereit. Der Editor unterstützt einen Prozessdesigner dabei, BPEL-Prozesse zu laden, zu annotieren und die annotierten Daten in angemessener Form abzulegen. Des Weiteren ermöglicht er es, XML-Schema- und WSDL-Dokumente direkt in ihrer Datei zu annotieren und abzuspeichern. Wie Abb. 6 zeigt, orientiert sich der Aufbau des Editors an gängigen Entwicklungsumgebungen und ähnlichen Werkzeugen.

Der Projektexplorer (1) zeigt die aktuell geladenen Prozesse und Dokumente an. Wie in der Abbildung zu sehen ist, ist aktuell nur der BPEL-Autovermietungsprozess „Rental-Process“ in den Workspace des Editors geladen.

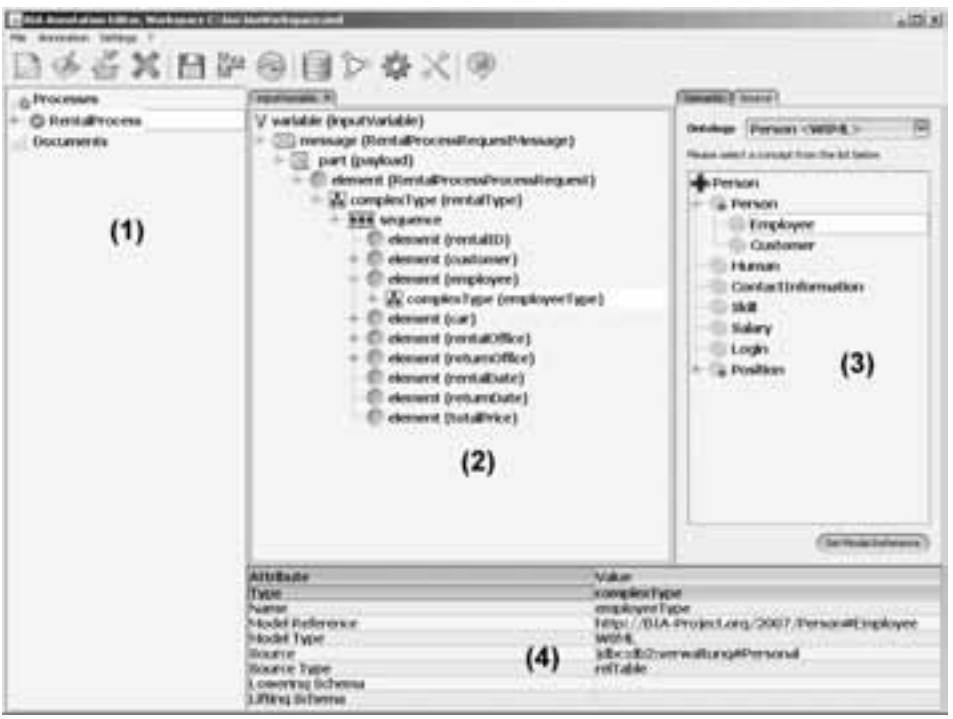

Abbildung 6: Screenshot des Editors

Das zentrale Element des Editors ist der Annotationsexplorer (2), welcher die Struktur von BPEL-Variablen sowie WSDL- und XML-Schema-Dateien visualisiert. In der Abbildung ist die BPEL-Variable „InputVariable“ geöffnet und das Element „employee“ ausgewählt. Der Benutzer kann einzelne Variablenelemente auswählen und entweder über die Attributstabelle (4) oder den graphischen Editor (3) für Ontologie- und Datenverknüpfung annotieren. Die Attributstabelle zeigt die textuellen Werte der Annotationsattribute an. Es ist möglich, mit Hilfe der Tabelle die Werte der Annotationsattribute manuell zu verändern. Meist ist es jedoch einfacher, dafür den graphischen Editor zu verwenden.

