

Diplomarbeit

**Anwendung von
Process-Mining-Ansätzen auf
Log-Daten aus SAP-Systemen**

Igor Blyufshteyn
Dezember 2011

Gutachter:
Prof. Dr. Jan Jürjens
Prof. Dr. Jakob Rehof

Technische Universität Dortmund
Fakultät für Informatik
Lehrstuhl 14 Software Engineering
<http://ls14-www.cs.tu-dortmund.de>

In Kooperation mit:
Fraunhofer-Institut für
Software- und Systemtechnik

Kurzdarstellung

Moderne Informationssysteme, wie das SAP-ERP-System, sind heutzutage nicht mehr aus den Unternehmen wegzudenken. Dort führen sie u.a die für das Unternehmen relevanten Geschäftsprozesse aus. Dabei wird eine große Menge an verschiedenen Arten von Informationen gespeichert, die für die Erreichung vielseitigen Ziele verwendet werden können. Eine Bestrebung ist das Analysieren von den Geschäftsprozessen aus verschiedenen Blickwinkeln. Ein etabliertes Verfahren zur Analyse von Prozessen bildet die Process-Mining-Technologie. Diese besteht aus verschiedenen Algorithmen, die Wissen aus den gespeicherten Informationen extrahieren und analysieren. Diese Arbeit befasst sich mit dem Extrahieren von Informationen aus einem SAP-ERP-System zur Erzeugung einer Log-Datei, die anschließend mit den Process-Mining-Methoden analysiert werden kann. Die Vorverarbeitung der Daten für die Analyse ist komplex. Die Schwierigkeit liegt darin, dass das System keine Log-Datei erzeugt und die Informationen in einem Rohzustand in einer Datenbank zur Verfügung stehen. Untersucht wird neben dem Gebrauch in der Praxis an einem SAP-ERP-System auch die Qualität der Analyse auf synthetischen Daten.

Abstract

Nowadays, modern information systems such as the SAP ERP system are an indispensable part of the company. They perform there - among other things - business processes that are relevant for the company. Hence, different pieces of information are being stored and can be used in order to achieve various goals. The point is to analyze the business processes from different perspectives. One established method of evaluating the business processes is the process mining technology. It consists of various algorithms that extract and assess data saved in the information source. This Master thesis deals with extracting pieces of information from a SAP ERP system in order to generate a log file which can be analyzed with the use of process mining methods. The preprocessing of data for the analysis is complex. The difficulty is that the system does not create a log file and only the raw information is available in a database. Apart from the practical use of the SAP ERP system, the analysis quality of any synthetic data will be addressed in this work.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation und Hintergrund	1
1.2	Verwandte Arbeiten	5
1.3	Aufbau der Arbeit	10
2	Grundlagen	11
2.1	Enterprise Resource Planning mit SAP ERP	11
2.1.1	Konzept	12
2.1.2	Das SAP-Modell-Unternehmen	14
2.2	Process Mining	15
2.2.1	Überblick	15
2.2.2	Ereignis-Log	16
2.2.3	XES-Format	18
2.2.4	ProM-Vorstellung	19
2.3	SAP ERP und Process Mining	20
2.3.1	Diskussion	20
2.3.2	SAP Referenzmodell	23
2.3.3	Log-Daten	26
3	Realisierung	27
3.1	Ziel	27
3.2	Vorbereitungsphase	28
3.3	Anbindung an SAP	30
3.3.1	Schnittstellentechnologien	30
3.3.2	SAP Java Connector	34
3.3.3	Datenextraktion	36
3.4	Anbindung an ProM	44
3.5	Benutzeroberfläche	47
3.5.1	Allgemeine Einstellungen	47
3.5.2	Filter	48
3.5.3	Visualisierung	49
3.6	Fazit	50
4	Fallstudie	52
4.1	Beschaffungsprozess	52
4.2	Durchführung	56

4.3	Ergebnisse	63
4.4	Fazit	70
5	Zusammenfassung und Ausblick	72
5.1	Zusammenfassung	72
5.2	Diskussion	74
5.3	Ausblick	75
A	BPMN zur Schaden-/Leistungsabwicklungsprozess	77
B	Schaden-/Leistungsabwicklungsprozess	78
C	Weitere Entwicklungsobjekte	83
	Abbildungsverzeichnis	86
	Literaturverzeichnis	87
	Erklärung	90

Kapitel 1

Einleitung

1.1 Motivation und Hintergrund

In den vergangenen Jahren gewann das Process-Mining-Forschungsfeld immer mehr an Popularität. Im Vorfeld dieser Entwicklung sind Anwendungen, wie *Business Process Reengineering* (BPR), *Business Process Management* (BPM) und insbesondere die Workflow-Technologie entstanden. Vor allem die Workflow-Technologie setzt ein Workflow-Modell für ihre Arbeit voraus. Dabei ist ein Workflow-Modell eine Beschreibung eines Arbeitsablaufes [G.00]. Unterdessen erfordert die Erzeugung eines Modells für die Anwendungen tiefe Kenntnisse über Business-Prozesse, wodurch der Vorgang ein kostspieliges Unterfangen werden kann. An dieser Stelle wird die Process-Mining-Technologie relevant. Die hier entwickelten Methoden zur Gewinnung von Geschäftsprozessmodellen aus den Prozessausführungen können Unternehmen viele Kosten ersparen. Dieses junge Forschungsgebiet lässt sich zwischen einer besonderen Form des *Data Minings* und den Geschäftsprozessmodellen positionieren. Abbildung 1.1 illustriert wie *Process Mining* die Softwaresysteme und die dazugehörigen Daten in Form von Log-Dateien einerseits und Prozessmodelle andererseits miteinander verknüpft, um verschiedene Analysemöglichkeiten zu generieren. Dabei sind die drei wichtigsten Typen von *Process Mining*, *Discovery*, *Conformance* und *Enhancement* in Abbildung ebenfalls dargestellt. Auf der einen Seite werden die Probleme eines Unternehmens, in Abbildung *World* bezeichnet, von den Softwaresystemen kontrolliert und unterstützt und auf der anderen Seite hinterlassen diese Systeme Spuren in Form von Log-Dateien, wobei die Log-Dateien mit *Process Mining* analysiert werden können, um Geschäftsprozessmodelle eines Unternehmens zu erfassen. Die Modelle werden anschließend von Business-Experten analysiert und das entstandene Geschäftsverständnis des Unternehmens kann dazu eingesetzt werden, um bessere Softwaresysteme zu entwickeln oder die bestehenden Systeme zu verbessern.

In dem Process-Mining-Forschungsgebiet wurden zahlreiche Techniken entwickelt, um Wissen über Prozesse zu erhalten. Dieses Wissen ist von besonderem Interesse für die Unternehmen. Die Unternehmen wollen die tatsächlichen Vorgänge, die durch die Ausführung eines Prozesses entstehen, verstehen und analysieren, um dementsprechend die Prozesse und letztendlich sich selbst zu verbessern. Das Vorhaben

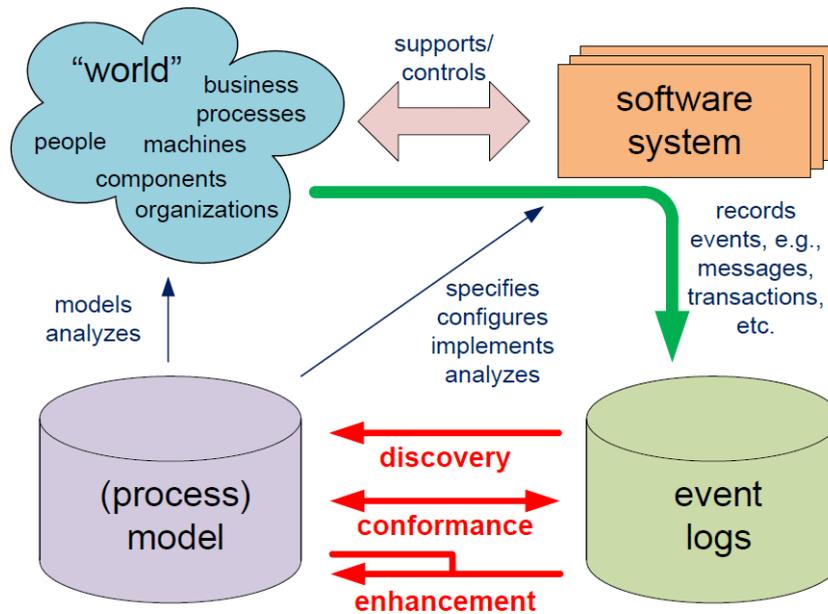


Abbildung 1.1: Positionierung von Process Mining [AWM11]

wird dadurch erschwert, dass Geschäftsprozesse über mehrere Funktionseinheiten oder gar über mehrere Unternehmen verteilt ablaufen. Zusätzlich wird die Komplexität durch die Tatsache erhöht, dass das Wissen über die einzelnen Prozessvorgänge nur bei den jeweiligen Mitarbeitern vorhanden ist, weil nur diese für die konkreten Vorgänge verantwortlich sind. Dies führt dazu, dass niemand einen Gesamtüberblick über die Abläufe eines Unternehmens hat. Folglich können nur mit großem Aufwand Fehler in der Prozessausführung aufgedeckt und Optimierungsarbeiten durchgeführt werden.

Gerade weil Prozesse immer komplexer werden, ist es die Aufgabe eines Unternehmens diese zu überwachen und zu analysieren. Dadurch gelingt es dem Unternehmen, den Überblick über die ablaufenden Prozesse zu behalten sowie die Entstehungsstellen von Fehler aufzudecken und zu beseitigen. Ein ebenfalls wichtiger Aspekt ist, nicht nur die „gelebten“ Prozesse zu analysieren, sondern auch die Kluft zu den „gedachten“ Prozessen zu schließen, wobei die „gelebten“ Prozesse von den Anwendern des Systems erzeugt werden und die „gedachten“ Prozesse von den Architekten des Systems zuerst modelliert und anschließend in dem System umgesetzt werden. Letztendlich gelingt es, die Prozesse zu optimieren und infolgedessen dynamisch auf den Markt zu reagieren.

Die Process-Mining-Technologie generiert Modelle sowie analysiert und überwacht Prozesse anhand der geloggtten Informationen aus verschiedenen Transaktionssystemen. Einige Vertreter [Aal05] der Transaktionssysteme sind *Enterprise Resource Planning* (ERP), *Supply Chain Management* (SCM) und *Customer Relationship Management* (CRM)- Systeme. Alle diese Systeme protokollieren in ihrer eigenen Manier die benötigten Informationen. Dabei müssen für die Anwendung von Process-Mining-Methoden die protokollierten Informationen in Form von ereignisbasierten Daten vorliegen [AWM11].

Der Fokus dieser Arbeit liegt auf der Anwendung von Process-Mining-Ansätzen in einem der größten Vertreter der ERP-Systeme, dem SAP-System. Dabei ist das SAP-ERP-System eine der meist verbreitetsten, betriebswirtschaftlichen Anwendung. Die Software wurde von dem Unternehmen SAP - Systeme, Anwendungen, Produkte in der Datenverarbeitung - 1972 entwickelt. Die Softwarepalette des Unternehmens reicht von der ersten Entwicklung von SAP RF, welches auf Mainframe Systemen funktionsfähig gewesen ist und für die Finanzbuchhaltung entwickelt wurde, über das System SAP R/3, mit dessen Einführung die Migration auf die Client-Server-Architektur vollzogen wurde, wobei das *R* für die Echtzeitverarbeitung und *3* für die Generation der Software steht, bis hin zu SAP ERP. Mit der Einführung von SAP ERP ist ein Paradigmenwechsel vollführt worden, der die hier stattgefundene Migration zur serviceorientierten Architektur (SOA) ermöglichte [M.09]. Während SAP R/3 monolithisch aufgebaut ist, das bedeutet, dass die Applikationen permanent mit der Business-Logik verbunden sind, beruht SAP ERP auf der Service ausgerichteten Systemarchitektur von SAP NetWeaver. Hierbei interagiert NetWeaver als ein Applikationsserver in der SAP-Systemlandschaft und realisiert zusätzlich die SOA. Ein entscheidender Erfolgsfaktor eines Unternehmens ist die Flexibilität, mit welcher das Unternehmen in der Lage ist, seine Geschäftsprozesse schnell anzupassen. Unter Anwendung von SOA lassen sich die vorhandenen Services kurzfristig und flexibel zu neuen Geschäftsprozessen verknüpfen. Zudem unterstützt SAP ERP die *Shared Services* und das *Business Process Outsourcing* (BPO). Auf diese Weise können Prozesse ausgelagert und Kosten gespart werden.

Das Einsatzgebiet von SAP ERP liegt in der Datenverarbeitung aller betrieblichen Funktionsbereiche. Bei dem System handelt es sich um ein integriertes Dialogsystem, wobei auf der Basis einer einheitlichen Datenbank gearbeitet wird. Zudem werden alle Daten einer Eingabemaske direkt nach der Eingabe in der Datenbank gespeichert und für alle weiteren Verarbeitungen verfügbar gehalten. Das SAP-System speichert bei einer Prozessausführung alle für Process-Mining-Methoden benötigten Informationen mit [GM04, J.07, AWM11]. Die zentrale Frage ist, welche Informationen konkret vorhanden sind und auf welche Art und Weise diese aus einem SAP-ERP-System in eine Logdatei extrahiert werden können, so dass gleichzeitig das ereignisbasierte Format entsteht. Das Erzeugen dieser Dateien wird insbesondere dadurch erschwert, dass die geschäftsrelevanten Vorgänge, wie z.B. die Beschaffung eines Materials oder die Materialplanung, nicht in einer Datei bei den SAP-Systemen geloggt werden [GM04, J.07, J.10]. Die Informationen zu diesen Vorgängen werden in einer Datenbank, über mehrere Tabellen verteilt, gespeichert. Dabei werden Datenbanken, wie SAP DB, MS SQL oder IBM DB2, unterstützt. Zudem enthält eine vollständige SAP-Implementierung mehr als 90.000 Tabellen [SB11b]. Demzufolge ist eine Vorverarbeitung für eine Logdatei in einem ereignisbasierten Format als Eingabe für *Process Mining* unumgänglich.

Diese Diplomarbeit findet im Kontext des SAP-ERP-Systems statt, welches zu Lehr- und Forschungszwecken von dem SAP-Hochschulkompetenzzentrum der Universität Magdeburg bereitgestellt wird. Dabei ist das SAP-System durch die Verwendung lokaler Clients via Remote-Zugriff verfügbar gemacht. Hier steht ein In-

ternational Demonstration and Education System-Mandant (IDES-Mandant) zur Verfügung. IDES ist ein integriertes, vollständig entwickeltes Modellunternehmen, das für die Forschung und Lehre benutzt wird. Abbildung 1.2 zeigt den Einstiegsbildschirm des SAP-ERP-Systems in Magdeburg. Die Benutzeroberfläche hat auf der

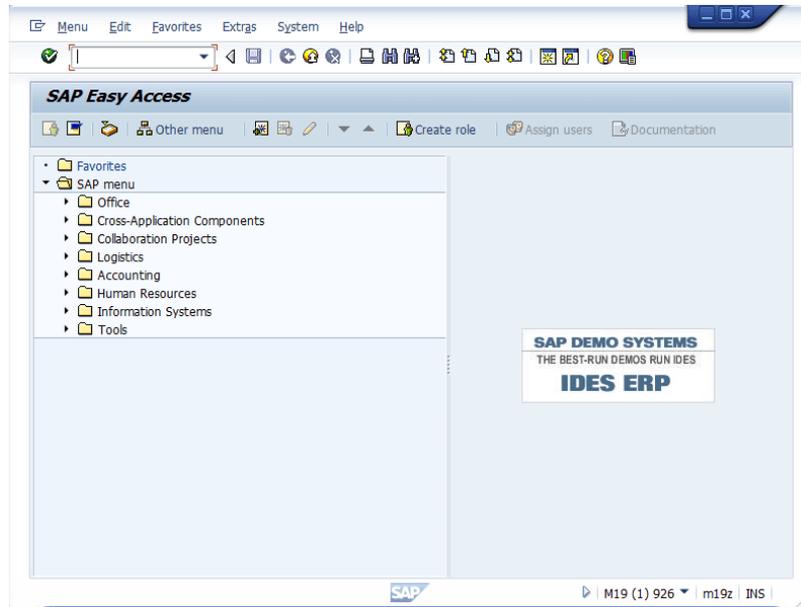


Abbildung 1.2: Screenshot: SAP-System-Benutzeroberfläche

linken Seite das *Easy Access Menu*. Diese Baumstruktur ermöglicht das Navigieren im System und auswählen von Funktionen, die die Aufgaben der Anwender lösen. Auf der rechten Seite befindet sich das Dialogfenster, in welchem die Bildschirmmaske der Funktionen dargestellt wird. Weiterhin erscheinen, im unteren linken Bereich, Meldungen des SAP-Systems.

Das Ziel eines Process-Mining-Projektes besteht in der Extraktion von Informationen aus einem Ereignis-Log, für die Rekonstruktion eines Modells einer logisch zusammenhängenden Kette der Geschäftsprozesse [Aal03]. Das Projekt ist in einem ERP-System in der Dimension von einem SAP-System ein kostspieliges Unterfangen, weil SAP kein Ereignis-Log zur Verfügung stellt und dieses erst erzeugt werden muss. Dazu ist eine Vorverarbeitung, die mit Expertenwissen verbunden ist, notwendig, um die Daten, die für ein Ereignis-Log in SAP relevant sind, zu identifizieren und anschließend zu extrahieren. Das primäre Ziel dieser Diplomarbeit besteht deshalb in der Ausarbeitung und prototypischen Umsetzung eines Lösungsweges, bei dem die in der Datenbank eines SAP-Systems gespeicherten Daten, die als Eingabe für *Process Mining* infrage kommen, zuerst identifiziert werden und anschließend mit den zu SAP konformen Methoden extrahiert werden. Nachdem Rausfiltern der Daten muss in einem weiteren Schritt des Verfahrens eine Logdatei in einem *Process Mining* konformen Format erzeugt werden. Auf der Basis dieser Logdatei soll dann ein Modell mit dem *ProM* erzeugt werden, welches Process-Mining-Methoden bereitstellt. Dabei gewährt *ProM* unter der Benutzung der Vielfältigkeit der Plugins

mehrere Perspektiven einer Prozessausführung. Perspektiven, wie die Leistung einer Ausführung, der Workflow und die Organisationsperspektive, werden betrachtet, wobei in dieser Arbeit insbesondere der Workflow relevant ist [WJ09].

Des Weiteren gilt es zusätzlich zu der grundlegenden Implementierung der Lösung, den gesamten Lösungsweg zu evaluieren. Dazu soll zunächst eine große Menge an Daten, in Form von Prozessausführungen eines Geschäftsprozesses, in dem IDES-Modellunternehmen erzeugt werden. Ferner werden die SAP-Referenzmodelle, die als Vorgabe für die Prozessmodellierung von SAP mitgeliefert werden, in der Evaluierung berücksichtigt. Validiert werden soll dabei, inwieweit das mit dem Werkzeug *ProM* erzeugte Modell dem SAP-Referenzmodell entspricht und dabei konsensuell die Prozessausführung wiedergibt.

1.2 Verwandte Arbeiten

Ferner gibt es weitere Arbeiten, deren Zielsetzung die Anwendung von Process-Mining-Ansätzen in SAP-ERP-Systemen ist. Diese Arbeiten unterteilen sich in wissenschaftliche- und kommerzielle Arbeiten. Die Anwendung von Process-Mining auf die Workflow-Management-Systeme ist ein weit erforschtes Themengebiet, dagegen ist die Anwendung von Process-Mining-Ansätzen in SAP-ERP-Systemen eine in der Wissenschaft junge Fragestellung, deshalb ergab die Recherche wenige Arbeiten.

Die **erste Masterarbeit** von Martijn van Giessel findet sich im Jahr 2004 [GM04]. In dieser Diplomarbeit wurde eine Zweischnitttechnik entwickelt, um eine Datei, die sich als Eingabe für Process-Mining-Methoden eignet, zu erzeugen. Im ersten Schritt ist die Applikation *TableFinder* implementiert, welche die benötigten Tabellen aus einem betriebswirtschaftlichen Sachverhalt im SAP-ERP-System ermittelt, der durch die sogenannten Business-Objekttypen (BO) repräsentiert ist. Nachdem die relevanten Tabellen herausgefiltert werden, wird aus ihnen im zweiten Schritt der Belegfluss einer Prozessausführung extrahiert. Anschließend werden die erforderlichen Daten aus den Tabellen in ein für *Process Mining* geeignetes Extensible Markup Language-Format (XML-Format) eingefügt.

Neben dem Expertenwissen der SAP-ERP-Systeme sind in dieser Arbeit viele Schritte manuell ausgeführt. Insbesondere die Extraktion der Daten aus der Datenbank eines SAP-Systems ist ein manueller Schritt. Dazu kann der *TableFinder* nicht alle relevanten Tabellen identifizieren. Diverse Tabellen sind nicht zu bestimmten BOs und Prozessen zugeordnet, weil sie z.B. die Änderungen von allen in dem System vorhandenen Objekttypen abbilden und deshalb nicht direkt dem Belegfluss eines bestimmten Prozesses zugeordnet sind, jedoch sind sie für *Process Mining* relevant. Diese Tabellen müssen in dieser Arbeit nachträglich manuell herausgefiltert werden. Daraufauf folgt im zweiten Schritt die logische Verbindung zwischen den Tabellen manuell eingefügt. Hierzu müssen die Primär- und die Fremdschlüssel zuerst in den Tabellen gefunden und dann miteinander verbunden werden. Trotz der Schwächen ebnete diese Arbeit den Weg zu weiteren Untersuchungen und zeigte, dass so ein hoch kompliziertes Projekt wie *Process Mining* in SAP ERP erfolgreich ausgeführt werden kann.

Die **zweite Masterarbeit**, die sich mit dem Thema befasst, ist die Diplomarbeit von Segers, aus dem Jahr 2007 [I.07]. Hier sind zwei Methoden zur Analyse eines Geschäftsprozessmodells des Beschaffungsprozesses in SAP ERP vorgestellt. Die Erste verwendet die Informationen, die in der Tabelle zu den Transaktionscodes einer Prozessausführung abgelegt werden. Das heißt, es wird auf die Tabelle des sogenannten Transaktionsmonitors zugegriffen, die CDHDR-Tabelle in SAP. Ein Transaktionsmonitor protokolliert Informationen zu einem bestimmten Prozessschritt. Dies ist eine erfolglose Vorgehensweise, die schon in der ersten Arbeit beschrieben wurde, weil ein Prozessschritt nicht nur durch denselben Transaktionscode ausgeführt werden kann. Es existieren Transaktionscodes, die mehrere z.B. Bestellungsprozesse in einem Schritt ausführen können, sodass keine eindeutige Zuordnung eines Prozessschritts zu einem Transaktionscode stattfinden kann. Deshalb ist ein sinnvolles Modell mit den Process-Mining-Methoden auf der Basis von Transaktionscodes nicht erstellbar.