Der Editor ist unter Java Version 1.5 implementiert. Zur Erbringung seiner Funktionalität verwendet er eine Reihe von APIs wie z.B. WSDL4J [Ws06] bzw. Apache Woden [Ap07] für das Laden der WSDL1.1- bzw. der WSDL2.0-Beschreibun- gen oder Jena [Je07] bzw. WSMO4J [Ws07] für das Laden und Visualisieren der OWL- bzw. WSML-Ontologien.

Unterstützt werden BPEL-Prozesse für IBM Process Server [IBM07] und Oracle SOA Suite [Or07]. Die Architektur des Editors bietet Erweiterungspunkte, die die Unterstützung weiterer Engines und weiterer operativer Datenquellen, wie z.B. XML, relativ problemlos zu einem späteren Zeitpunkt erlauben.

5 Ganzheitliche Analyse in der Praxis

Dieses Kapitel zeigt anhand eines Fallbeispiels, wie der Editor für das Zusammenführen von Daten verwendet wird, um später eine ganzheitliche Analyse im Unternehmen durchführen zu können. In dem Praxisbeispiel will eine Autovermietungsfirma ihre Geschäftsprozesse optimieren. Dazu werden hier die Prozesse zur Vermietung eines Autos an einen Kunden herausgegriffen und betrachtet. Viele Prozessdaten sind für die Analyse von Belang, uns interessieren aber besonders die Prozessvariablen, die mit Daten aus operativen Systemen verknüpft werden, um durch die Analyse zusätzliche Informationen zu generieren. Der Annotationseditor erleichtert die Zusammenführung, die in Abb. 7 schematisch dargestellt ist.

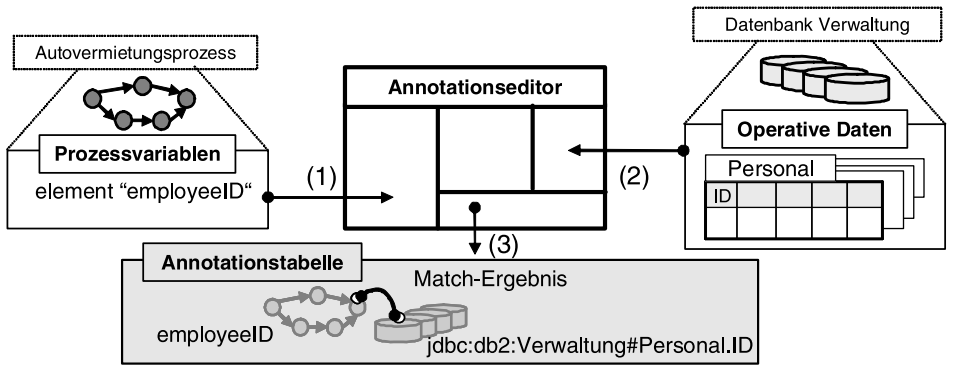

Abbildung 7: Ganzheitliche Analyse basierend auf direktem Match

In (1) werden die Variablen aus dem BPEL-Autovermietungsprozess, u.a. auch die „EmployeeID“ des Angestellten, der die Autovermietung abgeschlossen hat, in den Annotationseditor geladen. Operative Daten aus anderen Systemen, wie hier aus der relationalen Verwaltungsdatenbank mit der Tabelle „Personal“ aus der Personalabteilung werden in (2) in den Editor übertragen. Der Prozessdesigner wählt nun das Element „EmployeeID“ der Prozessvariablen „InputVariable“ aus und verknüpft es mit der Spalte ID aus der Tabelle Personal. Die Verknüpfung wird in (3) als Match-Ergebnis in eine Annotationstabelle abgelegt.

Diese Match-Tabelle erleichtert es dem Analysten später, die zusammengehörigen Verbindungen zu finden, sodass er die OLAP-Anfrage zügiger erstellen kann. Wenn der Analyst die OLAP-Anfrage erstellt (4), kann er im Match-Ergebnis nachschauen, welche Variablen zu welchen operativen Daten passen und für eine ganzheitliche Analyse miteinander verbunden werden müssen (Join-Operator). Mit der OLAP-Anfrage können z.B. die Eigenschaften der Angestellten ermittelt werden, die dazu führen, dass ein Mietprozess erfolgreich abgewickelt wird und die im Gegensatz dazu jenem Personal fehlen, deren bearbeitete Prozesse abgebrochen werden. In diesem Beispiel werden hierfür die Eigenschaften Training, Gehalt und Alter der Angestellten untersucht und mit den Prozessdaten aus dem Audit Trail verknüpft, um den Prozessstatus zu erhalten. Der Audit ist hier beispielhaft als eine Tabelle dargestellt. Abhängig vom Workflow-Management-System und seiner Protokollierung ist dies evtl. eine komplexe Subquery. Zur Analyse werden die Personaleigenschaften in verschiedenen Gruppierungen dargestellt (Rollup-Operator). Unsere zukünftigen Forschungsarbeiten beschäftigen sich ausführlicher mit diesen OLAP-Anfragen.

Gegebenenfalls wird nun aus der Analyse ersichtlich, dass die Eigenschaft, psychologisch geschult zu sein, ein positives Ergebnis auf den Ablauf der Mietprozesse hat. Eine Optimierung der Geschäftsprozesse erfolgt dann dadurch, dass die Aktivitäten zur Kundenberatung im Mietprozess hauptsächlich von psychologisch trainierten Mitarbeitern ausgeführt werden und zudem eine Psychologie-Fortbildung auch den anderen Kollegen angeboten wird.

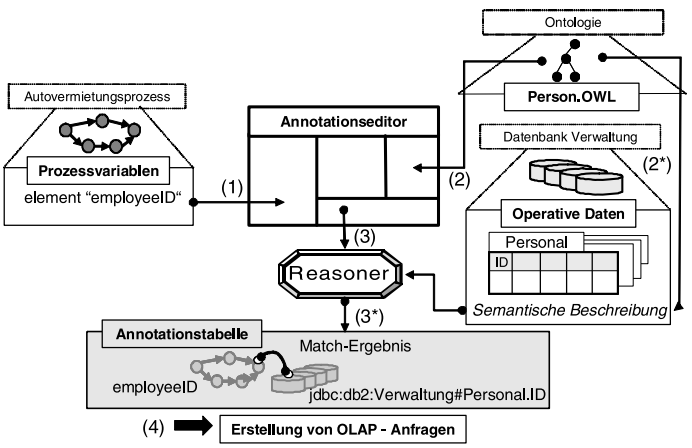

Abbildung 8: Ganzheitliche Analyse basierend auf semantischem Match

Für die semantische Annotation in Abb. 8 wird in (2) statt der operativen Daten, eine Ontologie geladen. So wird durch den Editor die Variable mit einem semantischen Konzept versehen. Das komplexe Element „Employee“ der Variable „InputVariable“ referenziert das OWL-Konzept „Employee“ und wird so in (3) in die Annotationstabelle abgelegt. Auch die operative Datenquelle muss mit einem Editor semantisch annotiert werden (2*). Schließlich findet ein Reasoner automatisch die passenden Verknüpfungen, die in (3*) als Match-Ergebnis abgespeichert werden. So kann auch mit Hilfe dieses Matches in (4) eine OLAP-Anfrage zwecks Analyse formuliert werden.

6 Evaluation

Dieses Kapitel analysiert den Nutzen einer Verknüpfung zwischen Prozessvariablen und operativen Daten und inwiefern der Annotationseditor dabei nützlich ist. Dazu werden zunächst die hier verwendeten Match-Methoden im Allgemeinen und danach der Editor selbst evaluiert. Da wir zu einem so frühen Zeitpunkt noch über keine empirischen Studien verfügen, stützt sich die Evaluierung auf eine von uns entwickelte Fallstudie und qualitative Überlegungen. Anschließend erfolgt eine Analyse des betriebswirtschaftlichen Nutzens des Editors und der Match-Methoden.