Die zweite Vorgehensweise verwendet die aus dem Beschaffungsprozess in SAP bekannte Tabellen und basiert ausschließlich auf der Beschreibung dieser Tabellen und der Zuordnung von diesen zueinander per Fremdschlüssel. Dabei ist der Weg zur Ermittlung der Tabellen sowie die Zuordnung der Inhalte zueinander nicht beleuchtet. Deshalb stellt sich auch bei dieser Arbeit die Frage, wie die relevanten Daten in einem SAP-System identifiziert, daraus extrahiert und anschließend in Verbindung gebracht werden.

Die **Masterarbeit von Buijs** aus dem Jahr 2010 verfolgt das Ziel der Implementierung eines generischen Werkzeuges, welches einen Businessanalysten dabei unterstützen soll, aus den Tabellen einer vorliegenden Datenbank eine Logdatei im Ereignis-Format zu erzeugen [J.10]. Diese Logdatei soll anschließend mit dem Werkzeug *ProM* analysiert werden können. Dazu entwickelte Buijs eine Anwendung, wobei diese die drei in Abbildung 1.3 dargestellte Schritte ausführt, um eine Log-Datei zu erzeugen. Im ersten Schritt muss ein Businessanalyst eine Datei erzeugen, wobei

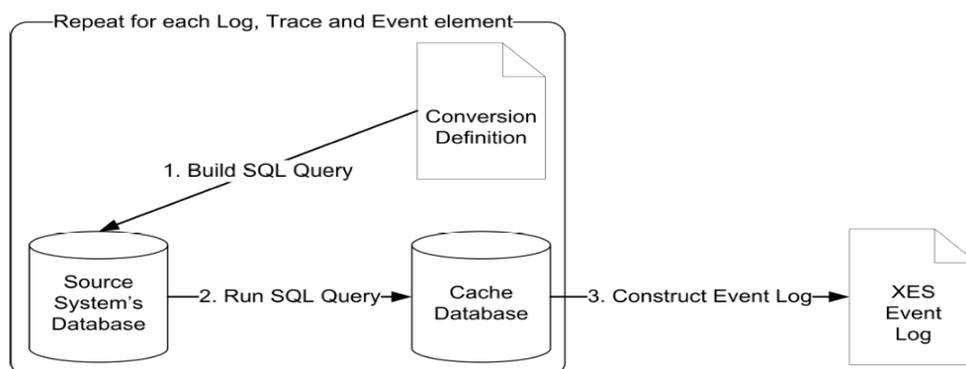


Abbildung 1.3: Drei-Schritte-Methode von Buijs [J.10]

in dieser Datei spezifiziert wird, welche Daten eines Geschäftsprozesses aus der Datenbank extrahiert werden sollen. Nachdem die Daten im zweiten Schritt extrahiert wurden, werden diese in einer lokalen Datenbank gespeichert. Im letzten Schritt wird

eine Log-Datei aus den Daten in der lokalen Datenbank konstruiert, wobei diese für die nachfolgende Analyse mit dem *ProM* zur Verfügung steht. Dabei liegt dieser Arbeit ein Ausschnitt einer Datenbank vor, die von Laboratory for Quality Software (LaQuSo)¹ zur Verfügung gestellt wurde, so dass die Daten nicht mit der in dieser Arbeit entwickelten Anwendung aus einem SAP-System extrahiert wurden. Des Weiteren sind die benötigten Tabellen als bekannt angenommen. Dabei wurden die Schritte zur Identifizierung der Tabellen in SAP ERP nicht erläutert.

Auch in dieser Arbeit ist die Frage nach dem Identifizieren von relevanten Tabellen offen geblieben. Weiterhin sind hier nicht SAP konforme Methoden benutzt worden, um die Daten aus den Tabellen zu extrahieren. Structured Query Language-Anweisungen (SQL-Anweisungen), welche in dieser Arbeit für die Extraktion der Daten vorgeschlagen wurden, die direkt auf die Datenbank zugreifen, werden von SAP nicht unterstützt [I.07].

Einige kommerzielle Arbeiten sind bereits entwickelt. Besonders zu erwähnen ist die **Enterprise Validation Suite** (EVS) ModelBuilder. Dies ist ein *Process- und Data Mining Framework*, welches von Businesscape vertrieben wird. Ingvaldsen et al. verwenden den *EVS ModelBuilder* in ihrer Arbeit, um *Process Mining* auf Ereignis-Ketten aus SAP-Systemen anzuwenden [J.07]. Ereignis-Ketten sind generische Gebilde im Gegensatz zu einer Spur, weil die Ereignisse in den Ereignis-Ketten nicht zu einer bestimmten Prozessinstanz korrespondieren. Dagegen ist eine Spur eine Aufzeichnung von Ereignissen, die während einer Prozessinstanzausführung aufgetreten sind. In dieser Arbeit ist gezeigt, wie der *EVS ModelBuilder* im Zusammenhang mit SAP-Systemen zu benutzen ist. Das System erlaubt dem Benutzer eine Abbildung zu beschreiben, um die Ereignis-Ketten zu extrahieren. Eine Beispielabbildung, die ein Ausschnitt eines Beschaffungsprozesses aus einem SAP-ERP-System extrahiert, ist in Abbildung 1.4 dargestellt. Dabei repräsentieren

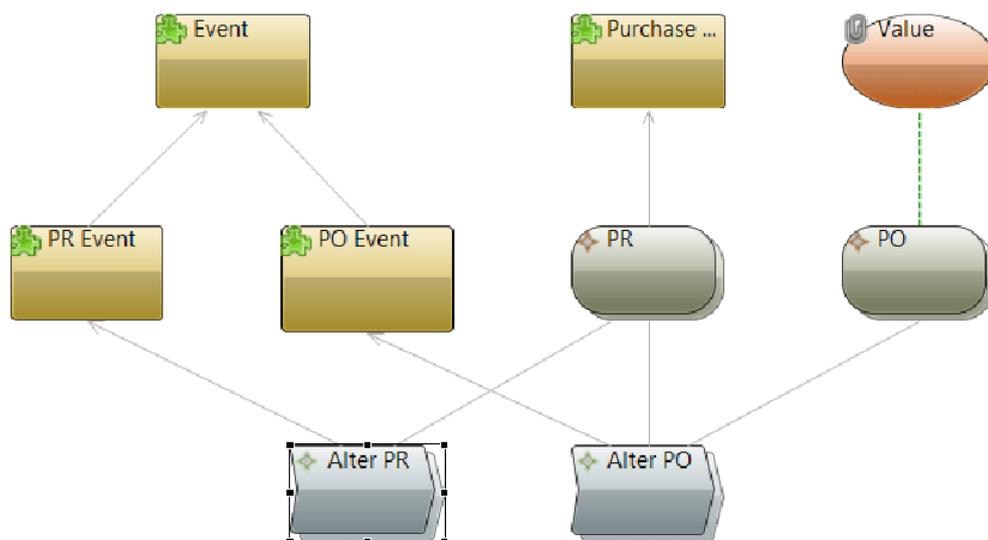


Abbildung 1.4: Eine Abbildung in: EVS ModelBuilder [J.10]

¹Siehe: <http://www.laquso.com>

die rechteckigen Flächen mit den abgerundeten Ecken die BOs. In Abbildung sind zwei dargestellt, die Bestellung (PO) und die Bestellanforderung (PR). Diese können weitere Eigenschaften, wie z.B. der Wert in dem PO, enthalten. Weiterhin können BOs und die Ereignisse einer Klasse angehören. In dem Beispiel gehören die Ereignisse „Alter PR“, „Alter PO“ und der Objekttyp PR zu den dazugehörigen Klassen, die durch die rechteckige Flächen dargestellt sind. Dadurch können die Elemente gruppiert werden, wobei die Klassen selbst weiteren Klassen angehören können, sodass eine Hierarchie entstehen kann. In diesem Beispiel wurden die beiden BOs geändert. Deshalb sind sie mit den beiden Änderungs-Ereignissen verbunden. Das Resultat der Abbildungsdefinition ist eine Ereignis-Kette.

Aus der Arbeit von Ingvaldsen et al. geht hervor, dass die Relation zwischen den Ereignissen und den einzelnen Prozessinstanzen nicht vorhanden ist, weshalb die Erzeugung einer Logdatei in XML-Format nicht möglich ist. Zusätzlich muss jedes Ereignis explizit spezifiziert werden. Diese Gründe führten zu der Entscheidung, dass der *EVS ModelBuilder* nicht für die Erreichung der in dieser Arbeit gestellten Ziele geeignet ist.

Eine weitere kommerzielle Lösung findet sich im **ARIS Process Performance Manager** (Arise PPM) wieder, einem System von Integrierte Datenverarbeitung Systeme Scheer (IDS Sheer)². Ab der Version 3.2 ist laut SAP.info unter anderem ein Process-Mining-Modul in der Software integriert [Sch04]. Das Kernkonzept des Systems ist jedoch, die Abläufe eines Unternehmens mithilfe der dazugehörigen *Key Performance Indicators* (KPI's) darzustellen [She05]. Dazu wird das System bei der Installation per Konfigurationsdatei konfiguriert. Hier werden unter anderem die relevanten Tabellen, aus denen die Informationen extrahiert werden müssen, sowie die Fremdschlüssel eingegeben. Des Weiteren werden in der Datei weitere Filter und die KPIs eingestellt. Eine Beispielarchitektur des Extraktionsvorgangs ist in Abbildung 1.5 dargestellt. Hier ist zu sehen, dass verschiedene Komponenten an der Ausführung eines Prozesses beteiligt sein können. Die relevanten Komponenten werden per Ad-

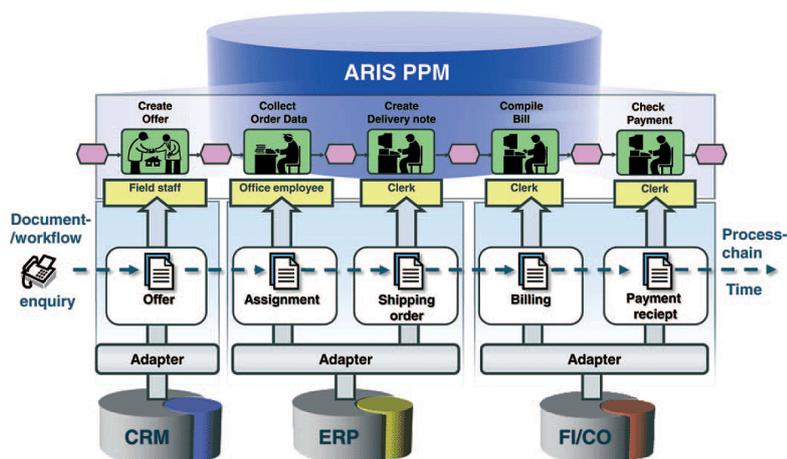


Abbildung 1.5: Extraktion von Prozessdaten [She05]

²Siehe: <http://www.softwareag.com/DE/>

apter, der durch die Konfigurationsdatei eingestellt ist, an das System angeschlossen. Aus der Komponente z.B. Finanzwesen werden mit einem Adapter, in Abbildung 1.5, die Information über eine Rechnungserstellung und über eine Bezahlung einer Rechnung extrahiert. Der Adapter beinhaltet den SAP Java Connector (SAP JCo), welcher per *Remote Function Call* (RFC) mit dem SAP-System kommuniziert. Dadurch können die in den bestimmten Komponenten aufgetretenen Ereignisse aus der Datenbank extrahiert werden und im ARIS PPM Process Warehouse abgelegt werden. Hier sind sie durch die in der Konfigurationsdatei angegebenen Fremdschlüssel miteinander verbunden, so dass ein Geschäftsprozessmodell rekonstruiert werden kann.

Bei dem Test der Software konnte eine Prozessausführung in ereignisgesteuerten Prozessketten-Notation (EPK-Notation) dargestellt werden. In Abbildung 1.6 ist ein Ausschnitt illustriert, welcher eine EPK zur Auftragsabwicklung darstellt. Eine Flä-

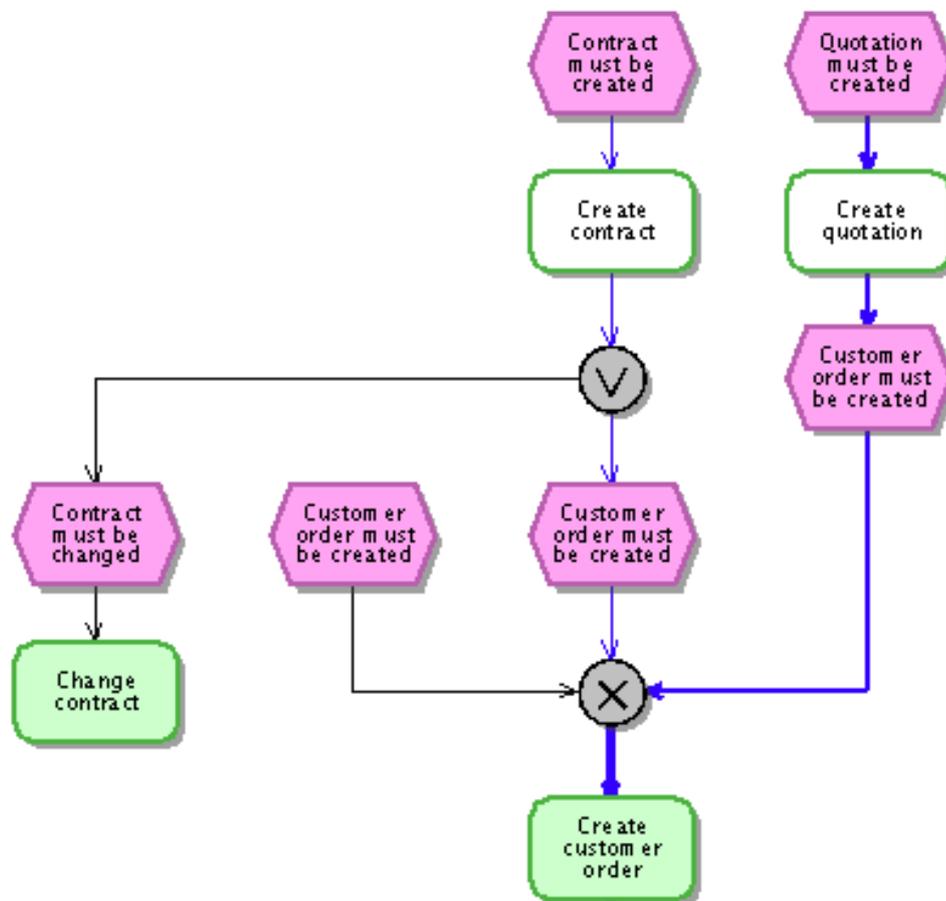


Abbildung 1.6: EPK-Ausschnitt zu Auftragsabwicklung aus ARIS PPM

che mit spitzen Ecken repräsentiert ein Ereignis. Dabei ist eine Funktion durch eine abgerundete Fläche dargestellt. In dem Beispiel ist erkennbar, dass eine Auftragsabwicklung entweder durch das Ereignis „Contract must be created“ oder „Quotation must be created“ beginnt. Anschließend folgen die Aktivität „Create contract“ oder „Create quotation“.

Weitere Process-Mining-Methoden sind in dem Testpaket nicht vorhanden und kann-

ten deshalb nicht getestet werden. Außerdem ist der Input des PPMs nicht eine Logdatei, so wie sie für *Process Mining* erforderlich ist, sondern bereits vorhandene Referenzmodelle in Form von EPKs. Die EPKs werden um weitere Ereignisse von der Anwendung erweitert, falls diese in dem SAP-System verändert wurden oder neu dazugekommen sind. Diese Gründe führen dazu, dass PPM ebenfalls für diese Arbeit ungeeignet ist.

1.3 Aufbau der Arbeit

Zunächst werden in Kapitel 2 die Grundlagen erläutert, die zum Verstehen der Funktionsweise eines SAP-Systems und der Process-Mining-Technologie erforderlich sind. Ein Überblick über die Technologien sowie die Aufbauweise eines SAP-ERP-Systems bis hin zu dem IDES-Modellunternehmen gehören dazu. Des Weiteren wird ein Überblick mit den Grundbegriffen sowie die Vorstellung von dem in dieser Arbeit benutzten Werkzeug *ProM* bis hin zu dem Eingabeformat für das Werkzeug ebenso gegeben. Letztlich wird die Verbindung zwischen den zuvor vorgestellten Themengebieten hergestellt. Kapitel 3 erläutert die Realisierung dieser Verbindung. Dazu wird zuerst betrachtet, wie der Vorbereitungsschritt auf der Seite von einem SAP-System erfolgt. Die Methoden zur Datenextraktion und die Zugriffsmöglichkeiten werden in Kapitel 3.3 dargestellt. Anschließend werden die aus einem SAP-System gewonnenen Daten mit dem *ProM* analysiert. Dazu wird in Kapitel 3.4 die in dieser Arbeit benutzte Möglichkeit zur Anbindung an das Werkzeug beschrieben. In Kapitel 3.5 wird die Bedienung des Systems sowie die Benutzeroberfläche vorgestellt. Kapitel 4 befasst sich mit der Fallstudie Beschaffung. Dazu wird ein Tutorial zur Durchführung der Fallstudie erläutert und im Anschluss daran wird die Validierung des gesamten Lösungsweges mit den Ergebnissen und gewonnenen Erkenntnissen beschrieben. Nach der Deutung der Ergebnisse schließt die Arbeit mit einem Ausblick in Kapitel 5.3 ab.

Kapitel 2

Grundlagen

Dieses Kapitel erläutert die beiden für diese Arbeit relevanten Themen. Zum einen wird das SAP-System und zum anderen die Process-Mining-Technologie dargestellt. Zunächst wird ein Überblick über das SAP und insbesondere das ERP-System gegeben. Dabei werden insbesondere die allgemeine Struktur und die Funktionsweise erörtert. Dazu wird das IDES-Modellunternehmen als Beispiel eingesetzt. Die Grundlagen von *Process Mining* werden im Abschnitt 2.2 beschrieben. In Bezug darauf werden vor allem das in dieser Arbeit verwendete *ProM* sowie die Ereignisprotokolldatei und das Format, welches als Eingabe für *ProM* geeignet ist, erläutert. Der letzte Teil dieses Kapitels befasst sich mit der Frage, wie ein SAP-System mit den Process-Mining-Methoden analysiert werden kann. Diesbezüglich wird besprochen, welche Daten für *Process Mining* benötigt werden und wie diese aus SAP-ERP-System gewonnen werden können.

2.1 Enterprise Resource Planning mit SAP ERP

Das Unternehmen SAP wurde von fünf ehemaligen IBM-Mitarbeitern gegründet, deren Vision die Entwicklung von Anwendungssoftware für die Echtzeitverarbeitung gewesen ist. Heute ist die SAP AG mit über 50.000 Mitarbeitern der drittgrößte Softwareanbieter von Geschäftslösungen [SAPa]. SAP war eines der ersten Unternehmen, welches Standard-Software für ERP entwickelte. Dabei ist der Begriff *Standard* in diesem Zusammenhang als vorkonfigurierte Software zu verstehen, die an die Anforderungen eines Unternehmens anpassbar ist. Der Prozess der Anpassung wird bei SAP als Customizing bezeichnet.

Seit der ersten Entwicklung 1973 der „rechnergestützten Abwicklung der Finanzbuchhaltung“, auch als RF-System bezeichnet, welches sich ausschließlich auf die Finanzbuchhaltung konzentrierte, vollzog sich ein Wandel in allen Bereichen der Software bis zu dem heute aktuellen SAP ERP 6.0-System [H.06]. Diese Arbeit findet im Rahmen der neuesten Generation der SAP-ERP-Software statt, der SAP-ERP-Central Component (ECC), wobei unter ECC die Vereinigung aller vorhandenen und der neu entwickelten Komponenten zu verstehen ist.

Nach dem SAP-RF-System mit der viel eingesetzten Komponente „Finanzbuchhaltung“ sind viele weitere Komponenten entwickelt worden. Das SAP ERP ist in drei

Hauptmodule strukturiert, wobei jedes dieser Module weitere Sub-Module beinhaltet. Die folgende Auflistung stellt die bedeutsamen Module in jedem Hauptmodul dar, wobei in dieser Arbeit insbesondere das SAP Materialwirtschaft (MM) relevant ist:

- Rechnungswesen
 - SAP FI: Finanzwesen
 - SAP CO: Controlling
- Logistik
 - SAP LO: Logistik Allgemein
 - SAP SD: Vertrieb
 - SAP MM: Materialwirtschaft
 - SAP PP: Produktionsplanung und -steuerung
- Personal Management
 - SAP HCM: Personalwirtschaft

Das MM-Modul ist der Logistikkomponente des SAP-ERP-Systems zugeordnet, wobei dieser Modul für die Verwaltung der Materialbewegungen innerhalb eines Unternehmens verantwortlich ist.

2.1.1 Konzept

Dieser Abschnitt erläutert einige grundlegende Konzepte des SAP-Systems. Eines der Konzepte ist das Organisationsmodell, welches alle Unternehmen ab einer bestimmten Größe haben. Das Modell bildet im Zusammenhang mit SAP ERP die logische Struktur eines Unternehmens auf die organisatorischen Einheiten des SAP-Systems ab. Die Abbildung geschieht für jeden Betrieb bei der Durchführung von Customizing einzigartig. Das Customizing beginnt mit der Beschreibung der höchsten Organisationseinheit des Unternehmens, weil SAP-ERP-Systeme als Mandantensysteme interagieren und ein Mandant die höchste Organisationseinheit in SAP darstellt [W.98, M.08]. Ein Mandant kann als Konzern, Unternehmen oder als Betrieb interpretiert werden. Dabei ist dieser eine abgeschlossene Einheit. Demzufolge hat jeder Mandant eigene Tabellen und Daten. Durch das Konzept ist die Möglichkeit, mehrere getrennte Unternehmen gemeinsam zu leiten, gegeben.