6.1 Evaluation der Match-Methoden

In den vorherigen Kapiteln wurden beide Match-Methoden, das direkte Matching und das semantische Matching, ausführlich erläutert. Sie werden in unterschiedlichen Situationen verwendet. Je nach Fachwissen des Designers ist die direkte oder semantische Methode schneller und effektiver. Das direkte Verknüpfen der Prozessdaten mit operativen Daten funktioniert sehr effizient, wenn der Prozessdesigner sich auch bei den operativen Datenquellen auskennt. Dies ist nützlich bei einem Match zwischen Entitäten, deren Daten ähnlich strukturiert sind, z.B. um ihre Schlüsselwerte miteinander zu verbinden (siehe Abb. 8 aus Kapitel 5).

Sobald komplexere Transformationen für das Matching der Daten notwendig sind oder dem Designer die zu matchenden operativen Datenquellen unbekannt sind, ist das semantische Matching effektiv. Dafür müssen jedoch auch die operativen Datenquellen von dem operativen DWH-Designer semantisch annotiert vorliegen. Außerdem ist das Vorhandensein eines Reasoners und einer Ontologie erforderlich, in deren Domäne sich der Designer für eine genaue Annotation der Daten gut auskennen muss. Das Matching der Daten ist auf diese Domäne beschränkt im Gegensatz zum direkten Matching, bei dem die Daten vom Designer aus verschiedenen Domänen verknüpft werden können. Durch die Bereitstellung dieser Ontologie entsteht zwar ein zusätzlicher Aufwand. Dieser fällt vor dem Hintergrund aktueller Entwicklungen jedoch nicht sehr stark ins Gewicht. Der Trend geht nämlich in die Richtung semantischer Web Services [CS06]. Daher bekommen entsprechende semantische Technologien gerade großen Auftrieb. Semantische Annotationen mit SAWSDL werden standardisiert und für die Analyse wieder verwertbar.

6.2 Evaluation des Editors

Die Annotationen für die beiden Match-Methoden können mit dem Editor effektiv umgesetzt werden. Der Editor befolgt die verschiedenen Richtlinien für eine graphische Schnittstelle zur Mensch-Computer-Interaktion aus dem ISO Standard [ISO06] und erleichtert so dem Benutzer seine Bedienung:

- *Selbstbeschreibungsfähigkeit*: Der Editor ist selbsterklärend. Der Benutzer erhält zudem Informationen zu Icons mit Hilfe von Tooltips und zusätzliche Textinformation während der Ausführung längerer Dialogfunktionen. Operative Datenquellen bzw. Ontologien werden intuitiv als Baumstruktur dargestellt. Alle Funktionen sind konsistent aufgebaut.
- *Aufgabenangemessenheit*: Nur für die Annotation wichtige Informationen werden im Editor dargestellt. Dadurch wird eine Reizüberflutung des Benutzers vermieden.
- *Steuerbarkeit*: Der Benutzer hat die Kontrolle über den Editor und kann jederzeit eine Funktion abbrechen.
- *Individualisierbarkeit*: Bestimmte Einstellungen sind durch den Benutzer konfigurierbar. Dazu gehören z.B. das automatische Laden von Ontologien und Abspeichern der Annotationen. Außerdem können die Teile der Prozessvariablen angegeben werden, die man im Editor annotieren darf.
- *Fehlerrobustheit*: Fehlerhafte Eingaben werden möglichst vermeidbar gemacht über die Dialoggeschlossenheit der Funktionen und das Deaktivieren von Funktionen, die zu dem jeweiligen Zeitpunkt nicht ausführbar sind.
- *Erlernbarkeit*: Zwar ist die Benutzung des Editors insgesamt selbsterklärend, als zusätzliche Unterstützung sind aber einige Tutorials vorhanden.
- *Erwartungskonformität*: Der Editor ist auf die Bedürfnisse des Prozessdesigners für das Matching der Daten ausgerichtet.

6.3 Betriebswirtschaftliche Nutzenpotentiale

Abb. 9 fasst die mit der Annotationsmethode und dem Ausführen einer ganzheitlichen Analyse verfolgten Nutzenpotentiale zusammen. Das ganzheitliche Analyseverfahren ermöglicht es einem Unternehmen, die Effizienz und Effektivität seiner Prozesse zu steigern. Indem die Annotationsmethode und der Editor Kosten-, Zeit- und Qualitätseffekte positiv beeinflussen, wird der Einsatz des Analyseverfahrens erst praktikabel („Enabler“).

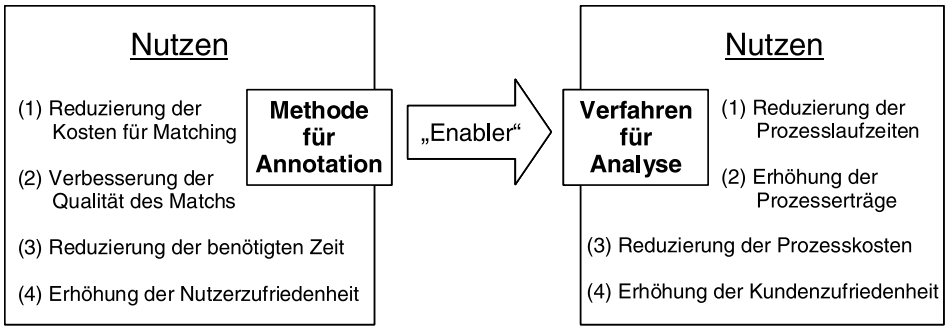

Abbildung 9: Nutzen der Annotationsmethode und der ganzheitlichen Analyse

Im Folgenden werden die Nutzenpotentiale der Methode und des Editors im Detail und analog zu Abb. 9 vorgestellt, die sich aus seiner Verwendung durch den Prozessdesigner gegenüber einer manuellen Annotierung zur Analysezeit oder ohne Editor ergeben:

(1) Reduzierung der Kosten für Matching: Der Aufwand und damit die Kosten des Personals für die Erstellung der Verknüpfung verringern sich durch die gut bedienbare graphische Benutzeroberfläche. Zudem muss der Match für die Variablen nur einmal pro Workflowmodell erfolgen und nicht für jede Analyse aufs Neue. Durch die Verwendung von SAWSDL im Editor ergeben sich noch weitere Kostenreduzierungen, da bereits erstellte Annotationen im Bereich der semantischen Web Services für die Analyse wieder verwendet werden können. Außerdem müssen bereits früher mit dem Editor erstellte semantische Annotationen nicht an neue operative Datenquellen angepasst werden, woraus sich weitere Einsparungen ergeben. Die Kosten zur Analysezeit verringern sich durch die Verwendung der direkten und semantischen Matches erheblich.

(2) Verbesserung der Qualität des Matches: Die Fehlerquote der direkten Verknüpfung zwischen Prozessdaten und operativen Daten sinkt. Bei dem semantischen Match können zudem Beziehungen gefunden werden, die einem Menschen nicht sofort aufgefallen wären. Eine standardisierte Ausgabe des Matches durch den Editor ermöglicht eine effektive Weiterverarbeitung durch den Analysten. Weiterhin erhöht die plattformübergreifende Nutzung verschiedener Workflow-Management-Systeme, Ontologiesprachen und beliebiger operativer Datenbanken die Qualität des Matches.

(3) Reduzierung der benötigten Zeit: Alle Faktoren, die bei den vorherigen Punkten eine Rolle spielen, können auch für die Zeitersparnis angeführt werden. Die graphische Benutzeroberfläche verkürzt die Zeit, die für eine Annotation benötigt wird. Die Analysten wiederum können effizienter und effektiver ihre Analysen erstellen, da sie bereits eine qualitativ wertvolle und standardisierte Verknüpfung vorliegen haben und nicht selbst danach suchen müssen. Dadurch ergibt sich insgesamt eine schnellere Analyse der Prozesse.