Ein Buchungskreis ist nach dem Mandanten die nächsthöhere Organisationseinheit [M.08]. Im Sinne der Betriebswirtschaft ist der Buchungskreis eine Gesellschaft, wie z.B. eine Firma oder eine Tochtergesellschaft. Eine beispielhafte, hierarchische Strukturierung der Organisationseinheiten Mandant, Buchungskreis, Werk und der Lagerhalle ist in Abbildung 2.1 dargestellt. Dabei verwaltet ein Mandant mehrere Buchungskreise. Im Beispiel verwaltet der Mandant mit der Nummer 001 die Buchungskreise 0001 und 0002. Des Weiteren kann ein Buchungskreis mehrere Werke

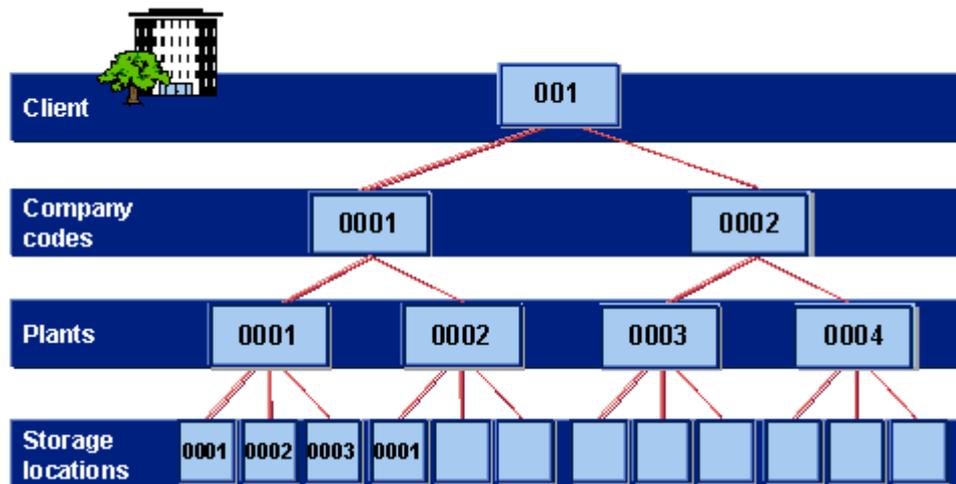


Abbildung 2.1: Ein Beispiel einer Unternehmensstruktur in SAP [SBc]

unterhalten. Der Buchungskreis 0001 in Abbildung 2.1 unterhält das Werk 0001 und 0002. Weiterhin sind jedem Werk eine oder mehrere Lagerhallen zugeordnet. Dabei sind dem Werk mit der Nummer 0001 auf der untersten Ebene drei Lagerhallen zugeordnet.

Ein weiteres Konzept findet sich bei der Ausführung von Transaktionen wieder, bei denen die Stamm- und Bewegungsdaten verwendet werden. Stammdaten sind dabei Daten, die dauerhaft im System für Geschäftsprozesse gesichert sind. Diese Informationen werden zentral abgelegt und sind für alle zuständigen Benutzer und Anwendungen zugänglich [M.08]. Beispielhafte Stammdaten sind Daten über Kunden, Lieferanten oder Materialien. Dagegen sind Bewegungsdaten prozessbezogene Daten, die im Vergleich zu Stammdaten eine kurze Lebensdauer aufweisen. Eine Bestellung entspricht den Bewegungsdaten, die Kunden oder Lieferanten zugeordnet sein kann. Bei der Ausführung von Anwendungsprogrammen, die in SAP als Transaktionen bezeichnet werden, werden erzeugte Bewegungsdaten in Form von Belegen im System gespeichert. Die Transaktionen führen dabei Geschäftsprozesse, wie z.B. „Kundenauftrag anlegen“ oder „Wareneingang buchen“, aus. Diese Prozesse sind Advanced Business Application Programming (ABAP)-Programme, wobei ABAP eine Programmiersprache ist, die von SAP entwickelt wurde [P.01]. Z.B. führt eine Transaktion mit dem Code „ME51N“ den Prozess „Bestellanforderung erzeugen“ aus. Dabei existieren über 100.000 verschiedene Transaktionen [AWM11]. Die nach der Ausführung einer Transaktion erzeugten Belege fassen alle relevanten Informationen aus den Stammdaten und Organisationselementen zusammen. Dazu wird für jeden Beleg eine Identifikationsnummer erzeugt. Abbildung 2.2 veranschaulicht die Struktur eines Beleges bei der Ausführung einer Transaktion. Dabei ist eine Transaktion dargestellt, bei der Stamm- und Bewegungsdaten involviert sind und nach der Ausführung die Daten in einem Beleg gespeichert werden. Des Weiteren bilden Belege, die nach der Ausführung von Transaktionen erzeugt werden und dabei miteinander verknüpft sind, einen Belegfluss. Dabei stellt ein Belegfluss

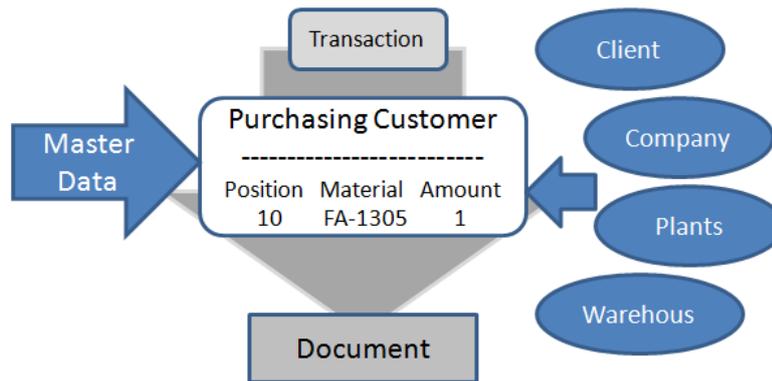


Abbildung 2.2: Transaktion in SAP ERP

ein Geschäftsprozess in SAP ERP dar. Ferner werden alle zu einem Beleg gehörenden Informationen in Tabellen abgelegt. Falls eine Bestellanforderung durch die ME51N-Transaktion angelegt wird, resultiert diese in einem Beleg mit einer eindeutigen Nummer in der Bestellanforderungs-Tabelle (EBAN). Falls im weiteren Verlauf zu dieser Bestellanforderung tatsächlich eine Bestellung durchgeführt wird, kann die Bestellung per Identifikationsnummer in Relation zur Bestellanforderung gebracht werden. Die dazugehörige Bestellung wird in einer anderen Tabelle (EK-PO) gespeichert als die Bestellanforderung. Deshalb spiegelt die Beziehung zwischen den Tabellen die Beziehung zwischen den Belegen ab.

Die Verknüpfung der Tabellen findet per Fremdschlüssel statt. Dabei definiert ein Fremdschlüssel die Abhängigkeit zu einem Primärschlüssel, wobei der Primärschlüssel einen Datensatz eindeutig identifiziert. Das bedeutet, dass die Verbindung zwischen einer Tabelle mit Bestellanforderungen und einer Tabelle mit den tatsächlichen Bestellungen per Fremdschlüssel stattfindet, wobei der Fremdschlüssel in SAP ERP eine Kombination sein kann, die u.a. eine Identifikationsnummer aus dem Beleg der Bestellanforderung und weitere Felder enthält.

2.1.2 Das SAP-Modell-Unternehmen

IDES ist ein weltweit agierender Konzern, welches eine Musterfirma darstellt. Der Konzern besteht aus vier Zweigen, die ihrerseits aus weiteren Teilbereichen bestehen. Die Zweige Industrie, Einzelhandel, Banken und Services sind vorhanden [SBe].

Der für diese Arbeit relevante Konzernbereich IDES Industrie beherbergt die Teilkonzerne IDES Europa, Amerika, Asien, die ihrerseits aus eigenen Organisationen aufgebaut sind. Außerdem ist jeder Teilkonzern in Buchungskreise untergliedert. Der hier relevante Teilkonzern ist der IDES Europa, wobei der Buchungskreis 1000 (Deutschland) insbesondere relevant ist. Der Buchungskreis 1000 beinhaltet Geschäftsprozesse aus Logistik, Rechnungswesen, Personalwirtschaft und Finanzbuchhaltung. Dabei liegt der Fokus in dieser Arbeit auf den Geschäftsprozessen aus der Logistik. Weiterhin interagiert dieser Buchungskreis als Hersteller von Automobilen im Werk 1000. Ferner interagiert das Werk 1000 seinerseits als Produktionsstätte in dieser Arbeit.

2.2 Process Mining

Die Process-Mining-Technologie ist eine Vereinigung von Algorithmen, wobei diese Algorithmen für die Geschäftsprozessentdeckung, die Analyse und das Monitoring eingesetzt werden. Dazu ist es sinnvoll, die vorhandenen Verfahren entsprechend ihren Aufgaben zu ordnen. Deshalb wird in diesem Abschnitt zuerst eine bereits vorhandene Ordnung der Algorithmen erläutert. Anschließend werden die Ereignis-Log und insbesondere das XES-Format dargestellt, welches in dieser Arbeit relevant ist. Letztendlich wird im letzten Abschnitt das Werkzeug *ProM* erläutert, welches die Algorithmen in Form von Plugins anbietet und als Input ein Ereignis-Log in XES-Format verwendet.

2.2.1 Überblick

Die Prozessmodellierung ist einer der größten Aspekte von *Process Mining*. In diesem Bereich werden Prozessmodelle aus Ereignisdateien ohne bestehendes Vorwissen erzeugt [Aal03, AWM11]. Dabei ist der α -Algorithmus das signifikante Verfahren, um Modelle in der Petri-Netz-Notation zu generieren [AWM11]. Weiterhin können verschiedene Arten von Modellen erzeugt werden. Falls z.B. die Ereignisse das Attribut Ressource haben, ist die Entdeckung von Ressourcenmodellen möglich.

Die Analyse ist der nächste Typ von *Process Mining*. Hier werden vorgegebene Modelle mit den aus der Ereignisprotokolldatei desselben Prozesses erfassten Modellen verglichen, um Unstimmigkeit aufzudecken [Aal03, AWM11]. Falls z.B. eine Bestellung in Auftrag gegeben wird und sie laut dem vorgegebenen Modell zugleich von zwei Mitarbeitern geprüft werden muss, dies jedoch laut dem erzeugten Modell aus dem Ereignislog nicht erfüllt wird, ist dieser Sachverhalt als Fehler identifiziert.

Der dritte Typ ist die Erweiterung. Hier geht es darum, die bestehenden Modelle zu erweitern oder zu verbessern [Aal03, AWM11]. Die Analyse zielt darauf ab, das gewonnene Modell mit dem existierenden Modell zu vergleichen. Dagegen wird in der Erweiterung an dem existierenden Modell gearbeitet. Falls dem Ereignislog entnommen wird, dass zwei Aktivitäten in paralleler Anordnung ausgeführt werden, sie jedoch in dem Modell sequenziell ausgeführt sind, kann das Modell erweitert und der Realität angeglichen werden.

Die Process-Mining-Technologie beschränkt sich nicht nur auf die drei Typen, sondern bietet auch verschiedene Perspektiven einer Prozessausführung an. Van der Aalst unterscheidet in seinem Buch „Process Mining: Discovery, Conformance and Enhancement of Business Processes“ drei Perspektiven [AWM11].

Die Control-Flow-Perspektive bietet unter Anwendung von Notationen, wie Petri-netze oder EPK, eine Beschreibung von allen möglichen Geschäftsprozessabläufen an. Das Ziel der Organisationsperspektive ist das Aufzeigen des sozialen Netzwerkes und die Strukturierung von Ressourcen in einem Unternehmen. Eine Case-Perspektive bietet eine detaillierte Darstellung eines Verlaufs und zeigt dessen Beitrag in den Geschäftsprozessen auf. Diese Perspektive kann durch den Wert ihrer aufgetretenen Ereignisse beschrieben werden. Falls z.B. eine bestimmte Bestellung

in einem Unternehmen stattfindet, dann sind die bestellte Menge sowie der mögliche Zulieferer an dieser Stelle interessant. Die letzte Perspektive der Geschäftsprozesse ist die Zeitperspektive. Falls Ereignisse einen Zeitstempel als Attribut aufweisen, kann z.B. das „Flaschenhalsproblem“ in der Ausführung aufgezeigt werden.

Die Pionierarbeit im Process-Mining-Forschungsobjekt hat Cook in seiner Doktorarbeit im Jahr 1996 geleistet [E.06]. Hier wurden Neuronale Netze und Markovmodelle für die Extraktion von Prozessmodellen verwendet. Schon damals zeigte sich das Potenzial von *Process Mining*, da keine Annahmen hinsichtlich der Eingabe getroffen werden. Lediglich eine Logdatei stellt die Grundlage für *Process Mining* dar. In Abschnitt 2.2.2 werden deshalb die Grundbegriffe einer Ereignisprotokolldatei beschrieben.

2.2.2 Ereignis-Log

In einer Ereignisprotokolldatei werden Ereignisse protokolliert, die von den Anwendern des Systems, welches die Ereignisprotokolldatei erstellt, ausgeführt wurden. Prozessorientierte Systeme, wie das SAP-ERP-System, speichern alle für eine mit Process-Mining-Methoden analysierbare Ereignisprotokolldatei benötigten Ereignisse [AWM11, J.10, GM04, I.07, Sch04]. In Abbildung 2.3 ist ein Auszug aus der Bestellanforderungstabelle des IDES-Modellunternehmens dargestellt. Dabei reprä-

	MANDT	BANFN	BNFPO	ERNAM	ERDAT	MATNR	WERKS	UZEIT
<input type="checkbox"/>	926	0010015263	00010	BLYUFSHTEYN	29.04.2011	SCHALTUNG-1305	1000	00:00:00
<input type="checkbox"/>	926	0010015263	00020	BLYUFSHTEYN	29.04.2011	SCHALTUNG-1305	1000	00:00:00
<input type="checkbox"/>	926	0010015283	00010	DISPONENT 00	16.06.2011	WIPV21-B99	1000	00:00:00
<input type="checkbox"/>	926	0010015284	00010	DISPONENT 00	16.06.2011	WIPV21-W99	1000	00:00:00

Abbildung 2.3: Ereignisdaten in SAP ERP

sentierte eine Zeile ein Ereignis. In Abbildung 2.3 ist ein Ereignis per Kombination aus den Datenfeldern *BANFN* und *BNFPO* eindeutig identifiziert. Diese Kombination entspricht einer Ereignisnummer. Weiterhin stellt das Feld *ERNAM* die Ressource dar, die an der Ausführung beteiligt ist. Diese kann sowohl eine Person als auch ein System sein. Der Zeitstempel ist durch die beiden Felder *ERDAT* und *UZEIT* repräsentiert. Das Feld *MATNR* bildet das Material ab, für welches die Bestellanforderung erzeugt wurde. Schließlich wird in dem Feld *WERKS* das Werk dargestellt, in welchem die Bestellanforderung stattfindet.

Ferner können Ereignisse einen Typ haben. Ein Typ beschreibt, in welcher Phase der Ausführung sich eine Aktivität befindet [Aal05, AWM11]. Die signifikanten Typen sind *Start*- und *Complete*. Falls keine Start- und Endzeit in einem Ereignis vorhanden sind, wird der Typ auf *Complete* gesetzt. Generell sind verschiedene Typen möglich, der Zustandsautomat in Abbildung 2.4 zeigt ein Ereignismodell. Hier sind alle Typen, die dazu möglichen Aktivitäten und die Resultate der Aktivitäten dargestellt. Ein Beispiel ist der Ereignistyp *Start*. Nachdem die Ausführung eines Ereignisses gestartet ist und den Ereignistyp *Start* zugewiesen bekommt, kann sie angehalten und den Ereignistyp *Suspend* erhalten und wieder fortgesetzt werden, wobei sie dann den Ereignistyp *Resume* beinhaltet. Diese Schleife kann unendlich

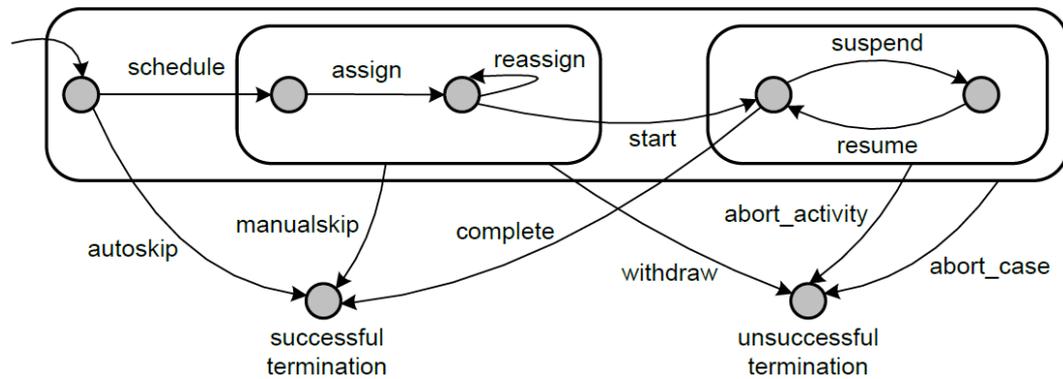


Abbildung 2.4: Ein Ereignismodell nach van der Aalst et al. [AWM11]

oft wiederholt werden, bis der Ereignistyp *Complete* oder *Abort_Activity* eintritt, wobei der *Complete* zu einer erfolgreichen Terminierung und der *Abort_Activity* zum Abbruch der Aktivität führt. Der Ereignistyp *Reassign* in Abbildung 2.4 stellt die zweite Schleife dar. Dabei weist eine Aktivität den Typ *Reassign* solange auf, bis sie gestartet wird und den Typ *Start* zugewiesen bekommt.

Eine Sequenz der Ereignisse stellt eine Spur dar. Dabei werden die Ereignisse per Ereignisnummer zusammengefasst und einer Spur, die per Prozessinstanznummer identifiziert wird, zugeordnet. Aus Abbildung 2.3 lassen sich folgenden drei Spuren bilden, $\langle BLYUFSHTEYN, BLYUFSHTEYN \rangle$, $\langle DISPONENT\ 00 \rangle$, $\langle DISPONENT\ 00 \rangle$. Hierbei entspricht der Prozessinstanznummer in diesem einfachen Beispiel das Datenfeld *BANFN*, weil dadurch die Spur einem Beschaffungsprozess zugeordnet werden kann. In Abbildung 2.3 sind die beiden ersten Positionen mit den Nummern 00010 und 00020 der Spur mit der Prozessinstanznummer 0010015263 zugeordnet. Dabei bilden die beiden Positionen ein Ereignis der Bestellanforderung.

Letztendlich vereinigt eine Logdatei eine Menge der vorhandenen Spuren, die ihrerseits eine Menge von Ereignissen vereinigen. Generell können Spuren verschiedener Prozesse in einer Logdatei zusammengefasst werden. In dieser Arbeit liegt der Fokus auf den homogenen Logdateien, demzufolge stellt eine Logdatei eine Menge der Ausführungen genau eines Prozesses dar. Eine Ereignisdatei beinhaltet mindestens eine Prozessinstanz. Jede Prozessinstanz hat mindestens ein Attribut, die Prozessinstanznummer. Weiterhin hat jede Prozessinstanz Ereignisse. Diese beinhalten eine Ereignisnummer, durch welche sie eindeutig zu identifizieren sind. Die Ereignisse können weitere Attribute haben. Die Ressource und der Zeitstempel sind dabei besonders für die Prozessmodellierung relevant [AWM11].

Im Vorfeld der Anwendung von Process-Mining-Methoden auf eine Logdatei muss diese in ein einheitliches Format transformiert werden. Zu diesem Zweck hat die Forschungsgruppe Architecture of Information Systems (AIS) von der Technischen Universität Eindhoven das Format *Magic eXtensible Markup Language* (MXML) im Jahr 2003 entwickelt [Aal05, AWM11]. Weiterhin ist im Jahr 2009 das Format *eXtensible Event Stream* (XES) entstanden [C.09]. Dieses beinhaltet einige Ver-

besserungen gegenüber MXML. Insbesondere das Problem mit der Semantik der Attribute, welche als eine einfache Zeichenkette in MXML gespeichert werden und keine weitere Bedeutung haben, ist in dem XES-Format gelöst [AWM11]. Deshalb wird in dieser Arbeit das XES-Format in der Version 1.0 verwendet. Dieses wird im Abschnitt 2.2.3 erläutert.

2.2.3 XES-Format

Das XES-Format basiert auf dem XML-Format und bietet ein einheitliches Format zur Repräsentation einer Logdatei [C.09, AWM11, Aal10]. Die Homogenität wird durch die klar definierte Struktur des Formats erreicht. Die Struktur ist in Abbildung 2.5 dargestellt. Dabei stellt das Element *Log* das erste Element in einer

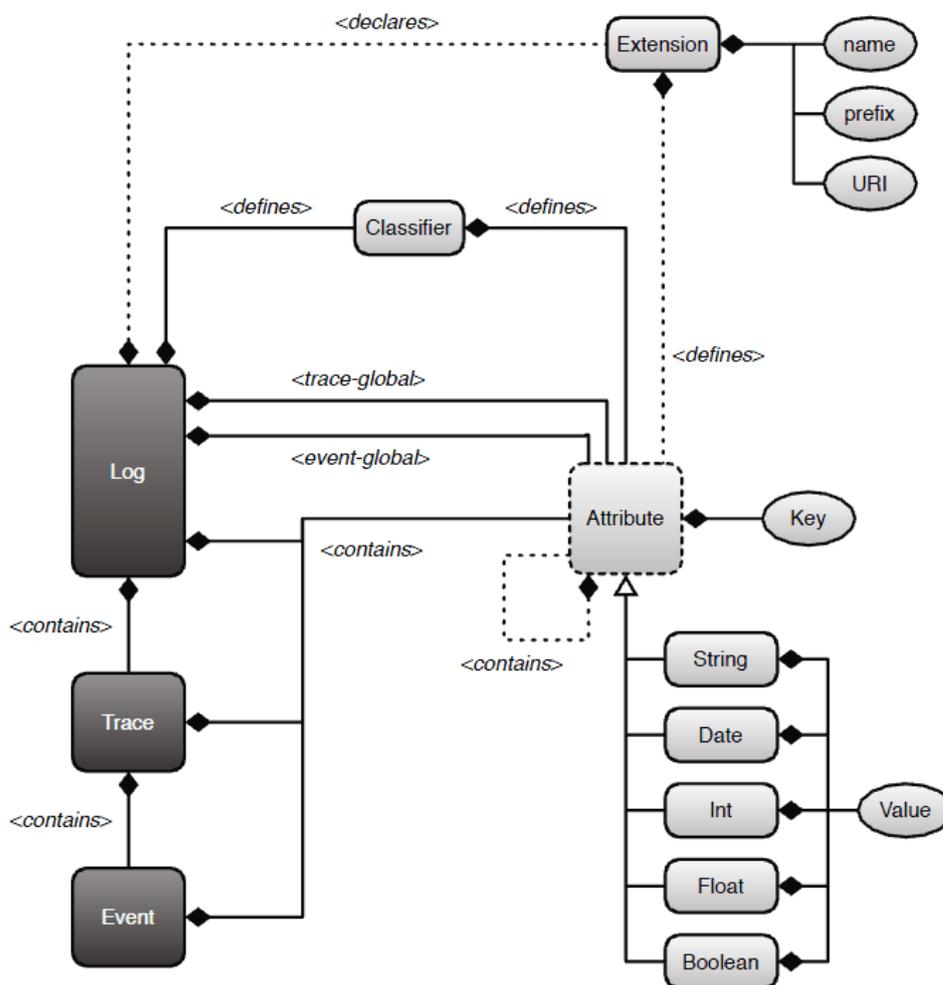


Abbildung 2.5: Das XES-Metamodell [C.09]

XES-Logdatei dar. Dieser kann Eigenschaften in Form von Attributen und *Extension* enthalten. Ferner beinhaltet das Element die *Trace*, welche eine Spur in einer Logdatei repräsentiert, die ihrerseits das Element *Event* beinhaltet. Der *Event* stellt die Ereignisse in einer Logdatei dar. Neben dem *Log* können die *Trace* und der *Event* Eigenschaften in Form von Attributen aufweisen.