(4) Erhöhung der Nutzerzufriedenheit: Eine gute Handhabung des Editors (siehe auch Kapitel 6.2) und ein schnelles und zuverlässiges Match-Ergebnis erhöhen die Bereitschaft des Nutzers diesen Editor zu verwenden.

Die Nutzenfaktoren Zeit- und Kostenersparnis machen es sinnvoll, den Editor für die Verknüpfung zu verwenden. Die Erhöhung der Nutzerzufriedenheit führt dazu, dass der Editor auch wirklich benutzt wird und letztendlich die Annotationen für einen Match überhaupt erst erstellt werden, was ohne den Editor selten der Fall wäre. Ansonsten bliebe die Erstellung des Matches die Aufgabe der Analysten, was aufgrund des hohen Zeitaufwandes in der Regel nicht durchgeführt wird. Der Editor macht also ein ganzheitliches Analyseverfahren erst möglich.

Das Nutzenpotential, das sich aus dem Analyseverfahren und der daraus resultierenden Prozessoptimierung ergibt, ist nachfolgend aufgelistet:

(1) Reduzierung der Prozesslaufzeiten: Durch die Verknüpfung der Prozessdaten mit operativen Daten können Faktoren erkannt werden, die die Laufzeit eines Prozesses negativ beeinflussen. Mit der Erstellung neuer Prozessregeln, die diese Eigenschaften umgehen oder alternativ bearbeiten, kann die Laufzeit gesteigert werden.

(2) Erhöhung der Prozesserträge: Prozesse, die abgebrochen wurden oder nur wenig Gewinn einbringen, werden in Verbindung mit den operativen Datenquellen ganzheitlich analysiert und optimiert.

(3) Reduzierung der Prozesskosten: Durch die Reduzierung der Prozesslaufzeit werden oft auch die Prozesskosten reduziert. So können z.B. die Kosten der im Prozess eingesetzten Ressourcen wie Maschinen und Mitarbeiter optimiert werden.

(4) Erhöhung der Kundenzufriedenheit: Die ganzheitliche Analyse erlaubt es, Kundenbedürfnisse und -wünsche schneller zu erkennen und in die Geschäftsprozesse zu integrieren. Dies erhöht die Kundenzufriedenheit und kann Kunden langfristig an das Unternehmen binden.

7 Verwandte Arbeiten

In der Welt der Datenbanken liegen Methoden zur Informationsintegration im Fokus der Forschung. Dies betrifft sowohl die direkte Zusammenführung von Schemata [LN06] als auch das Matching auf Basis der semantischen Beschreibung der Datenquellen [DH05] mit dem Ziel, einen einheitlichen Zugriff auf unterschiedliche operative Datenquellen zu schaffen. Auch unsere Methoden zielen auf eine Schemaintegration ab. Jedoch betrachten wir eine andere Ebene der Integration. Die operativen Daten sollen mit Variablen gematcht werden, die von Workflows verwendet werden, also in einem ganz anderen Umfeld vorkommen als die operativen Daten. Zudem unterscheiden sich der Inhalt und die Struktur eines Audit Trails, in dem die Ablaufdaten des Workflows gespeichert werden, grundlegend von den operativen Datenquellen, was bei der Zusammenführung ebenso eine Rolle spielt.

Auf der Seite der Geschäftsprozesse beschreiben [CS06] und [He05] semantische Technologien mit dem Ziel einer automatischen Web-Service-Komposition. Dazu muss der Web Service als Ganzes beschrieben werden, während in unserer Arbeit vor allem eine detaillierte semantische Variablenannotation von Belang ist vor dem Hintergrund einer ganzheitlichen Analyse.