Gegenüber dem MXML-Format erlaubt das XES-Format sowohl die Benutzung der vordefinierten als auch der frei definierbaren Attribute. Das semantische Definieren der Attribute geschieht durch die Verwendung von *Extension*. Dabei wurden bereits sechs der *Extension* als *Standard Extension* vordefiniert und stehen somit zur Verfügung. Das oft verwendete *Extension* ist das *Concept Extension*, welches den Namen der Spuren oder der Ereignisse definiert. Das in Abbildung 2.4 vorgestellte Ereignismodell ist für jedes Ereignis durch die *Lifecycle Extension* abgebildet. Die Organisationsperspektive findet sich in dem *Organizational Extension* wieder. Die Zeit ist durch die *Time Extension* abgebildet. Durch das *Classification Extension* können die Ereignisse zu einem logischen Konzept in einer XES-Logdatei gruppiert werden. Weiterhin kann ein Ereignis zur selben Zeit in mehreren Konzepten beinhaltet werden. Diesbezüglich erlaubt das *Semantic Extension* das Beschreiben dieser Ereignisse. Des Weiteren können für jede Ebene der Logdatei Attribute definiert werden. Diese haben die übliche XML-Repräsentation und die üblichen Typen, wie sie in Abbildung 2.5 dargestellt sind. Ein Ausschnitt aus einer Beispieldatei wird in Abbildung 2.6 vorgestellt. Hier ist eine Logdatei mit einer Spur und zwei Ereignissen

```

<log>
  <trace>
    <string key="concept:name" value="Case3.0"/>
    <event>
      <string key="org:resource" value="UNDEFINED"/>
      <date key="time:timestamp" value="2008-12-09T08:20:01.527+01:00"/>
      <string key="concept:name" value="A"/>
      <string key="lifecycle:transition" value="complete"/>
    </event>
    <event>
      <string key="org:resource" value="UNDEFINED"/>
      <date key="time:timestamp" value="2008-12-09T08:21:01.527+01:00"/>
      <string key="concept:name" value="E"/>
      <string key="lifecycle:transition" value="complete"/>
    </event>
  </trace>
</log>

```

Abbildung 2.6: Ein Ausschnitt einer XES-Logdatei

abgebildet. Die Spur hat ein Attribut, welches den Namen darstellt. Dies entspricht einer *Concept Extension* von dem Typ *String* mit dem entsprechenden Wert. Die beiden Ereignisse haben eine *Organizational Extension*, wobei diese die Ressource darstellt, eine *Time Extension*, eine *Concept Extension* und die *Lifecycle Extension*. Dabei hat diese den Typ *Complete*.

2.2.4 ProM-Vorstellung

Die erste Version von dem in dieser Arbeit verwendeten Werkzeug *ProM* ist eine Entwicklung aus dem Jahr 2004 [Aal10, AWM11]. Ein Fortschritt des Open-Source-Werkzeugs von 29 Plugins bis hin zu der Version ProM 6 aus dem Jahr 2010 mit über 300 Plugins ist zu konstatieren. Dabei hat sich das Eingabeformat von MXML bis hin zu dem standardisierten Format XES von dem *Institute of Electrical and Electronics Engineers* (IEEE), gewandelt. Der primäre Gedanke bei der Entwicklung ist, ein Werkzeug zu konzipieren, welches weg von der bis dahin gewohnten datenzentrischen Arbeitsweise zur prozessorientierter Arbeitsweise übergeht [Aal10, AWM11, J.10].

Die über 47 von den 300 vorhandenen Plugins erlauben die besonders für diese Arbeit relevante Erzeugung von Geschäftsprozessmodellen. Abbildung 2.7 zeigt das Ergebnis von der Analyse einer Logdatei mit dem *Heuristic Miner* Plugin. Die

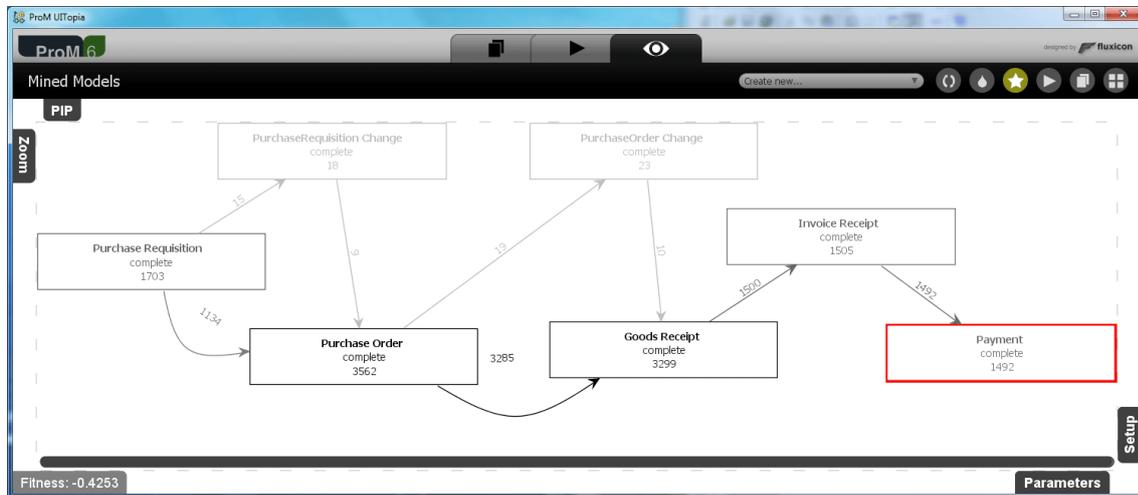


Abbildung 2.7: Analyse mit *Heuristic Miner*

Rechtecke repräsentieren die Ereignisse, wobei die Zahl im Rechteck anzeigt, wie oft ein Ereignis eingetreten ist. Wie oft der Übergang eingetreten ist, zeigt die Zahl an den gerichteten Kanten. Gleichzeitig zeigen diese die Richtung an, in welcher der Übergang stattfindet.

2.3 SAP ERP und Process Mining

Nachdem in den vorherigen Abschnitten die Grundlagen von SAP ERP und *Process Mining* im Vordergrund standen, beschäftigt sich dieser Abschnitt mit den in SAP vorhandenen und den für das *Process Mining* benötigten Informationen. Dabei wird erläutert, auf welchem Wege die Informationen aus SAP extrahiert und *ProM* zur Verfügung gestellt werden.

2.3.1 Diskussion

Der syntaktische Aufbau einer XES-Datei ist bereits in Abschnitt 2.2.3 erläutert. Zusätzlich ist der semantische Aspekt einer Logdatei, der die unterschiedlichen Perspektiven einer Process-Mining-Analyse ermöglicht, zu berücksichtigen. Dazu ist die Auswahl der Informationen aus einem SAP-ERP-System entscheidend. Um z.B. die Organisationsperspektive zu analysieren, müssen die Ereignisse ein Attribut beinhalten, das die Organisation widerspiegelt, weil dadurch ein Ereignis einer Organisation zugeordnet werden kann. Für die Leistung einer Prozessausführung müssen z.B. die Attribute vorhanden sein, die den zeitlichen oder den mengenmäßigen Durchsatz wiedergeben, weil dadurch analysiert werden kann, wie schnell ein Prozess ausgeführt wurde und welche Menge dabei umgesetzt wurde. Diese Arbeit konzentriert sich auf das Erfassen der Geschäftsprozessmodelle aus den Prozessausführungen in einem

SAP-System. Dazu ist in Abbildung 2.8 die Struktur einer Logdatei im XML-Schema dargestellt, die auf die vorhandenen Daten in SAP ERP unter Berücksichtigung der Geschäftsprozessmodellierung angewendet wurde. Dabei beinhaltet eine Logdatei

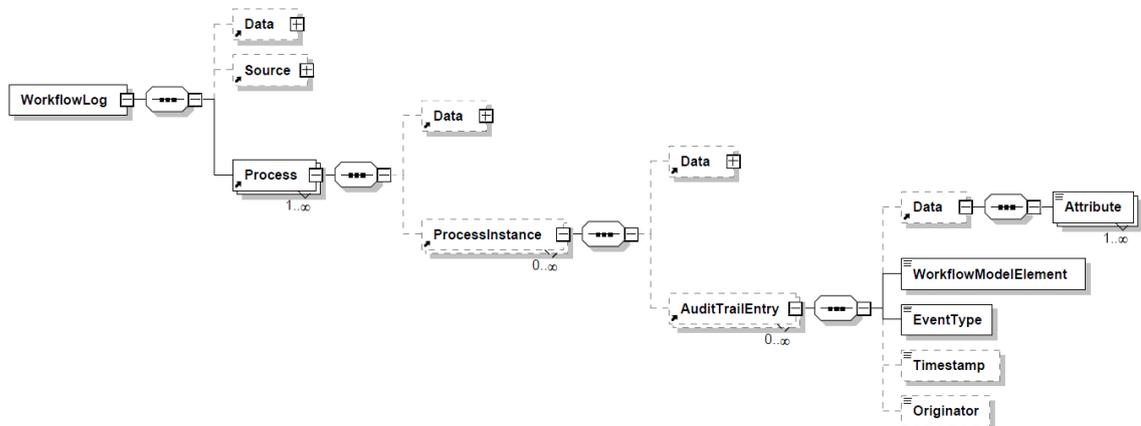


Abbildung 2.8: Der Aufbau einer Logdatei angewendet auf SAP ERP

einen homogenen Prozess, der aus Spuren aufgebaut ist, wobei die Homogenität die Darstellung eines Prozesses eines Typs bedeutet. Das oberste Element *Process*, welches den Prozess darstellt, wird durch die Eigenschaften in Form von Attributen beschrieben, wobei eine Eigenschaft z.B. die Quelle sein kann. Das nächste Element, die *Processinstance* gruppiert eine Menge der Ereignisse und wird ebenfalls durch die Eigenschaften in Form von Attributen dargestellt. Diese Eigenschaften beschreiben vor allem die Zuordnung einer Spur zu einem Prozess, wobei sie diesbezüglich eine eindeutige Instanznummer beinhalten. Weiterhin sind an die zu einer Spur gehörenden Ereignisse (in Abbildung *AuditTrailEntry*) Mindestanforderungen für die Geschäftsprozessmodellierung mit *Process Mining* gestellt. So müssen diese einen Zeitstempel (in Abbildung *Timestamp*) zur Einordnung des Ereignisses sowie eine Ressource beinhalten.

Eine Spur entspricht in SAP einem Belegfluss (siehe Abbildung 2.9). Jeder Beleg

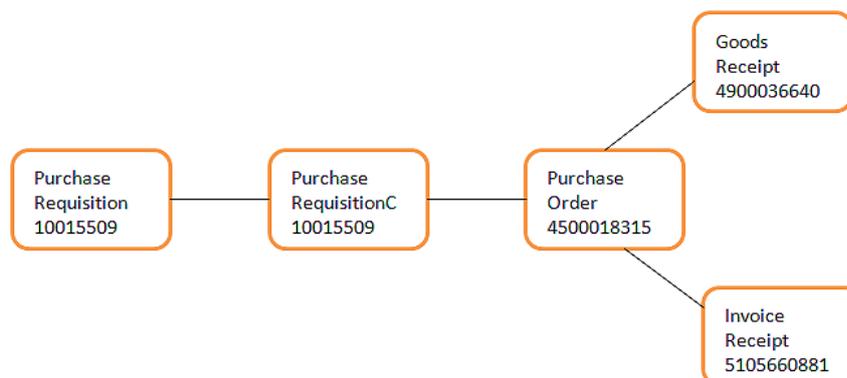


Abbildung 2.9: Beispiel:Belegfluss

in einem Belegfluss ist durch eine eindeutige Belegnummer identifiziert und entspricht einem Ereignis. Im Beispiel ist dem Beleg „Purchase Order“ die Belegnum-

mer 4500018315 zugeordnet. Die Informationen zu den Belegen werden in Tabellen gespeichert. Durch die Verknüpfung dieser Tabellen ist eine Spur konstruierbar, die einer Sammlung von zueinander gehörenden Belegnummern entspricht. Die Mindestanforderungen für die Prozessmodellierung mit *Process Mining* an die Attribute werden in einem SAP-System erfüllt, indem der Zeitstempel und die Ressource in Tabellen, die zu dem entsprechenden Ereignis gehören, abgelegt werden.

Die Vollständigkeit und die Qualität der Daten bestimmen die Anwendbarkeit von *Process Mining* maßgeblich. Dabei steht im Fall von Geschäftsprozessen in SAP die Vollständigkeit der identifizierten Ereignisse eines Prozesses im Vordergrund. Falls die Vollständigkeit nicht gegeben ist, kann nicht garantiert werden, dass das Ergebnis einer Analyse ein Modell erzeugt, welches den analysierten Geschäftsprozess wieder gibt. Eine wichtige Frage ist daher, wie die zu einem Prozess gehörenden Ereignisse identifiziert werden können und in welchen Tabellen die dazugehörigen Daten abgelegt werden. Ferner sichert die Qualität, dass die Daten der Datenbereinigung unterzogen werden und somit z.B. die Duplikate entfernt werden.

Durch die unterschiedlichen Geschäftsmodelle führen Unternehmen individuelle Geschäftsprozesse aus. Dies hat zur Folge, dass ein SAP-System für jedes Unternehmen individuell zugeschnitten wird, wobei das individuelle Customizing dazu führt, dass die Menge der Ereignisse, die in einer Prozessausführung stattfinden, ebenfalls individuell ist. Deshalb muss für jedes Unternehmen im Einzelnen untersucht werden, welche Prozesse welche Ereignisse beinhalten. Dazu existiert kein standardisiertes und automatisiertes Vorgehen [J.10, GM04]. Es kann wie bei dem Customizing eines SAP-Systems vorgegangen werden, wobei hier die *Business Blueprints* verwendet werden [SBb]. Diese geben Aufschluss darüber, welche Prozesse in einem Unternehmen implementiert werden. Dabei beinhalten die *Business Blueprints* die Strukturelemente Organisationseinheiten, Stammdaten, Geschäftsszenarien, Geschäftsprozesse und Prozessschritte. Durch die Benutzung dieser Dokumente werden die Prozessausführungen und die dazugehörigen Ereignisse sowie Daten aufgezeigt. Ferner bietet das *Easy Access Menu* einen Einblick in die Unternehmensstruktur. In Abbildung 2.10 ist ein expandiertes Menü dargestellt, in dem die Komponente *Logistics* und vor allem das Modul *Materialmanagement* (MM) sichtbar sind. Außerdem ist der Prozess der Bestellanforderung (*Purchase Requisition*) dargestellt. Hier kann die Bestellanforderung z.B. erzeugt, geändert oder angezeigt werden. Bei dieser Vorgehensweise werden Prozesse und Ereignisse aufgezeigt, die von den Benutzern durch das Ausführen der Transaktionen ausgeführt und erzeugt werden.

Weiterhin ist mit den Benutzern des Systems und den SAP-Beratern die Istvorgehensweise einzugrenzen. Dabei stehen die Benutzer des Systems für die von ihnen ausgeführten Vorgänge. Aus diesem Grund können sie vor allem einen Einblick in die Aktivitäten gewähren, die von dem System nicht erfasst werden, wie z.B. das Archivieren der gedruckten Belege.

Ferner werden bei dem Customizing des Systems von einem Berater, die von SAP entwickelten Referenzmodelle eingesetzt. Dabei beschreibt ein Referenzmodell das Leistungsspektrum eines SAP-Systems [SB11a]. Diese Modelle werden *SAP Best Practices* genannt, wobei die Unternehmen in der Praxis häufig nicht das SAP-

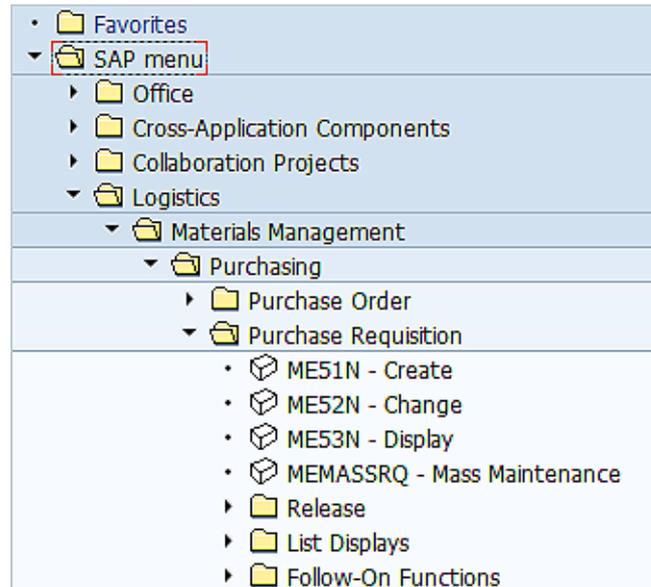


Abbildung 2.10: *Easy Access Menu*

System an die Unternehmensprozesse anpassen, sondern ihre Geschäftsprozesse an die Referenzmodelle. Aus diesem Grund sind die Referenzmodelle eine grundlegende Dokumentation, die Aufschluss darüber gibt, wie die Prozessschritte und die dazugehörigen Ereignisse in einem Unternehmen aussehen. Deshalb werden in Abschnitt 2.3.2 die Struktur der Referenzmodelle und eine Anleitung erläutert, wie diese bei der Identifikation der Daten für *Process Mining* in einem SAP-ERP-System in dieser Arbeit benutzen werden.

2.3.2 SAP Referenzmodell

Zunächst sind die Referenzmodelle dafür zuständig, um die von SAP standardisierten Module zu beschreiben. Falls z.B. ein Unternehmen die Vorbereitungen für das Einsetzen eines MM-Moduls trifft, ist ein dazugehöriges Referenzmodell die Grundlage für das Verstehen, wie das Modul aufgebaut ist. Im nächsten Schritt wird das System an ein Unternehmen angepasst, so dass die *Business Blueprints* entstehen. Ein Referenzmodell beinhaltet verschiedene Modelltypen und ist dazu aus Modellierungsobjekten aufgebaut. In Abbildung 2.11 sind die Modellierungsobjekte eines SAP-Referenzmodells dargestellt, wobei in dieser Arbeit insbesondere die Modellierungsobjekte *Szenario*, *Prozess*, *Business Objekt* und *Ereignis* relevant sind, weil diese die erforderlichen Informationen für *Process Mining* darstellen. Die gestrichelten Linien in Abbildung weisen auf den Aufbau eines zusammengesetzten Objektes hin. Beispielsweise setzt sich ein Prozess aus Funktionen und Ereignissen zusammen. Eine durchgezogene Linie beschreibt eine Zuordnung zu einem Objekt. In Abbildung 2.11 sind einem Modellierungsobjekt *Komponente* Modellierungsobjekte *Business Objekt* und *Prozeß* zugeordnet [SB11a].