Mit WSMO Studio [Ws08] oder Radiant aus dem Meteor-S Projekt der Universität Georgia [Me05] existieren Editoren, die die semantische Annotation von WSDL-Dokumenten unterstützen. Beide Werkzeuge sind an bestimmte Ontologien gekoppelt, entweder an WSML-Ontologien im ersten Tool oder an OWL im zweiten. Die jeweils andere Ontologie kann graphisch nicht eingebunden werden. Auch IBM redet von einem semantischen PlugIn [IBM06] für den WebSphere Integration Developer. Die Benutzeroberfläche für die Annotation ist dann jedoch nur textuell und nicht graphisch ausführbar. Bei allen Editoren fehlt außerdem die direkte Integration mit operativen Datenquellen.

Für die direkte Verknüpfung von Prozessdaten und operativen Daten zur Prozessdesignzeit steht bisher außer unserem Editor mit der Methode „Direct PD“ keine Annotationsmethode zur Verfügung. Vorhandene Ansätze führen die Daten erst zur Analysezeit manuell zusammen [ZM04; Ca07]. Ein weiterer Ansatz von [ZM04] fügt künstliche Aktivitäten in den Workflow ein, die dann das Matching übernehmen. Die zusätzlichen Aktivitäten verfälschen jedoch die Semantik und evtl. sogar die Ausführungszeit der Prozesse, da sie das Workflowmodell verändern.

8 Zusammenfassung und Ausblick

Dieser Beitrag stellt zwei Methoden vor, wie Prozessdaten und operative Daten zusammengeführt werden können, um eine ganzheitliche Unternehmensanalyse zu ermöglichen. Die zwei Ansätze erlauben es, BPEL-Variablen mit einer Erweiterung von SAWSDL zu annotieren. Die Annotation zeigt auf die operative Datenquelle, die sich auf dasselbe reale Objekt bezieht wie die Prozessvariable. Der Link kann vom Prozessdesigner entweder direkt spezifiziert werden oder er referenziert ein Konzept aus einer Ontologie, das diese Variable semantisch beschreibt, um den Match später durch einen Reasoner herzustellen.

Die Evaluierung der beiden Methoden und ihre Betrachtung in einem globalen Fallbeispiel haben gezeigt, dass beide Methoden je nach Situation, in der sie eingesetzt werden, sehr hilfreich für den Prozessdesigner sind und einen großen Nutzen für die Optimierung der Geschäftsprozesse und damit für ein Unternehmen erzielen. Mit Hilfe eines Editors hat der Prozessdesigner die Möglichkeit, die Methoden effektiv und unter minimalem Lernaufwand anzuwenden. Bisher können damit BPEL-Variablen für die BPEL-Engines IBM Process Server und Oracle SOA Suite annotiert werden. Es werden ein direkter Match zu den meisten relationalen Datenbanken und eine semantische Annotation durch OWL- und WSML-Ontologien für einen halbautomatischen Match unterstützt.

Zurzeit beschäftigen wir uns mit der Umsetzung der ganzheitlichen Analyse. Die Matches müssen dazu zwischen Instanzdaten im Audit Trail und dem operativen DWH durchgeführt werden, damit ein Verbund zwischen Prozessdaten und operativen Daten neue und wertvolle Analyseergebnisse liefern kann. Hierfür müssen für die halbautomatische semantische Zusammenführung noch die operativen Datenquellen semantisch annotiert und ein Reasoning Tool bereitgestellt werden.

9 Literaturverzeichnis

[Ap07] Apache Woden: WSDL2.0 API für Java. Version 1.0, Juli 2007. Verfügbar unter: http://ws.apache.org/woden[1]

[Br05] Bruijn, J. et.al.: Web Service Modeling Language (WSML), W3C Member Submission, 2005. Verfügbar unter: http://www.w3.org/Submission/WSML/[1]

[CS06] Cardoso, J.; Sheth, A.: Semantic Web Services, Processes and Applications. Springer 2006.

[Ca07] Casati, F. et.al.: A Generic solution for Warehousing Business Process Data. In Proc. of 33rd Int. Conf. on Very Large Data Bases, Österreich 2007.