Bei der Einführung eines SAP-Systems werden insbesondere die Modelltypen, die Komponentenhierarchie, das Prozessmodell und das Business-Objekt-Modell ver-

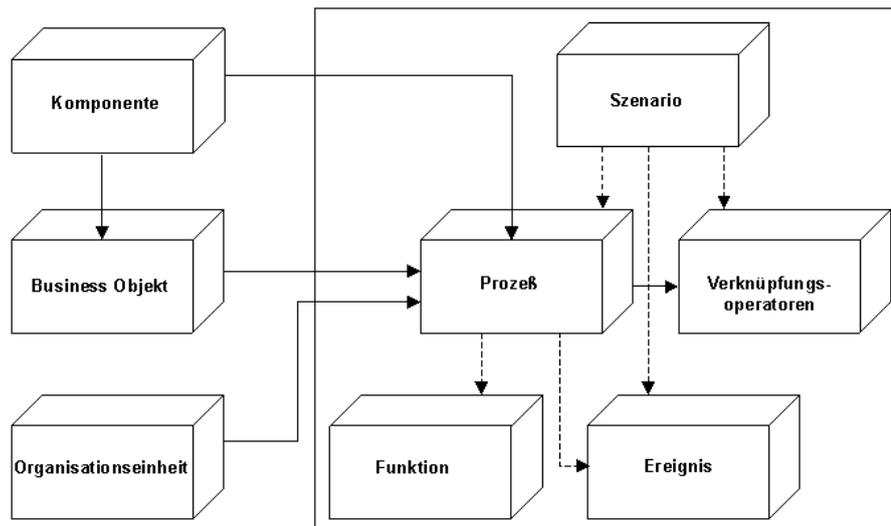


Abbildung 2.11: Struktur eines SAP-Referenzmodells [SB11a]

wendet. Wie der Name bereits suggeriert, beschreibt die Komponentenhierarchie die in dem System implementierten Komponenten. Das Business-Objekt-Modell beschreibt die Objekttypen und deren Verknüpfung untereinander, die in einem Prozessschritt verwendet werden [M.08, SBd]. Z.B. sind die Business-Objekte *Material* und *Lieferant* miteinander verknüpft und werden in dem dazugehörigen Business-Objekt-Modell beschrieben. Ferner beschreibt das Prozessmodell den Prozess mit dazugehörigen Prozessschritten. Ein Beispiel eines Prozessmodells des Beschaffungsprozesses in SAP ist in Form einer Wertschöpfungskette in Abbildung 2.12 dargestellt. Hier sind die Schritte die Bestellanforderung, die Bestellung, der Warenein-

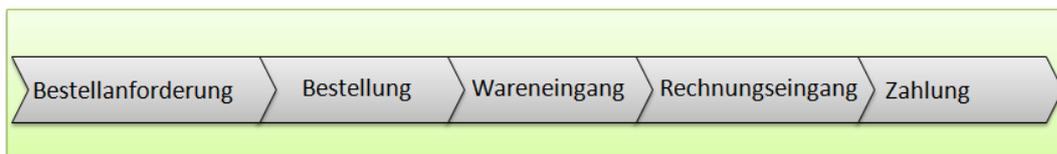


Abbildung 2.12: Wertschöpfungskette des Beschaffungsprozesses

gang, der Rechnungseingang und die Zahlung des Beschaffungsprozesses (in SAP *Procure to Pay* (PTP) genannt) dargestellt. Wenn ein Prozessschritt eines Prozesses ausgeführt wird, werden Informationen dazu benötigt oder bereitgestellt. Diese Informationen sind von Objekttypen repräsentiert [SBa, M.09]. Während eines PTP-Prozesses werden z.B. Objekttypen wie der Debitor oder das Material benötigt, wohingegen z.B. die Bestellanforderung in der Bestellung bereitgestellt wird. In Abbildung 2.13 sind Ausschnitte von zwei Business-Objekt-Modellen dargestellt. Das Modell 1 stellt den Business-Objekttypen der Bestellanforderung dar, wobei das Modell 2 die Bestellung darstellt. Die Beziehung unter den Modellen ist manuell eingefügt, weil die Bestellanforderung im ersten und in Bezug zu einer Bestellanforderung die Bestellung im darauffolgenden Prozessschritt erzeugt werden. Da die Prozessschritte in dem Prozessmodell nacheinander folgen, ist zu folgern, dass zwi-

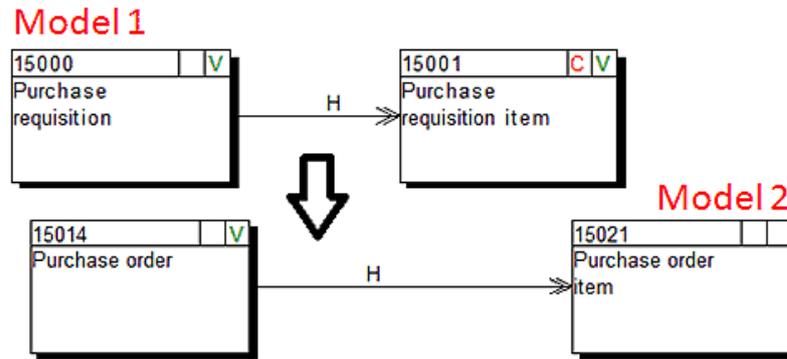


Abbildung 2.13: Bestellanforderung und Bestellung

schen den Business-Objekt-Modellen eine Beziehung besteht.

Ein weiteres Beispiel eines Prozessmodells in SAP ist das Prozessmodell des Prozesses *Schaden-/Leistungsabwicklung für Lebensversicherung* aus dem Modul *FS-CM-Schaden-/Leistungsmanagement* der Komponente *SAP for Insurance*. Abbildung 2.14 zeigt das Prozessmodell des Schaden-/Leistungsabwicklungsprozesses in Form einer Wertschöpfungskette, wobei im Anhang A das zugehörige Geschäftsprozessmodell in *Business Process Modeling Notation* (BPMN) dargestellt ist. Das

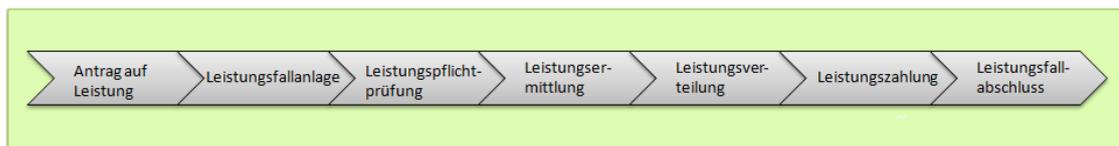


Abbildung 2.14: Wertschöpfungskette des Schaden-/Leistungsabwicklungsprozesses

Geschäftsprozessmodell besteht aus den Schritten dem Antrag auf Leistung, der Leistungsfallanlage, der Leistungspflichtprüfung, der Leistungsermittlung, der Leistungsverteilung, der Leistungszahlung und dem Leistungsfallabschluss, wobei die Schritte weitere Prozessabläufe beinhalten. Diese sind im Anhang B gezeigt [SBh]. Im ersten Schritt werden die Anträge auf Leistungen in einem Versicherungsunternehmen entgegengenommen und auf die Falldubletten geprüft. Nach der Prüfung wird in dem Prozessschritt Leistungsfallanlage in der Sparte Lebensversicherung ein Fall angelegt. Dazu wird in der Leistungspflichtprüfung ermittelt, inwieweit die Versicherung leistungspflichtig ist. Falls die Ermittlung ergibt, dass das Unternehmen leistungspflichtig ist, wird in der Leistungsermittlung der Gesamtbetrag berechnet, wobei in der Leistungsverteilung der Betrag auf einzelne Positionen verteilt wird, falls diese existieren. Anschließend wird die Bezahlung in der Leistungszahlung angestoßen und abschließend der Fall in dem Leistungsfallabschluss geschlossen. Die zugehörigen Business-Objekt-Modelle sind nicht frei zugreifbar, weil dazu die Komponente *SAP for Insurance* von SAP zusätzlich erworben werden muss und in dieser Arbeit nicht zur Verfügung steht.

2.3.3 Log-Daten

Zu Beginn eines Process-Mining-Analyseprojektes müssen das Modul und daraufhin der Geschäftsprozess, der mit Process-Mining-Methoden analysiert werden soll, ausgewählt werden. Nachdem die Frage nach den Prozessschritten eines ausgewählten Prozesses beantwortet ist, müssen die zugehörigen Objekttypen identifiziert werden. Auf der untersten Applikationsebene werden die Informationen zu diesen Objekttypen in Tabellen abgelegt, so dass der nächste Schritt in der Identifizierung der zugehörigen Tabellen liegt. Der *Object Navigator* (Transaktion: SE80) ist ein Werkzeug zum Anlegen, Ändern und Verwalten von Entwicklungsobjekten, wobei in Abbildung 2.15 der *Einkauf* als Entwicklungsobjekt dargestellt ist. Das in dieser Arbeit ebenfalls benötigte Entwicklungsobjekt *Bezahlung* ist im Anhang C gezeigt. In dem

Object Name	Description
▼ Database Tables	
• A160	Plant Info Record: Variants
• A161	Info Record: Variants
• AMPL	Table of Approved Manufacturer Parts
• AMPL REJECTION	Rejection of AMPL Row
• EBAN	Purchase Requisition
• EBKN	Purchase Requisition Account Assignment
• EBUB	Index for Stock Transport Requisitions for Material
• EINA	Purchasing Info Record: General Data
• EINE	Purchasing Info Record: Purchasing Organization I
• EIPA	Order Price History: Info Record
• EKAB	Release Documentation
• EKAN	Vendor Address: Purchasing Document
• EKBE	History per Purchasing Document
• EKBEH	Removed PO History Records
• EKBZ	History per Purchasing Document: Delivery Costs
• EKBZH	History per Purchasing Document: Delivery Costs
• EKCPA	Purchasing Contract Commitment Plan Account A
• EKCPH	Purchasing Contract Commitment Plan Header Info

Abbildung 2.15: *Object Navigator*

expandierten Baum gibt der Knoten *Database Tables* den Aufschluss darüber, welche Tabellen das Paket *Einkauf* beinhaltet, wobei das Paket den Beschaffungsprozess in SAP beschreibt. Dabei sind die Tabellen, die Änderungen protokollieren, in dieser Ansicht nicht einsehbar, weil sie allgemeingültig sind und keinem Paket angehören. Deshalb ist die Ansicht nicht vollständig. An dieser Stelle wird das zuvor beschriebene Vorgehen relevant. Die SAP-Dokumentation gibt Aufschluss darüber, dass Änderungen in den Tabellen *CDHDR* und *CDPOS* abgelegt werden. Dabei protokollieren *CDHDR* den Änderungsbelegkopf, wobei zu jedem Änderungsbelegkopf die Änderungsbelegpositionen in *CDPOS* protokolliert werden.

Kapitel 3

Realisierung

Nachdem im vorangegangenen Kapitel die Struktur und die theoretischen Grundlagen von SAP ERP und *Process Mining* im Vordergrund standen, widmet sich dieses Kapitel der Realisierung des Verfahrens zum Erzeugen einer XES-Logdatei, die anschließend mit Process-Mining-Methoden analysiert wird. Dabei werden die Implementierung und die dazu getroffenen Entscheidungen erläutert. Das Kapitel folgt dem Aufbau des letzten Kapitels und beschreibt die Möglichkeiten seitens eines SAP-Systems, gefolgt von der Anbindung an das ProM. Der letzte Teil dieses Kapitels befasst sich mit der Benutzeroberfläche, der in dieser Arbeit entwickelten Anwendung und den dort gegebenen Möglichkeiten.

3.1 Ziel

Die vorangegangenen Kapitel machen deutlich, dass aktuell kein Werkzeug vorhanden ist, welches einen Businessanalysten dabei unterstützt aus den Ausführungen der Anwender eines Geschäftsprozesses in einem SAP-System ein Geschäftsprozessmodell zu gewinnen, um dieses analysieren zu können. Deshalb ist das Ziel dieses Projektes eine Anwendung zu entwickeln, die den Businessanalysten unterstützt. Abbildung 3.1 veranschaulicht das Projektziel, wobei die Anwendung im Bild an der Schnittstelle zwischen dem SAP-System und dem Werkzeug *ProM* steht. Der Input

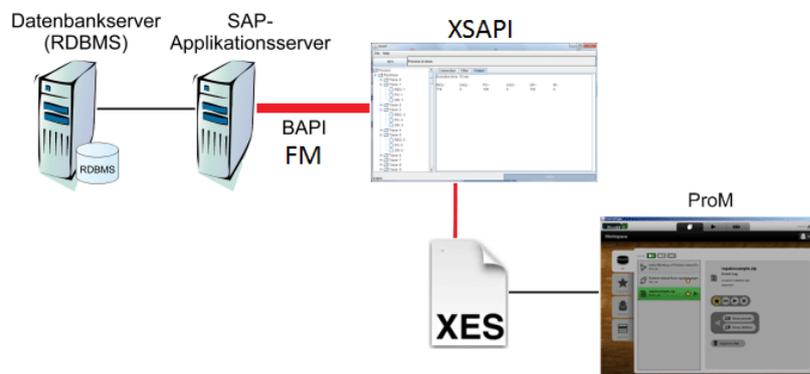


Abbildung 3.1: Projektziel

für die Anwendung sind die Daten der Geschäftsprozesse, welche in den Tabellen eines SAP-Systems während einer Prozessausführung abgelegt werden.

Zu weiteren Zielen der Anwendung gehört die Extraktion der Daten aus einem Datenbanksystem eines SAP-Systems sowie die Transformation und das Laden der Daten in eine XES-Logdatei. Dabei erfolgt der Zugriff auf das Datenbanksystem über den SAP-Applikationsserver.

Ferner ist das Ziel eine einfache Bedienung des Prototyps zu ermöglichen, so dass von einem Anwender keine Programmierkenntnisse verlangt werden. Weitere Zielsetzung sind die Filterfunktionen und die Visualisierung. Dabei ermöglichen die Filter eine Logdatei zu extrahieren, die z.B. die Prozessausführungen eines bestimmten Monats oder einer Filiale widerspiegeln. Die Visualisierung gibt dem Businessanalysten die Möglichkeit die extrahierten Daten vor der Erzeugung einer XES-Logdatei zu sehen und falls nötig ist, die Extraktion z.B. zu wiederholen.

Letztendlich ist die Zielsetzung der Anwendung den Belegfluss eines beliebigen Geschäftsprozesses aus einem SAP-ERP-System zu extrahieren und diesen in eine XES-Logdatei zu transformieren. Dadurch wird der Input für das *ProM* erzeugt. Der Fokus der Extraktion in dieser Arbeit liegt auf den Daten des Beschaffungsprozesses, die die anschließende Gewinnung der Geschäftsprozessmodelle dieses Prozesses mit dem *ProM* ermöglichen, wobei die Anwendung für alle Prozesse aus SAP ERP konfiguriert werden kann. Dazu sind mehrere Schritte und Entscheidungen notwendig, die in den folgenden Abschnitten erläutert werden.

3.2 Vorbereitungsphase

Bei der Einführung eines SAP-Systems in einem Unternehmen wird dieses an die Geschäftsprozesse des Unternehmens angepasst. Dadurch ist in diesen Unternehmen ein individuelles SAP-System, mit einzigartigen Geschäftsprozessen vorhanden. Infolgedessen wird die in dieser Arbeit entwickelte Anwendung während eines Process-Mining-Projekts an das SAP-System angepasst. Dazu ist vor allem die in Kapitel 2.3 beschriebene Vorgehensweise zur Identifikation der Prozesse, die analysiert werden sollen sowie den zugehörigen Objekttypen und Tabellen durchzuführen. Anschließend ist die Vorbereitungsphase auf der Datenebene vorzunehmen.

Nachdem die benötigten Vorbereitungen in der Applikationsebene getroffen worden sind, ist im nächsten Schritt die Beziehung zwischen den Tabellen in der Datenebene herzustellen. Das Diagramm in Abbildung 3.2 zeigt die Beziehung der EBAN-Tabelle, welche die Daten einer Bestellanforderung protokolliert zu den Tabellen aus der Bestellung, der EKKO (Einkaufsbelegkopf) und der zugehörigen EKPO-Tabelle (Einkaufsposition) sowie zu der EKBE-Tabelle (Historie zum Einkaufsbeleg) an. Die Verbindungslinien bilden die Beziehung zwischen den Tabellen ab, wobei die Werte an den Linien die Kardinalität angeben. Die Kardinalität zwischen der EBAN- und der EKPO-Tabelle ist mit 1:N angegeben. Die Eins bedeutet, dass eine bestimmte Bestellung zur gleichen Zeit nur in einer Bestellanforderung ausgeführt sein kann. Dagegen bedeutet das N, dass einer Bestellanforderung mehrere Bestellungen zugeordnet sein können. Die Verknüpfung der Tabellen ist in dem Werkzeug *Object Navigator* in SAP ERP einsehbar. Hier kann das



Abbildung 3.2: Tabellenverknüpfung

Entity-Relationship-Diagramm (ERD) für ausgewählte Tabellen aus dem ABAP Dictionary (*ABAP Dictionary* → *Database Tables* → *Table Name* → *Execute (F8)* → *Doppelklick auf den Tabellennamen* → *Graphic* → *Zugehörige Tabellen auswählen* → *Copy*) extrahiert werden.

Des Weiteren ist in Abbildung 3.3 die Fremdschlüsselbeziehung zwischen der EKPO- und der EKBE-Tabelle dargestellt. Die EKPO-Tabelle interagiert in diesem Fall

Check table			
EKPO			
Foreign Key Fields			
Check table	ChkTabFld	For.key ta...	Foreign Key Field
EKPO	MANDT	EKBE	MANDT
EKPO	EBELN	EKBE	EBELN
EKPO	EBELP	EKBE	EBELP

Abbildung 3.3: Fremdschlüsselbeziehung EKPO - EKBE

als eine Prüftabelle (sogenannte referierte Tabelle), dabei ist die EKBE-Tabelle die Fremdschlüsseltabelle (sogenannte abhängige Tabelle). Ein Datensatz ist in der EKPO-Tabelle durch die Primärschlüssel *MANDT*, *EBELN*, *EBELP* eindeutig identifiziert, gleichzeitig identifizieren diese Felder in Form der Fremdschlüssel einen Datensatz in der EKBE-Tabelle. Zufällig haben sie in beiden Tabellen die gleiche Bezeichnung, wobei dies nicht der Regelfall ist. Anhand dieser Beziehung können die Tabellen in SAP miteinander verknüpft werden.

Ferner sind bei der Vorbereitungsphase der Datenebene, Entscheidungen bezüglich der Granularität der Prozesse zu treffen. Die Granularität in einem Prozessschritt z.B. der Bestellanforderung lässt sich auf zwei Arten betrachten. Anstatt eines einzelnen Ereignisses, *Change PurchaseRequisition*, kann der Prozess detaillierter in folgende Ereignisse unterteilt werden *Delete PurchaseRequisition*, *Undelete PurchaseRequisition* und *Release PurchaseRequisition*. Diese drei Ereignisse entsprechen bei näherer Betrachtung dem Änderungsereignis bezüglich einer Bestellanforderung in SAP ERP. Dabei hängt die für die Analyse von Geschäftsprozessen notwendige Granularität der Prozesse von der gefragten Modellperspektive ab. Letztendlich beeinflusst die Granularität das Ergebnis maßgeblich, was sich in dem erzeugten Modell widerspiegelt.

Nachdem die Vorbereitungsphase abgeschlossen ist, ist der Schritt der Anbindung an ein SAP-ERP-System im Nachfolgenden zu bewerkstelligen. Dazu sind in dem Kapitel 3.3 die Details erläutert.

3.3 Anbindung an SAP

Letztendlich gilt es, eine Verbindung zu einem SAP-System aufzubauen und die in der Vorbereitungsphase identifizierten Informationen zu extrahieren. Dazu sind in SAP mehrere Schnittstellentechnologien vorhanden. In Abbildung 3.4 ist die Zugriffsarchitektur eines SAP-Systems dargestellt. Das Ziel ist die Extraktion der Da-

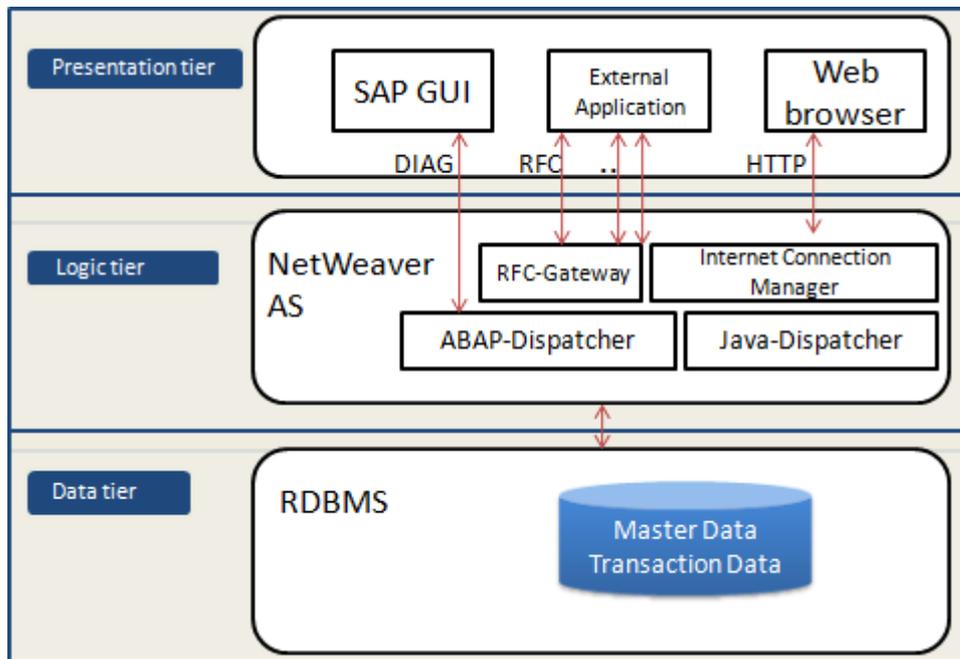


Abbildung 3.4: Dreischichtige SAP-Architektur [M.09]

ten aus der *Datatier*. Dabei stellt die *Datatier* ein oder mehrere relationale Datenbanksysteme dar. Hier sind die Stammdaten sowie Bewegungsdaten abgelegt, welche während der Prozessausführungen gespeichert werden. Die Informationen aus der *Datatier* sind in der *Präsentationstier* dargestellt und in der *Logictier* verarbeitet. Die *Logictier* beinhaltet den SAP NetWeaver Application Server, wobei dieser das Fundament des SAP NetWeavers ist. Der Server besteht aus Prozessen, welche die Programme des Systems, die entweder in ABAP oder Java sind, ausführen [M.09, P.01]. Dazu umfasst der Server zwei Technologien, den ABAP-Stack sowie den Java-Stack. Die *Präsentationstier* ist für die Eingaben der Benutzer, welche an die *Logictier* weitergeleitet werden, und die Darstellung der Ergebnisse verantwortlich. Dazu kann sowohl eine SAP-Benutzeroberfläche, ein Webbrowser, aber auch ein externes Programm verwendet werden.

3.3.1 Schnittstellentechnologien

Eine in SAP weit verbreitete Methode, Daten aus den Tabellen eines Datenbanksystems in *Datatier* zu erhalten, ist die **Transaktion SE16**. In dieser Transaktion ist zu spezifizieren, welche Tabellen unter welchen Bedingungen zu extrahieren sind. In Abbildung 3.5 ist ein Ausschnitt aus der RSEG-Tabelle (Belegposition Eingangs-

rechnung) dargestellt. Die Daten werden unkonvertiert als Tabellenkalkulation im

MANDT	BELNR	GJAHR	BUZEI	EBELN	EBELP	ZEKKN	MATNR
926	5105608792	2011	000001	4500018081	00010	00	101-110
926	5105608802	2011	000001	4500018095	00010	00	SCHALTUNG-1305
926	5105608802	2011	000002	4500018095	00020	01	SCHALTUNG-1305

Abbildung 3.5: Ausschnitt aus der RSEG-Tabelle

Rich-Text-Format oder in Hypertext Markup Language (HTML)-Format exportiert und können im Nachhinein in eine lokale Datenbank zu Weiterverarbeitung importiert werden. Da das Ziel dieser Arbeit die Entwicklung einer vollautomatischen Schnittstelle ist, wird diese Schnittstellentechnologie nicht angewendet. Der Nachteil dieser Technologie ist, dass die Tabellen manuell exportiert werden müssen. Zudem ist der Speicherplatz zum Exportieren begrenzt, so dass größeren Tabellen aufgeteilt werden müssen.

Eine weitere Technologie ist die **Application Link Enabling (ALE)-Technik** [M.09]. Diese bietet Dienste zum Austausch von Nachrichten zwischen verschiedenen Anwendungen sowohl intern als auch extern an. Dazu werden die Intermediate Documents (IDocs) verwendet. IDocs sind Dokumente, in denen die Daten einer Anwendung gespeichert und per ALE an andere Anwendungen gesendet werden [M.09, P.01, SB11b]. Ein IDoc besteht aus einem Header von 63 Byte Länge sowie den eigentlichen Daten mit bis zu 1.000 Byte Länge. Zudem muss im Rahmen eines ALE-Customizings festgelegt werden, welche Dokumente an welche Empfänger zu verschicken sind. Der Vorteil dieser Methode ist die automatische Benachrichtigung über ein stattgefundenes Ereignis, wodurch eine Online-Analyse ermöglicht wird. Der Nachteil, die bestehende Begrenzung der Datenlänge, welche für die Übertragung einer kurzen Nachricht zwischen zwei Prozessschritten und nicht der vollständigen Prozesse ausreichend ist, führt dazu, dass nicht gesichert ist, dass die Daten der Prozessausführungen ohne Verluste übertragbar sind. Dies kann im Customizing umgangen werden, indem definiert wird, dass ein IDoc mehrmals übertragen werden soll. Diesbezüglich bringt die Verwaltung der IDocs einen großen Aufwand mit sich, weil in dem System für jeden Prozess die IDocs konfiguriert werden müssen. Zudem haben nicht alle Anbieter den freien Zugang zu Customizing-Daten. Infolgedessen wird die Technologie nicht in dieser Arbeit verwendet.

In dieser Arbeit werden zwei Schnittstellentechnologien verwendet, die im Folgenden erläutert werden. Beide Technologien benutzen das von SAP entwickelte proprietäre Protokoll zum Aufruf von Funktionen, das Remote Function Call (RFC), wobei die zweite Technologie neben dem RFC objektorientierte Methodenaufrufe erlaubt. Über das RFC-Protokoll oder die Methodenaufrufe sind die beiden Schnittstellentechnologien von externen Programmen aufrufbar.

Die erste Schnittstellentechnologie sind die **Funktionsbausteine**, wobei diese die Prozeduren darstellen, die in SAP-Anwendungen innerhalb der ABAP-Programme

ausgeführt werden. Jedes ABAP-Programm wird modular aus den Verarbeitungsblöcken aufgebaut. Dabei sind die Verarbeitungsblöcke aus Programmteilen aufgebaut, welche von außerhalb eines ABAP-Programms oder aus Verarbeitungsblöcken des gleichen ABAP-Programms aufgerufen werden. Dazu bestehen die Programmteile aus Unterprogrammen, Funktionsbausteinen oder Methoden. Die Unterprogramme dienen der lokalen Modularisierung der ABAP-Programme. Dagegen dienen die Funktionsbausteine ausschließlich der globalen Modularisierung, das bedeutet, dass diese für die Aufrufe aus den anderen ABAP-Programmen zur Verfügung stehen. Die Methoden in einem Verarbeitungsblock beinhalten die Funktionalität der Klassen in ABAP-Objects, wobei die ABAP-Objects die neue Technologie ist, welche die Daten und die Funktionalität statt in Funktionsgruppen in Klassen definiert. Dabei stellt eine Funktionsgruppe ein Behälter für die zusammengehörigen Funktionsbausteine dar. Im Gegensatz zu den Unterprogrammen können die Funktionsbausteine die Daten lediglich über die fest definierten Schnittstellen austauschen, welche für die Kapselung und die Wiederverwendbarkeit von der Funktionalität in SAP verantwortlich sind. Die Schnittstelle eines Funktionsbausteins besteht aus den Importparametern, den Exportparametern, den Changing-Parametern, den Tabellen und den Ausnahmen. Die Importparameter dienen der Übergabe von Daten an die Funktionsbausteine. Falls z.B. ein Funktionsbaustein benutzt wird, welcher die Datensätze einer Tabelle ausliest, dann können die Importparameter den speziellen Feldern entsprechen, die ausgelesen werden sollen. Dagegen werden über die Exportparameter die Ausgaben eines Funktionsbausteins entgegengenommen. Die Changing-Parameter stellen eine Mischung der Import- und der Exportparameter dar, wobei die Changing-Parameter einem Funktionsbaustein die Daten übergeben, welche dieser bearbeitet und anschließend werden die Daten von den Changing-Parametern entgegengenommen. Weiterhin stellen die Tabellen, die Möglichkeit einem Funktionsbaustein per Referenz Arrays zu übergeben. Die Tabellenparameter dienen, wie die Changing-Parameter sowohl der Übergabe als auch der Rückgabe von Daten. Letztendlich werden die Ausnahmen zur Verarbeitung von Fehlern benutzt, die in einem Funktionsbaustein ausgelöst wurden.

Jeder Funktionsbaustein hat eine solche Schnittstelle. Sobald diese einmal vom Entwickler freigegeben wird, bleibt sie immer lauffähig und kann dabei ausschließlich erweitert und nicht verändert werden [M.09, SB11b]. Die Schnittstellen interagieren als eine Art Filter, weil dadurch die Ausgabe eingegrenzt wird. Die RFC, welche dem Aufruf von den Funktionsbausteinen dient, unterstützt dabei synchrone, asynchrone und transaktionale Aufrufe [SB11b].

Die zweite Technologie sind die **Business Application Programming Interfaces** (BAPIs) [SBa]. Diese stellen die Methoden der Business-Objekttypen dar, wobei die Business-Objekttypen in Kapitel 2.3.2 erläutert wurden. Ferner werden aus den Business-Objekttypen die Business-Komponenten in SAP ERP aufgebaut. Abbildung 3.6 zeigt die sich dabei ergebene Architektur. Die Business-Komponente *Logistik* wird z.B. aus den Business-Objekttypen *Kreditor* und *Kundenauftrag* aufgebaut. Des Weiteren werden die Objekttypen ebenfalls aus Komponenten aufgebaut. Dazu gehören die Grunddaten eines Objekttyps, welche die technischen Daten eines Objekttyps darstellen. Weiterhin sind die Interfaces ein Bestandteil der Objekt-

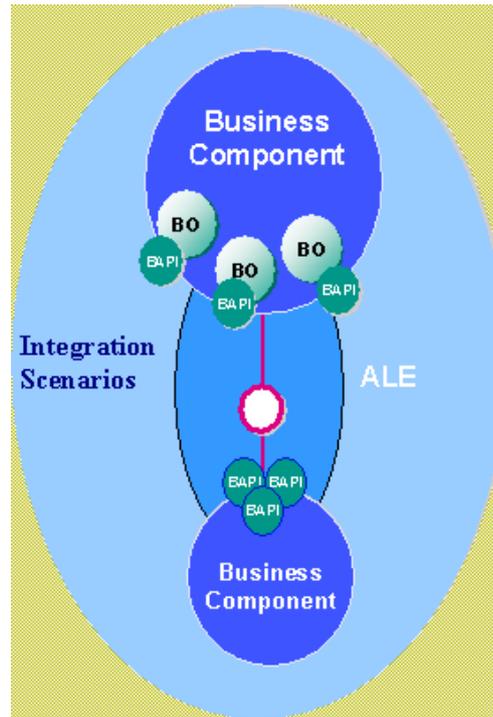


Abbildung 3.6: Der Aufbau der Komponenten in SAP ERP [SB11b]

typen, wobei diese die Schnittstellen repräsentieren, welche von den Objekttypen implementiert werden. Die Schlüsselfelder eines Objekttyps identifizieren eine Instanz des Objekttyps. Die Eigenschaften werden von den Attributen dargestellt, die ebenfalls zu den Komponenten eines Objekttyps gehören, wobei diese entweder die Referenzen auf die Datenbankfelder oder auf die anderen Business-Objekte beinhalten. Weiterhin rufen die Methoden eines Objekttyps die Transaktionen oder die Funktionsbausteine auf. Letztendlich gehören die Ereignisse ebenfalls zu den Komponenten eines Objekttyps. Dabei werden diese über die Nachrichten oder die Zustandsänderungen in SAP ERP ausgelöst.

Einige Methoden eines Objekttyps werden als BAPIs gekennzeichnet und ermöglichen dadurch den objektorientierten Zugriff auf ein SAP-ERP-System. Zusammen mit den Objekttypen stellen die BAPIs den Schnittstellenstandard auf betriebswirtschaftlicher Ebene dar. Dazu wird, wie in Abbildung 3.6 gezeigt ist, die ALE-Technik benutzt, um die Integration von Geschäftsprozessen zu ermöglichen, die über mehrere SAP- oder Fremdsysteme verteilt werden. Diesbezüglich beschreibt das *Integration Scenarios*, wie die Komponenten, die Business-Objekttypen und die BAPIs zusammenarbeiten.

In dieser Arbeit werden die BAPIs benutzt, um die Daten eines Prozesses aus SAP ERP zu extrahieren. Die BAPIs werden in SAP im *Business Object Repository* (BOR) definiert und werden als Funktionsbausteine implementiert, wobei ein BOR ein Repository darstellt, welches für die Speicherung der Business-Objekttypen und der zugehörigen BAPIs verantwortlich ist [SB11b, M.09]. Durch die Trennung der Definition von der Implementierung eines BAPIs ergeben sich in SAP zwei Möglichkeiten, wie auf die BAPIs zugegriffen werden kann. Diesbezüglich stellt Abbil-

dung 3.7 die beiden Möglichkeiten dar. Die erste Möglichkeit stellen die objektorien-

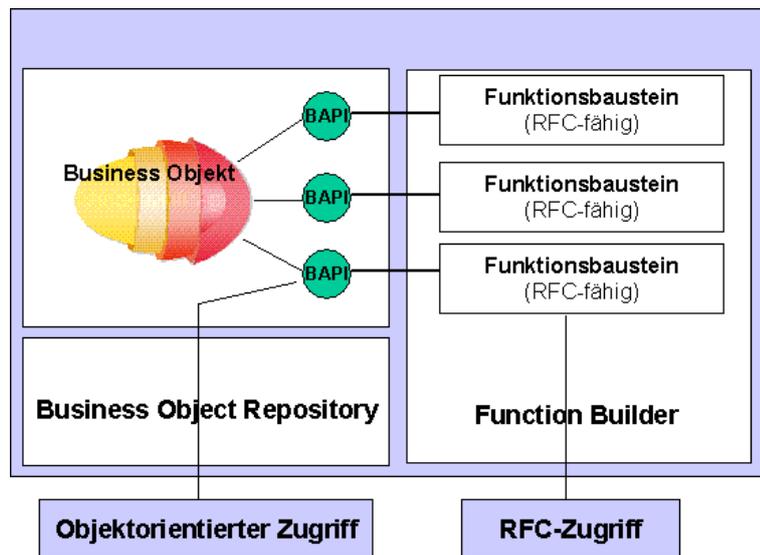


Abbildung 3.7: Zugriffsmöglichkeiten auf BAPIs [SB11b]

tierten Methodenaufrufe eines BAPIs im BOR. Dazu wird der *SAP Java Connector* (SAP JCo) benutzt, wobei dieser die Kommunikation zwischen Java-Anwendungen und SAP-Systemen erlaubt. Der SAP JCo wird im nachfolgenden Kapitel erläutert. Die zweite Möglichkeit stellen die RFC-Aufrufe der Funktionsbausteine, auf den die BAPIs basieren. Dazu wird entweder die RFC-Bibliothek oder die C/C++ Klassenbibliothek verwendet. Diese interagieren mit dem *Function Builder*, einem Werkzeug in SAP, welches für die Verwaltung der Bausteine verantwortlich ist, um einen RFC-Aufruf an die Funktionsbausteine zu übergeben.

Der Unterschied zwischen den BAPIs zu gewöhnlichen Funktionsbausteinen besteht darin, dass der Funktionsumfang eines BAPIs von SAP garantiert ist und in einem höheren Release nicht inkompatibel geändert wird [SAPb]. Weiterhin lösen die BAPIs keine Ausnahmen aus, weil zu diesem Zweck ausschließlich die Exportparameter der BAPIs verwendet werden [M.09]. Letztendlich stellt jedes Business-Objekt eine Liste der Standard-BAPIs zu Verfügung. Z.B. gibt das Standard-BAPI *GetList* eine Liste der zu dem BO gehörenden Daten wieder, deren Inhalt durch die zuvor dem BAPI übergebenen Importparameter eingegrenzt werden kann. Weiterhin liefert das BAPI *GetDail* nach zuvor übergebenen Importparametern detaillierte Informationen zu einem bestimmten Eintrag aus der Liste eines Objekttyps.

3.3.2 SAP Java Connector

Ein SAP NetWeaver-Applikationsserver ermöglicht den SAP-Systemen, außer der internen Integration, die Öffnung der Strukturen des Systems [M.09]. Der Bestandteil des Applikationsservers, der Internet Communication Manager (ICM), realisiert dabei die Kopplung zwischen dem SAP-System und dem Nicht-SAP-System. Die Integration Engine ist ihrerseits der Bestandteil der Integrationsschicht des ICM. Mit Hilfe von Adaptern verarbeitet die Integration Engine sowohl die IDocs als auch die

RFC-Aufrufe. Der für diese Arbeit relevante Adapter ist der SAP JCo. Abbildung 3.8 veranschaulicht die SAP JCo-Architektur. Mit dem SAP JCo ist die Einschränk-

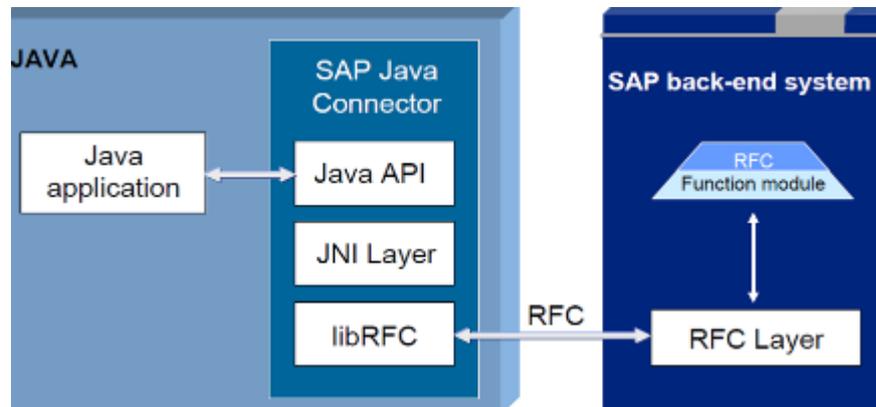


Abbildung 3.8: SAP JCo-Architektur [SB11b]

kung auf die Funktionsbausteine als ABAP-Programme aufgehoben. Der SAP JCo erlaubt einen bidirektionalen Datenaustausch mit SAP-Systemen nicht nur aus den ABAP-Anwendungen, sondern auch aus in Java geschriebenen Anwendungen. Somit interagiert JCo als eine Middleware-Schicht zwischen dem ABAP- und dem Java-Stack. Dieser Aspekt ist insbesondere wichtig, weil die Ankopplung auf der anderen Seite an das *ProM* ebenfalls in der Programmiersprache Java stattfindet. Dabei unterscheidet eine bidirektionale Verbindung zwischen *InboundCall* (Java-Anwendung ruft ABAP-Anwendung), bei der eine direkte oder eine Verbindung über einem *Connection Pool* aufgebaut ist, und dem *OutboundCall* (ABAP-Anwendung ruft Java-Anwendung), bei dem ein Java-Server über einen Socket aufgerufen wird. Ein *Connection Pool* ist in diesem Zusammenhang eine definierte Anzahl an Verbindungen, die einer Anwendung zur Verfügung stehen [M.09]. Benötigt eine Anwendung eine Verbindung zu einem SAP-System, ist die Anfrage an den *Pool* zu stellen, welcher diese bereitstellt. Im Gegensatz dazu müssen die direkten Verbindungen bei einer Anfrage jedes Mal neu aufgebaut werden. Bei Verbindungsbeendigung ist die Verbindung an den *Pool* zurückzugeben, so dass sie wieder zur Verfügung steht. Dabei ist ein Overhead verhindert, welcher bei Verbindungsaufbau entsteht. Zudem wird die Anzahl der Verbindungen begrenzt werden, so dass die Ressourcen gesichert werden können.

Eine Java-Anwendung muss, um die Verbindung aufzubauen, das SAP JCo-Paket integriert haben. Nach der Integration stehen die Methoden des *Java Application Programming Interface* (Java API) bereit. Der Aufruf dieser Methoden wird über die *Middleware Interface*, eine Abstraktionsschicht zwischen der API und der eigentlichen RFC-Bibliothek, in einen ABAP-Aufruf transformiert und an das SAP-System weitergeleitet. Umgekehrt wird ein ABAP-Aufruf über das *Middleware Interface* und die JCo-Java-API in eine Java-Methode transformiert [M.09, SB11b].

Grundsätzlich müssen die Metadaten, wie Import- und Exportparameter sowie die Typisierung dieser, den JCo-Methoden vorliegen. Die Metadaten aller Funktionsbausteine, die von der Java-Anwendung verwendet werden, sind von dem JCo aus

dem SAP-Server übertragbar. Dabei ist insbesondere die Abbildung zwischen den ABAP- und Java-Datentypen wegen ihrer Unterschiede zu beachten.

3.3.3 Datenextraktion

Die Extraktion der Daten findet statt, nachdem eine Verbindung mit dem SAP-JCo-Adapter zu einem SAP-System aufgebaut ist. Zwei Vorgehensweisen sind in der Anwendung, die in dieser Arbeit entwickelt wurde, realisiert und werden im Folgenden erläutert. Das erste Verfahren basiert auf der Benutzung von BAPIs, welche geeignet kombiniert sind, so dass ein Belegfluss eines Prozesses abgebildet ist. In dem zweiten Verfahren kommen die Funktionsbausteine zum Einsatz, die den Zugriff auf das Datenbanksystem ermöglichen, wobei dadurch die in der Vorbereitungsphase identifizierten Informationen aus den Tabellen extrahiert sind. In beiden Methoden ist der in Abbildung 3.9 dargestellte Extract-Transform-Load-Prozess (ETL-Prozess) umgesetzt. Dieser Prozess besteht aus drei Schritten. In dem ersten Schritt, in der



Abbildung 3.9: ETL-Prozess

Extraktionsphase, werden Daten aus einer SAP-Datenbank in ein Repository geladen. Dabei wird eine asynchrone Extraktion verwendet, die anfragegesteuert ist. Der Prozess der Extraktion ist so konzipiert, dass ein ganzer Datenbank-Snapshot verwendet wird. Dies führt zu Effizienzvorteilen gegenüber einem teilweisen *Snapshot*. Bei dem teilweisen *Snapshot* muss zuvor geklärt werden, welche Daten neu eingefügt sind oder eventuell geändert wurden. Demgegenüber gestaltet sich die Extraktion mit einem ganzen *Snapshot* der Daten aus einer begrenzter Menge der Tabellen, wie dies im Fall eines gezielt für die Analyse ausgesuchten Prozesses ist, effizienter. Die BAPIs oder die Funktionsbausteine werden in dieser Phase in der entwickelten Anwendung eingesetzt. Auf eine Anfrage extrahieren die beiden Methoden die zuvor von dem Benutzer spezifizierten Informationen.

Im nachfolgenden Schritt werden die Daten, die sich nach der Extraktion in einem Repository befinden, transformiert. Die Transformation unterteilt sich in zwei Bereiche. Bei der syntaktischen Transformation werden die vorhandenen Daten verbessert und korrigiert. Die Verbesserung der Daten spiegelt sich in der Bezeichnung eines Ereignisses wieder. Diese ist durch eine Zeichenkette ergänzt, weil sie durch die Primärschlüssel repräsentiert ist und für die Interpretation der Modelle durch einen Analysten eine Zeichenkette geeigneter ist als eine Folge von Zahlen. Weiterhin ist der Zeitstempel für die Logdatei aufbereitet, weil in einem SAP-System ein Datum und die zugehörige Zeit vorhanden sind, aber in einer Logdatei im XES-Format ein Zeitstempel, der aus dem Datum und der Zeit zusammengesetzt ist, benötigt wird [AWM11, Aal10]. Des Weiteren ist der Typ eines Ereignisses ergänzt, weil dieser für

die Einträge in einer SAP-Datenbank nicht gespeichert wird. Der Typ für alle Ereignisse ist *Complete*, weil in SAP ein Ereignis in der Datenbank nach vollständiger Beendigung aufgenommen wird und z.B. für einen Typ wie Start die Startzeit nicht protokolliert wird. Außerdem ist bei fehlender Bezeichnung der ausführenden Ressource diese durch eine zufällige Zeichenkette ergänzt. Dies hat z.B. für die Erfassung eines Ressourcenmodells mit *Process Mining* zur Folge, dass diese ihre Aussagekraft verliert. In dem Kontext dieser Arbeit ist die Erfassung der Geschäftsprozessmodelle relevant, wobei die Prozessmodelle gegenüber den hier aufgeführten synthetischen Ergänzungen robust sind. Auch die Datentypanpassung findet in diesem Bereich statt.

Die semantische Transformation gehört dem zweiten Bereich an. Hier findet die Entfernung der Duplikate statt. Eine Bestellanforderung, für die tatsächlich eine Bestellung stattfindet, wird doppelt in das Repositorium aufgenommen, weil sie doppelt in dem SAP-System protokolliert ist. Dieser Umstand wird deshalb in der Phase der Aufbereitung korrigiert.

In dem sich der Transformation anschließenden Schritt wird das Laden der Daten ausgeführt. In dieser Phase werden die gewonnenen und bereinigten Informationen in der Benutzeroberfläche dargestellt und unter Berücksichtigung der Bestimmungen des XES-Formats in eine XES-Logdatei geladen.

Der ETL-Prozess ist in der dreischichtigen Prototyparchitektur integriert, die in Abbildung 3.10 dargestellt wird. Das Repositorium, in welchem sich die Daten nach

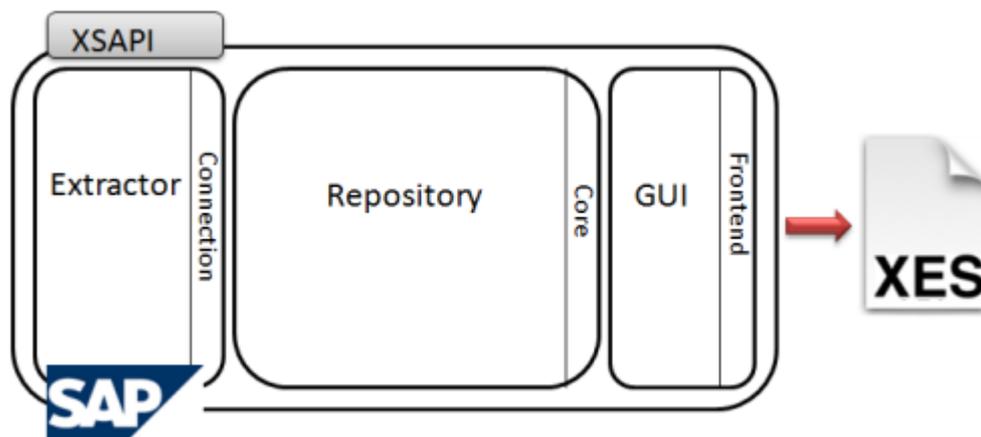


Abbildung 3.10: Dreischichtige Prototyparchitektur

der Extraktion befinden, wird über die Extraktoren mit einem SAP-System verbunden. Dabei ist das Repositorium zur Laufzeit der Anwendung für die Aufbewahrung und die Aufbereitung der Daten verantwortlich. Weiterhin besteht die Möglichkeit, einige Filterparameter in der Benutzeroberfläche zu setzen, welche dem Extraktor das gezielte Extrahieren von Daten erlauben. Die Filterparameter setzen sowohl die Importparameter eines BAPIs als auch eines Funktionsbausteins. Außerdem visualisiert die Benutzeroberfläche die in dem Repositorium enthaltenen Daten. Schließlich werden die extrahierten Daten in einer XES-Logdatei gespeichert und stehen zu weiteren Analysen mit *Process Mining* bereit.