[Ch07] Chinnici, R. et.al.: Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. W3C Recommendation, 2007. Verfügbar unter: http://www.w3.org/TR/wsdl20/[1]

[DH05] Doan, A.; Halevy, A.: Semantic Integration Research in the Database Community: A Brief Survey. AI Magazine 2005.

[FL07] Farrell, J.; Lausen, H.: Semantic Annotations for WSDL and XML Schema. W3C Recommendation, 2007. Verfügbar unter: http://www.w3.org/TR/sawsdl[1]

[He05] Hepp, M. et.al.: Semantic Business Process Management: Using Semantic Web Services for Business Process Management. In Proc. of IEEE Int. Conf. on e-Business Engineering, China 2005.

[IBM06] IBM: Semantic Tools for Web Services. Verfügbar unter: http://www.alphaworks.ibm.com/tech/wssem[1]

[IBM07] IBM: IBM WebShpere Process Server Version 6.0.2. Verfügbar unter: http://www-306.ibm.com/software/integration/wps/[1]

[ISO06] International Organization for Standardization: ISO Norm 9241-110 – Ergonomics of human-system interaction -- Part 110: Dialogue principles, 2006. Verfügbar unter: http://www.iso.org/iso/home.htm[1]

[Je07] Jena: A Semantic Web Framework for Java. Version 2.5.5, Januar 2007. Verfügbar unter: http://jena.sourceforge.net[1]

[Jo07] Jordan, D. et.al.: Web Service Business Process Execution Language 2.0: Oasis Standard. April 2007. Verfügbar unter: http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.pdf[1]

[1] Letzter Zugriff auf aufgeführte Links am 14.05.2008

[LN06] Leser, U.; Naumann, F.: Informationsintegration – Architekturen und Methoden zur Integration verteilter und heterogener Datenquellen. Dpunkt Verlag, 2006.

[Ma04] Martin, D. et.al.: OWL-S: Semantic Markup for Web Services. W3C Member Submission, 2004. Verfügbar unter: http://www.w3.org/Submission/OWL-S/[1]

[Me05] Meteor-S: Semantic Web Services and Processes. Verfügbar unter: http://lsdis.cs.uga.edu/projects/meteor-s[1]

[Or07] Oracle: Oracle SOA Suite Version 10.1.3.1, 2007. Verfügbar unter: http://www.oracle.com/technologies/soa/soa-suite.html[1]

[RML08] Radeschütz, S.; Mitschang, B.; Leymann, F.: Matching of Process Data and Operational Data for a Deep Business Analysis. In (Mertins, K. et.al. Hrsg.): Proc. 4th Int. Conf. on Interoperability for Enterprise Software and Applications, Berlin 2008.

[VD07] Vasudevan, S.; Dave, R.: Oracle Application Server Adapters for Files, FTP, Databases, and Enterprise Messaging User's Guide (10.1.3.1.0), 2007.

[We05] Weerawarana, S. et.al.: Web Services Platform Architecture: Soap, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging and More. Prentice Hall, New Jersey, USA 2005.

[Wm98] Workflow Management Coalition: Audit Data Specification. 1998. Verfügbar unter: http://www.wfmc.org/standards/docs/TC-1015_v11_1998.pdf [1]

[Ws06] WSDL4J: WSDL1.1 API für Java. Version 1.6.2, November 2006**.** Verfügbar unter: http://sourceforge.net/projects/wsdl4j[1]

[Ws07] WSMO4J: WSML API für Java. Version 0.6.1, Juni 2007. Verfügbar unter: http://wsmo4j.sourceforge.net/[1]

[Ws08] WSMO Studio: Semantic Web Service Environment for WSMO. Verfügbar unter: http://www.wsmostudio.org/[1]

[ZM04] zur Mühlen, M.: Workflow based controlling. Foundation, Design and Application of Workflow-driven Process Information Systems. Logos Verlag, Berlin 2004.