Das Aufbrechen der Architektur auf die drei Bestandteile, *Connection*, *Core*- und *Frontend* erlaubt das flexible Austauschen dieser Teile. Die Extraktoren sind für unterschiedliche Geschäftsprozesse unterschiedlich konfiguriert, weil sie auf verschiedene Tabellen einer Datenbank zugreifen. Für einen Beschaffungsprozess, der in dem Prototyp integriert ist, ist ein anderer Extraktor zu benutzen als für einen Verkaufsprozess. Durch die Flexibilität der Architektur beeinflusst das Austauschen der Extraktoren nicht die übrigen Bestandteile. Ein weiteres Beispiel ist die Benutzeroberfläche, die ausgetauscht werden kann, ohne das Repositorium oder den Extraktor zu beeinflussen.

Vor allem die Extraktoren sind vor der Benutzung des Prototyps einmalig für jedes System zu konfigurieren. Im weiteren Verlauf werden die Architektur des BAPI- und Funktionsbaustein-Extraktors und die Vorbereitungsphase für die beiden Extraktoren für den Beschaffungsprozess aus SAP ERP erläutert.

Das primäre Ziel eines Extraktors ist das Extrahieren der Daten aus dem Quellsystem. Mittels der BAPIs werden die Informationen bereitgestellt, wobei die genaue Vorgehensweise zur Extraktion der Daten seitens eines SAP-Systems dem Entwickler verborgen bleibt, da die BAPIs unter anderem für die Kapselung verantwortlich sind [SB11b, SBa]. Für die Nachbildung eines Belegflusses des Beschaffungsprozesses und damit der Extraktion genau der Daten, die einen Beschaffungsprozess abbilden, wurden die Schritte aus dem Abschnitt 2.3 durchgeführt. In dem *Business Object Browser* (SWO2) unter der Materialwirtschaft sind die Objekttypen, die für den Einkauf in dem Beschaffungsprozess verantwortlich sind, angezeigt. Abbildung 3.11 veranschaulicht die Vorgehensweise. Die Bestellanforderung ist ein Business-

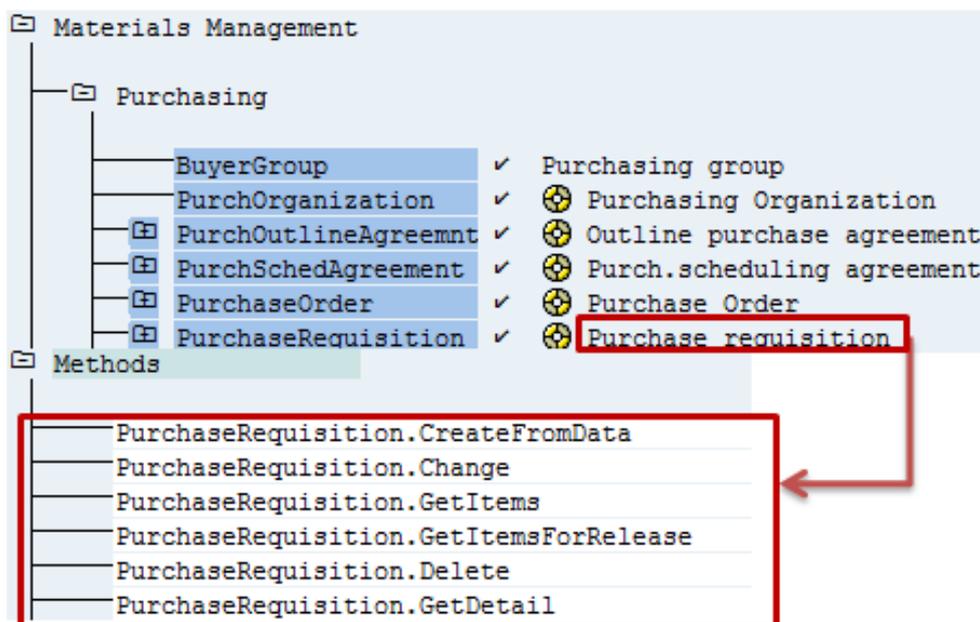


Abbildung 3.11: Vorgehensweise: Identifikation von BAPIs

Objekttyp des Einkaufs, wobei diese in Abbildung expandiert ist. Dazu sind die zur Verfügung stehenden BAPIs aufgeführt, wobei nicht alle für die Abbildung eines

Belegflusses benötigt werden. Methoden, die z.B. für das Löschen (`PurchaseRequisition.Delete`) oder das Ändern (`PurchaseRequisition.Change`) einer Bestellanforderung verantwortlich sind, werden nicht benötigt, da sie den Belegfluss nachträglich verändern und nicht die Ereignisse protokollieren. Dagegen spiegeln die Methoden, die für das Wiedergeben (`PurchaseRequisition.GetItems`) der Einträge und der zugehörigen Details (`PurchaseRequisition.GetDetail`) zuständig sind, den Belegfluss wieder. Weiterhin liefert die Recherche in dem *Business Object Browser* unter der Komponente *Finanzwesen* den Objekttyp, der für das Begleichen der Rechnung zuständig ist. Letztendlich werden für jeden in Abbildung 2.12 dargestellten Prozessschritt zuerst die Komponente, die an der Prozessausführung beteiligt sind, identifiziert und daraufhin die zugehörigen Objekttypen mit den zur Verfügung stehenden BAPIs. Die Menge der identifizierten, für die Modellierung des Beschaffungsprozesses benötigten BAPIs, ist in Abbildung 3.12 dargestellt. Der Prozess der Beschaffung

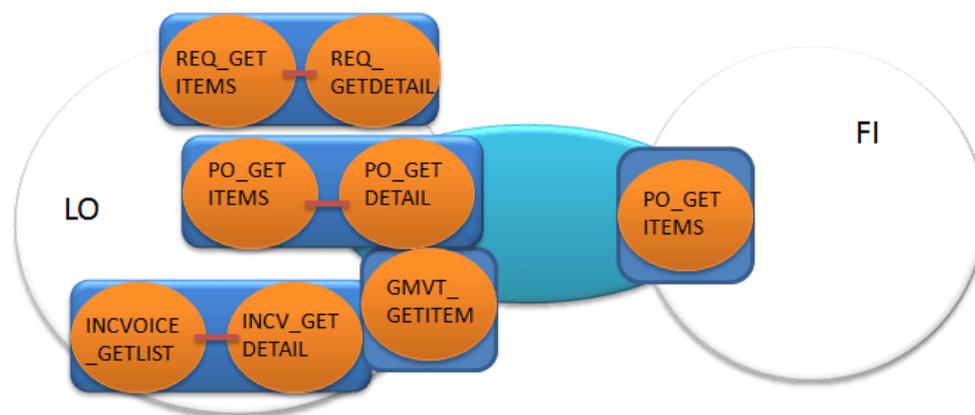


Abbildung 3.12: Extraktion mittels BAPIs

findet komponentenübergreifend statt. Dabei sind die Komponenten die Logistik und das Finanzwesen involviert. Die beiden großen Kreise stellen die Einheiten dar, über die sich der Prozess erstreckt. Des Weiteren sind in Abbildung die Business-Objekttypen dargestellt, die das Auslesen der bereits von den Benutzern erzeugten Spuren und der zugehörigen Ereignisse des Beschaffungsprozesses ermöglichen. Dabei sind die Objekttypen durch die rechteckigen Flächen veranschaulicht. Für den Beschaffungsprozess sind folgende Objekttypen identifiziert: *<Bestellanforderung>*, *<Bestellung>*, *<Warenbewegung>*, *<Eingangsbuchung>* und *<Kreditorenkonto>*. Jeder Objekttyp stellt verschiedene BAPIs zu Verfügung. Dabei sind die benötigten BAPIs durch die Kreise in rechteckigen Flächen dargestellt. Die Bestellanforderung stellen das *BAPI_REQ_GETITEMS* und das *BAPI_REQ_GETDETAIL* zur Verfügung. Während die erste Methode alle erzeugten Bestellanforderungen wiedergibt, gibt die zweite Methode die Details eines konkreten Eintrages wieder. Weiterhin sind für die Darstellung der Daten aus dem Business-Objekttyp Bestellung das *BAPI_PO_GETITEMS* und das *BAPI_PO_GETDETAIL* notwendig. Das erste BAPI stellt alle Bestellungen dar, die von den Benutzern des Systems ausgeführt sind. Unterdessen gibt das zweite bestimmte Details zu einem beliebigen Eintrag wieder. Generell werden Informationen, wie das Datum der Erzeugung oder der Änderung

eines Eintrages, mit dem *GETDETAIL-BAPI* reflektiert. Nachdem eine Bestellung getätigt ist, kann die Ware eingehen. Die Informationen zu dem Objekttyp Warenbewegung liefert *BAPI_GOODSMVT_GETITEMS*. Parallel zu dem Warenerhalt kann die Rechnung eingehen. Die Informationen zu Eingangsrechnung sind von dem *GETITEMS*- und dem *GETDETAIL-BAPI* wiedergegeben. Falls die Rechnung eingegangen ist, kann sie abschließend beglichen werden. Dieser Prozessschritt findet in dem Finanzwesen statt und wird von dem Objekttyp Kreditorenkonto verwaltet. Dazu steht das *GETITEMS-BAPI* zur Verfügung.

Die Verbindungen, die durch die roten Linien in Abbildung 3.12 visualisiert wurden, wurden manuell hinzugefügt. Dabei stellt die Verknüpfung der BOs einen Belegfluss dar. Das folgende Beispiel 3.13 illustriert den Belegfluss anhand der untereinander verbundenen Daten aus den BOs. Die beiden BAPIs aus dem BO Bestellanforderung

Document Number	Resource Name	Date	Change Date	Related fields
10015508	BLYUFSHTEYN	13.07.2011	-	4500018315
10015510	BLYUFSHTEYN	13.07.2011	-	4500018317
4500018315	BLYUFSHTEYN	21.07.2011	-	10015508
4500018317	BLYUFSHTEYN	14.07.2011	18.07.2011	10015510
5000013294	BLYUFSHTEYN	26.08.2011		4500018317

Abbildung 3.13: Herstellung eines Belegflusses mit den BAPIs

verbindet die Bestellanforderungsnummer. Diese geht aus den Informationen hervor, die mit dem ersten BAPI der Bestellanforderung zu bekommen sind. In Abschnitt 2.2.2 wird erläutert, wie ein Ereignis der Bestellanforderung in einer SAP-Datenbank eindeutig identifiziert wird. Dabei identifiziert eine Bestellanforderungsnummer ein Ereignis der Bestellanforderung auf die gleiche Weise. Zu jeder Bestellanforderungsnummer sind detaillierte Informationen mit dem zweiten BAPI des BOs extrahiert. Falls zu einer Bestellanforderung tatsächlich eine Bestellung von einem Benutzer durchgeführt wird, ist die Bestellnummer mit dem zweiten BAPI ebenfalls zu erhalten, wobei diese Information in Abbildung 3.13 durch die erste Zeile gezeigt ist und befindet sich in dem Bezugfeld (*Related fields*) in Abbildung. Dadurch gelingt die Verbindung zwischen dem BO Bestellanforderung und der Bestellung. Diese Nummer wird in der Bestellung mit dem ersten BAPI des BOs extrahiert. Weiterhin sind mit dem zweiten BAPI der Bestellung für jede Bestellnummer detaillierte Informationen bereitgestellt. Infolge einer Bestellung kann die bestellte Ware sowie die zugehörige Rechnung von dem System aufgenommen werden. Das BO Warenbewegung liefert deshalb insbesondere zu jedem Eintrag die Bestellnummer. Über diese findet die Verbindung des BOs Bestellung mit dem BO Warenbewegung statt, wobei die letzte Zeile in Abbildung 3.13 zeigt, wie die bestellte Ware mit der Bestellnummer 4500018317, mit der erhaltenen Ware mit der Nummer 5000013294 durch die Bestellnummer im Feld *Related fields* in Verbindung gebracht wird. Das BO Eingangsrechnung liefert ebenfalls zu jeder Rechnung die zugehörige Bestellnummer, so dass geprüft werden kann, inwieweit für eine Bestellung die Rechnung bereits

eingegangen ist. Der Prozessschritt der Begleichung, einer eingegangenen Rechnung, findet in dem BO Kreditorenkonto statt. Dazu prüft das BO für jede Rechnung, inwieweit der Eintrag für diese ausbalanciert ist. Zu diesem Zwecke sind die detaillierten Informationen aus dem BO Eingangsrechnung als Importparameter für das BAPI des BOs Kreditorenkonto eingesetzt, wodurch eine eingegangene Rechnung identifiziert ist. Letztendlich wird durch das Feld *Related fields* der Workflow eines Prozesses rekonstruiert.

Das SAP-System stellt Funktionsbausteine zu Verfügung, die das Extrahieren der Daten aus der vorliegenden Datenbank erlauben. Die Recherche ergab folgende, für diese Arbeit relevante Bausteine:

1. *RFC_GET_TABLE_ENTRIES*
2. *RFC_READ_TABLE*

Der erste Baustein liefert die Datensätze einer beliebigen Tabelle aus einer SAP-Datenbank, wobei 512 Zeichen pro Datensatz übertragen werden und ein Datensatz in Form einer zusammenhängenden Zeichenkette dargestellt wird. Aus diesem Grund muss eine Zeichenkette vor der Verarbeitung zuerst in die einzelnen Datenfelder zerlegt werden. Der zweite Baustein liefert ebenfalls die ersten 512 Zeichen eines Datensatzes, wobei der Vorteil des zweiten Bausteines gegenüber dem ersten der Parameter *OPTIONS* ist. Hier können die SQL-Anfragen formuliert werden, wie bei SQL-Anfragen in einer gewöhnlichen Datenbank. Da die Belegnummer, die Ressource und der Zeitstempel, die für die Herstellung eines Prozessmodells benötigt werden, in SAP eine feste Länge haben, die nicht 512 Zeichen überschreitet, entspricht dies den Anforderungen. Weiterhin kann diesem Baustein der Parameter *FIELDNAME* übergeben werden, so dass die Datenfelder einer Tabelle, die benötigt werden, spezifiziert werden können und dadurch können diese einzeln extrahiert werden und sind nicht in Form einer zusammenhängenden Zeichenkette dargestellt. Deshalb wird in dieser Arbeit der zweite Funktionsbaustein benutzt, um die Daten zu extrahieren.

Zur Konfiguration des Extraktors sind die Schritte aus dem Abschnitt 2.3 zum Identifizieren der Tabellen durchzuführen, die dem Funktionsbaustein mit dem Parameter *QUERY_TABLE* übergeben werden. Die benötigten Tabellen sind mit dem Werkzeug *Object Navigator* in SAP auffindbar, wie in dem Abschnitt 2.3.3 gezeigt wurde. Tabelle 3.1 zeigt ein Ausschnitt der Ereignisse aus dem Beschaffungsprozess und die zugehörigen Tabellen auf. Dabei entspricht z.B. das erste Ereignis *Create Purchase Requisition* in dem Beispiel einer Bestellanforderung, welche die erste Aktivität in einem Beschaffungsprozess sein kann und in der EBAN-Tabelle protokolliert wird.

Ferner stellt das SAP-System das Werkzeug *Function Builder* (Transaktion: SE37) zur Verfügung, mit welchem die Funktionsbausteine getestet werden können. Hier sind die Eigenschaften der Bausteine einsehbar, wobei dadurch die Import- und die Ausgabeparameter der Funktionenbausteine aufgelistet werden.

eine zugehörige Bestellung erfolgt ist, in das Repositorium extrahiert. Das Datenfeld, welches die Ressource (ERNAM) identifiziert, wird ebenso extrahiert sowie der Zeitstempel der erzeugten Bestellanforderung, welcher aus dem *BADAT*- und *UZEIT* gebildet wird.

Weiterhin beinhalten die EKPO- und die EKKO-Tabellen in Abbildung 3.14, die Informationen zu einem Bestellungsprozess. Aus der EKPO-Tabelle ist das EBELN-Feld zum Identifizieren einer Bestellung sowie das EBELP-Feld zum Identifizieren einer Position in der Bestellung extrahiert. Unter der Anwendung des zweiten Datenfeldes sind die tatsächlichen Positionen einer Bestellung aus der EKKO-Tabelle in das Repositorium eingefügt.

Ferner speichert die EKBE-Tabelle die Bestellentwicklung. Eine Bestellentwicklung existiert ab dem Zeitpunkt der tatsächlichen Bestellung. Zum Identifizieren einer Bestellung, für die eine Historie in der Tabelle angelegt ist, ist das EBELN-Feld extrahiert. Dazu protokolliert das VGABE-Feld die Einkaufshistorie einer Bestellung. In dem SAP-System in Magdeburg sind diesbezüglich die in Abbildung 3.15 dargestellten Ereignisse realisiert. Für diese Arbeit sind insbesondere die Vorgänge 1, 2

Trans./event t...	Short Descript.
1	Goods Receipt
2	Invoice Receipt
3	Subseq. Debit/Credit
4	Down Payment
5	Payment
6	Goods Issue for Stock Transfer
7	Consumption (Subcontracting)
8	Delivery (Stock Transfer)
9	Service Entry Sheet
A	Down Payment Request
C	Down Payment Clearing
Q	Comparison for Internally Posted Material (Only IS-OIL)
R	Return Delivery via Delivery Note
P	Invoice Parking
V	Down Payment Request Clearing

Abbildung 3.15: Einkaufshistorie in SAP

und 5 relevant, weil diese die Ereignisse aus dem Beschaffungsprozess darstellen. Der Vorgang mit der Nummer 1 entspricht dem Erhalt von Ware. Falls die Rechnung eingeht, resultiert das Ereignis in dem Vorgang mit der Nummer 2. Das Begleichen eingegangener Rechnung entspricht dem letzten Vorgang. Zu allen diesen Ereignissen werden der Name der Ressource sowie der Zeitstempel extrahiert.

Die Änderungen eines Prozesses werden in den CDHDR- und CDPOS-Tabellen protokolliert. Dabei speichert die CDHDR-Tabelle die allgemeinen Informationen einer Änderung. Dagegen sichert die CDPOS-Tabelle die detaillierten Änderungen eines Objektes. Der Name der Ressource, die eine Änderungen ausführt, ist in dem USERNAME-Feld abgelegt. Dazu ist der Zeitstempel in den UDATE- und UTIME-

Feldern gesichert. Das OBJECTID-Feld protokolliert die Belegnummer, für welche die Änderung stattfindet. Außerdem sichert das CHANGE_IND-Feld die Art der Änderung. Dazu sind die in Abbildung 3.16 dargestellte Änderungsarten möglich. Dabei sind in dieser Arbeit alle Änderungsarten unter dem Ereignis *Change Purchase*

Appl. object cha...	Short Descript.
U	Update
I	Insert
E	Delete (Single Field Documentation)
D	Delete
J	Insert (Single Field Documentation)

Abbildung 3.16: Änderungsarten eines Objektes

Order zusammengefasst, weil für die Erzeugung eines Prozessmodells nicht relevant ist, inwieweit eine Einfügung einer fehlenden Information oder eine Änderung bereits vorhandenen Inhaltes in einem Objekt stattfindet.

Des Weiteren ist die Verknüpfung der Inhalte aus den Tabellen zur Erzeugung eines Belegflusses relevant. Das Aufbauen der Verbindung zwischen der EKPO- und der EKBE-Tabelle ist in Abschnitt 3.2 erläutert. Zu jeder Position einer Bestellung aus der EKKO-Tabelle wurde auf die gleiche Art die Verbindung zu EKBE-Tabelle hergestellt und damit die Historie dieser Positionen extrahiert. Weiterhin wurde infolge einer Bestellung im Zusammenhang mit einer Bestellanforderung die Verbindung zwischen der EBAN- und der EKBE-Tabelle hergestellt. Hier interagiert das EBELN-Feld als Verbindungsschlüssel zwischen den Tabellen. Falls zu einer Bestellanforderung keine Bestellung erfolgt ist, ist in dem Feld keine Bestellnummer vorhanden und die Position hat deshalb keine Historie.

Aus dem EBELN-Feld in der EBAN-Tabelle resultiert beim Herstellen des Beschaffungsprozesses das Problem der Duplikate. Eine Bestellung ist in der EBAN- und EKPO-Tabelle gleichzeitig protokolliert. Die Lösung besteht in der Entfernung der Bestellungen aus den extrahierten Daten aus der EKPO-Tabelle, für die eine Bestellanforderung vorhanden ist.

Letztendlich sind die Bestellanforderungs- und die Bestelldaten mit den Daten aus der CDHDR-Tabelle verbunden. Dazu steht das OBJECTID-Feld zur Verfügung. Dadurch ist für jede Bestellanforderung sowie jede Bestellung ermittelt worden, inwieweit eine Änderung für die Belege aus den Prozessschritten erfolgt ist.

3.4 Anbindung an ProM

Nach der Erläuterung der Anbindung an das SAP-System, wird im Folgenden die Anbindung der Anwendung an das *ProM* dargestellt. Dazu wird eine Logdatei in XES-Format erzeugt, welche von dem Werkzeug analysiert wird. Zu diesem Zweck wurde die Klasse *XESMapper* entwickelt, wobei dazu die OpenXES-Bibliothek verwendet wurde. Diese ist eine Referenzimplementierung des XES-Standards, die unter

der GNU Lesser General Public License (LGPL) steht [C.09].

Die Bibliothek stellt Methoden für das Lesen und Schreiben einer Logdatei im XES-Format bereit. Diese Methoden folgen dabei dem in Abbildung 3.17 dargestellten UML-Klassendiagramm. Der linke Teil des Diagramms ist der Kern der Bibliothek,

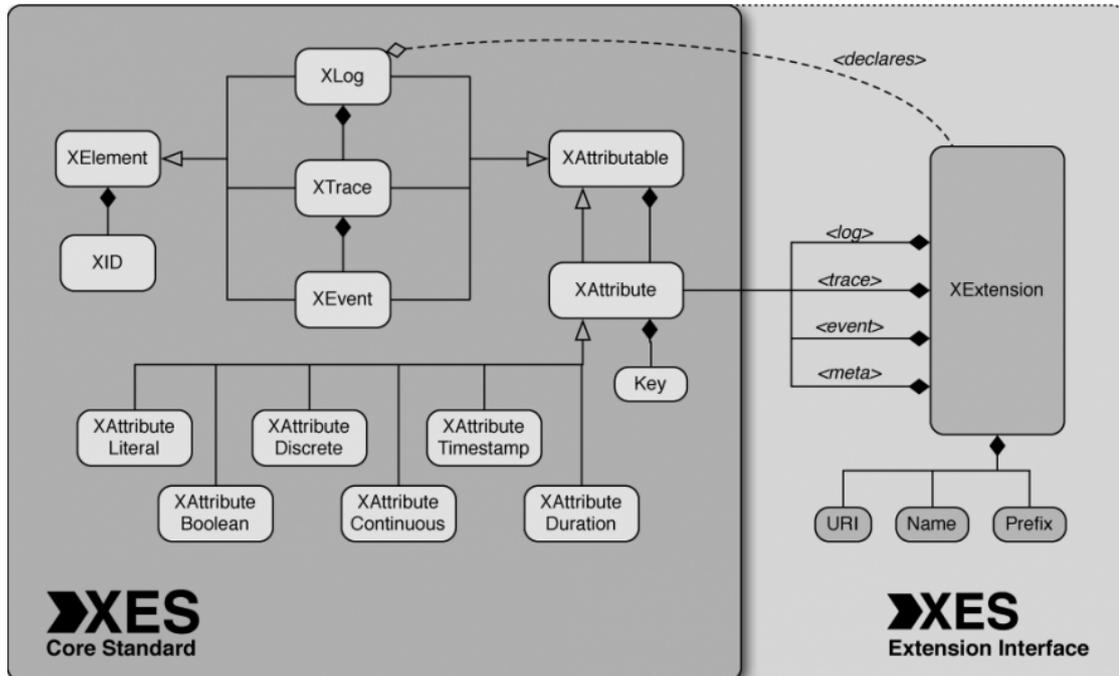


Abbildung 3.17: Struktur eines Ereignislog in UML-Notation [C.09]

welcher in dieser Arbeit, für die Erzeugung einer XES-Logdatei, implementiert wurde. Der rechte Teil stellt die Erweiterungen dar, welche in Kapitel 2.2.3 beschrieben wurden.

Im Vorfeld der Einfügung der Daten aus dem Repositorium in die Logdatei wird der Kopf der Datei generiert. Abbildung 3.18 zeigt einen Ausschnitt der aus der Applikation resultierenden XES-Logdatei, die dem Aufbau aus dem UML-Klassendiagramm in Abbildung 3.17 folgt. Dabei beginnt eine Logdatei im XES-Format mit einem XLog-Element [C.09], wobei dieses einen Container für die Spuren darstellt. Dem Element ist das Standard-Ereignismodell zugeordnet, welches in Abschnitt 2.2.2 erläutert wurde. Anschließend werden mit dem XTrace-Element die Spuren mit der Standard-Erweiterung *XConceptExtension* der Logdatei hinzugefügt, die eine fortlaufende Nummerierung sowie einen Namen enthält. Weiterhin enthält jede Spur die zugehörigen Ereignisse. Dabei sind diese mit dem XEvent-Element erzeugt. Jedem Ereignis liegen vier Erweiterungen bei. Die erste ist die *XConceptExtension*, wobei diese für die Namensgebung eines Ereignisses verantwortlich ist. Des Weiteren stellt die zweite Erweiterung, die *XOrganizationalExtension*, die Ressource dar. Ferner ist der Zeitstempel eines Ereignisses durch die dritte Erweiterung *XTimeExtension* abgebildet. Die vierte Erweiterung *XLifecycleExtension* repräsentiert den Ereignistyp, welcher den Wert *Complete* beinhaltet.

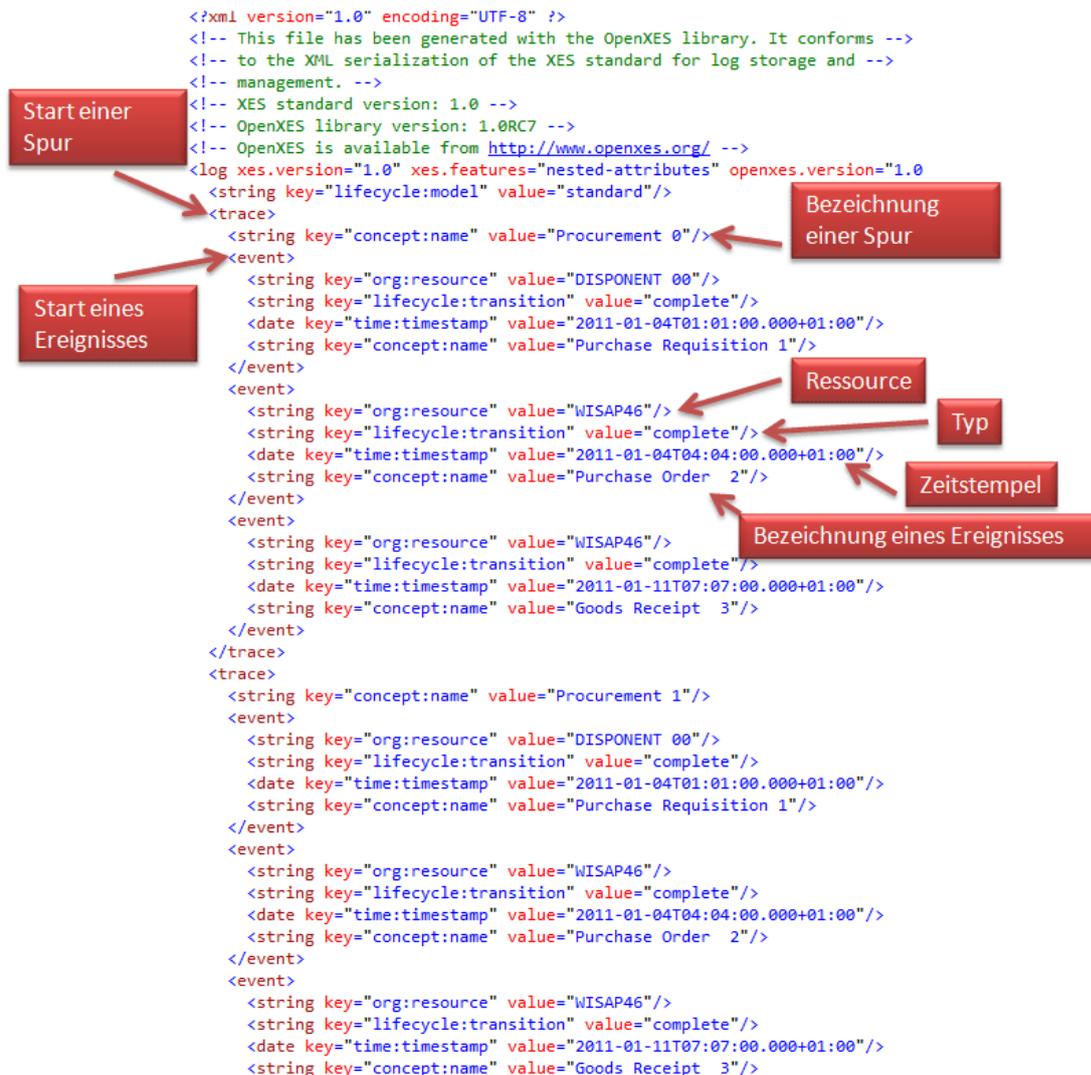


Abbildung 3.18: Ein Ausschnitt der generierten XES-Logdatei

3.5 Benutzeroberfläche

In diesem Kapitel wird die Benutzeroberfläche der in dieser Arbeit implementierten, XSAPI-Anwendung sowie eine Bedienungsanleitung zum Extrahieren einer Ereignislogdatei vorgestellt. Dieser Workflow umfasst den Verbindungsaufbau mit einem SAP-System über die Auswahl eines zu untersuchenden Prozesses bis hin zur Erzeugung einer Logdatei. Abbildung 3.19 zeigt die Benutzeroberfläche der Anwendung. Dabei unterteilt sich die Oberfläche in drei Bereiche. In dem oberen Teil der Anwen-

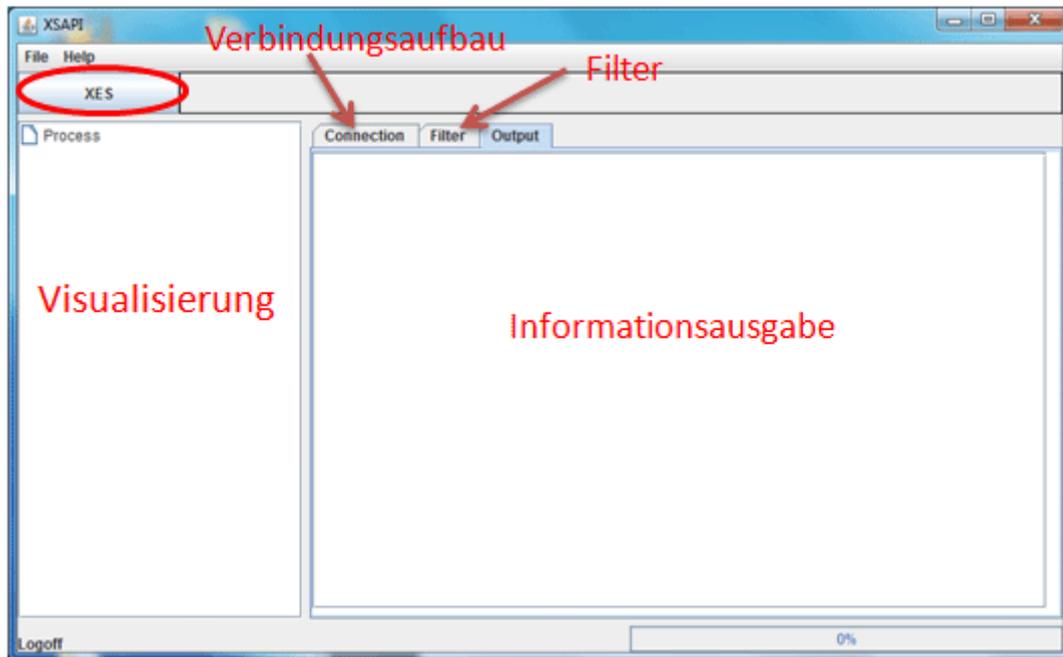


Abbildung 3.19: Benutzeroberfläche der Anwendung

derung befinden sich der XES-Knopf und der Bereich für die aktuelle Informationsausgabe. Der linke Teil der Oberfläche visualisiert die Daten nach der Extraktion aus einem SAP-System. Des Weiteren ist der rechte Bereich für den Verbindungsaufbau, die Filtersetzung und die Ausgabe der zusätzlichen Informationen reserviert. Letztendlich ist in der unteren Leiste der Prozessfortschritt dargestellt. In den nächsten Abschnitten werden die Bereiche im Detail vorgestellt.

3.5.1 Allgemeine Einstellungen

Zum Extrahieren der Daten ist die Verbindung zu einem SAP-System aufzubauen. Der SAP Java Connector, welcher in Abschnitt 3.3.2 erläutert wurde, stellt hierzu Methoden zur Verfügung. Die Parameter dieser Methoden sind von dem Benutzer zu setzen. Abbildung 3.20 veranschaulicht die Verbindungskonsolle in dem Connection-Tab. In der Konsole sind fünf Felder zu setzen. Zuerst ist der Servername zu bestimmen, zu welchem die Verbindung aufgebaut wird. In dieser Arbeit wird der Server der Universität Magdeburg verwendet. Weiterhin ist der Client spezifiziert, welcher den Mandanten darstellt. Das Prinzip des Mandanten-Systems wurde in Abschnitt

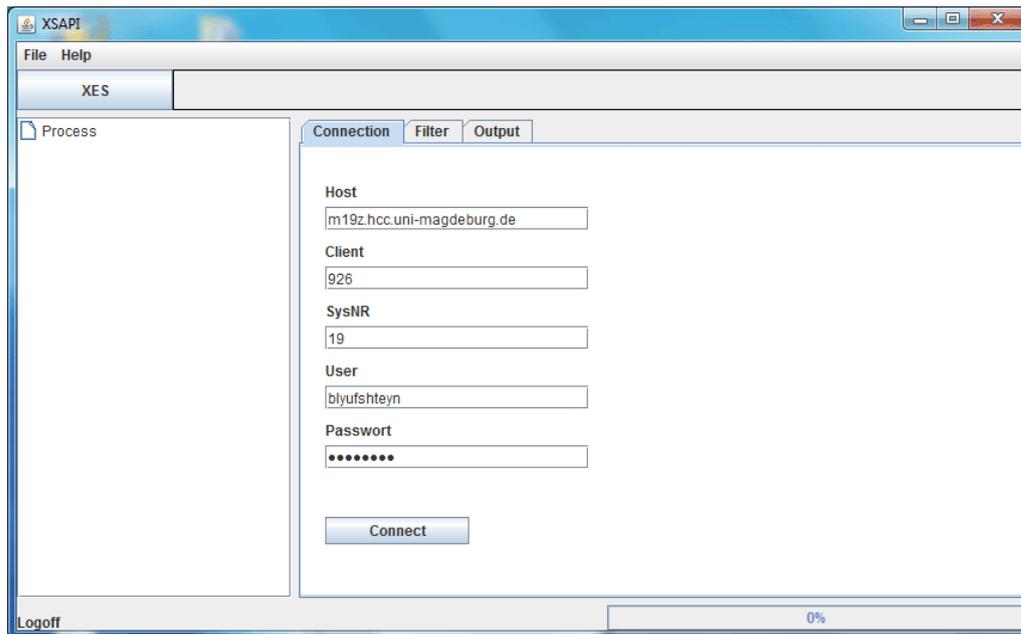


Abbildung 3.20: Verbindungsaufbau mit einem SAP-System

2.1.1 erläutert. Dabei interagiert in dieser Arbeit der Mandant 926. Des Weiteren ist die Systemnummer zu spezifizieren, welche in dieser Arbeit 19 ist. Schließlich sind die Benutzerangaben zu füllen, die aus dem Benutzernamen und dem Passwort bestehen.

Nach dem Setzen der Parameter und Betätigen des Connection-Knopfes, wird eine direkte InboundCall-Verbindung zu einem SAP-System aufgebaut. Dabei wechselt die Anwendung aus dem Zustand *Logoff*, welcher im unterem, linken Bereich angezeigt ist, in den Zustand *Login*. Der erfolgreich ausgeführte Zustandswechsel stellt eine erfolgreich aufgebaut Verbindung dar.

3.5.2 Filter

Nachdem die Verbindung erfolgreich aufgebaut wurde, wird der Filter-Tab aktiviert. Der Tab ist in Abbildung 3.21 dargestellt. Die Auswahl des zu analysierenden Prozesses sowie die Setzung der Filter ist dadurch ermöglicht. Zu Beginn ist der zu analysierende Prozess auszuwählen. In dieser Arbeit ist die Erzeugung einer Ereignislogdatei realisiert, die das Analysieren eines Beschaffungsprozesses aus SAP ERP ermöglicht.

Weiterhin ist ein Datumsfilter realisiert. Dieser erlaubt dem Benutzer, ein Datum zu setzen, so dass die Daten in einer Logdatei inkludiert sind, die mit dem Datumintervall übereinstimmen. Auf diese Weise ist z.B. die Analyse eines Prozesses eines Monats realisierbar und somit die Effizienzsteigerung während des Analysevorgangs erreichbar, da nicht der gesamte Inhalt aus den Tabellen zu extrahieren ist.

Die Analyse beginnt mit der Auswahl eines Mandanten, wobei dieser eine abgeschlossene Einheit mit einer Datenbank innerhalb eines SAP-Systems darstellt. Des Weiteren wird das Werk festgelegt, in welchem der Beschaffungsprozess durch die

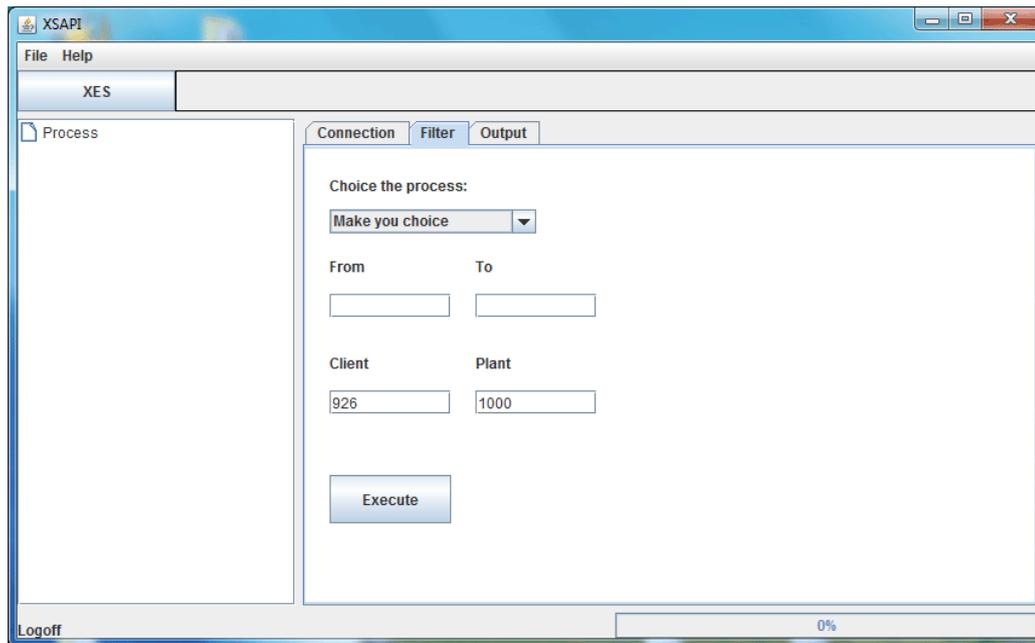


Abbildung 3.21: Filter-Tab

Benutzer des Systems ausgeführt wurde. Dadurch ist die Analyse für einen speziellen Buchungskreis und insbesondere für ein bestimmtes Werk ausführbar. Schließlich fängt die Datenextraktion aus der festgelegten Datenbank nach dem Betätigen des Execute-Knopfes an.

3.5.3 Visualisierung

Im Anschluss an die Datenextraktion werden die Daten in der Anwendung visualisiert. Abbildung 3.22 stellt die Visualisierung der Informationen dar. Der Bereich für die zusätzliche Informationsausgabe gibt die erfolgreiche Ausführung der Datenextraktion bekannt. Weiterhin sind in der Output-Konsole die Ausführungszeit sowie die Ereignisse des Prozesses mit der Anzahl ihres Auftretens dargestellt.

Im linken Bereich der Anwendung befindet sich die Visualisierung der extrahierten Daten. Der Beschaffungsprozess ist mit den zugehörigen Spuren in einem Baum dargestellt. Eine Spur beinhaltet, abhängig von der Ausführung des Prozesses, eine Menge an Ereignissen. Die Spur mit der Nummer 386 in Abbildung umfasst z.B. fünf Ereignisse: die Bestellanforderung, die Bestellung, den Wareneingang, den Rechnungseingang und die Bezahlung. Die Visualisierung der Daten entspricht der Logdatei, die im folgenden Schritt zu erzeugen ist. Dazu ist zuerst der Speicherort auszuwählen und abschließend der XES-Knopf zu betätigen. Die Darstellung der Daten in der Anwendung gibt dem Anwender die Möglichkeit, die Anzahl der Spuren und die ausgeführten Ereignisse vor der Erzeugung der Logdatei zu kontrollieren. Falls die Menge der Spuren nicht expressiv ist, ist das Datumsintervall zu vergrößern und erneut die Extraktion der Daten vor dem Betätigen des XES-Knopfes auszuführen.

The screenshot shows the XSAPI application window. The title bar reads 'XSAPI'. Below the title bar is a menu bar with 'File' and 'Help'. A status bar at the top indicates 'Process is done.' and 'XES'. The main area is divided into two panes. The left pane shows a tree view of traces, including Trace-384 through Trace-389, each with sub-items like REQ, PO, GR, IR, and PAY. The right pane has tabs for 'Connection', 'Filter', and 'Output'. The 'Output' tab is active, showing a table of data with columns: REQ--, CHG--, PO--, CHO--, GR--, IR--, and PAY--. Below the table, it says 'Execution time: 53 sec'. At the bottom of the window, there is a 'Logon' button and a progress indicator showing '100%'.

REQ--	CHG--	PO--	CHO--	GR--	IR--	PAY--
1703	18	3584	23	3309	1510	41

Abbildung 3.22: Visualisierung der Daten

3.6 Fazit

In Kapitel 3 wurde zuerst das Ziel dieser Arbeit und anschließend die Anwendung vorgestellt, welche in dieser Arbeit entwickelte wurde. Dazu wurden die Schritte zur Konfiguration der Applikation sowie die Anbindung, sowohl an das SAP-System als auch an das *ProM* erläutert. Die Konfiguration beinhaltet die Vorbereitungsschritte, welche durchgeführt werden müssen, um die Daten für den zu analysierenden Prozess in SAP ERP zu identifizieren und in Relation zu setzen. Dazu sind Kenntnisse in SAP notwendig, da SAP ein Informationssystem ist, welches keine konkrete Ereignislogdatei zur Verfügung stellt, sondern lediglich alle Information, die für eine Logdatei benötigt sind, wobei diese Informationen in dem System in einem Process-Mining-Projekt zuerst identifiziert werden müssen, weil sie notwendig sind, um Process-Mining-Analysen zu betreiben [AWM11].

Zwei Technologien sind zur Anbindung an ein SAP-System in dieser Arbeit implementiert. Zuerst wurde die Datenextraktion mit den BAPIs durchgeführt. Das Vorgehen zeigte, dass weit tiefere Kenntnisse des SAP-Systems notwendig sind, als die in dem Vorbereitungsschritt geplanten, weil die Relationen zwischen den BAPIs nicht exakt sind und das System keine implizite Fremdschlüsselbeziehung zwischen den BAPIs bereitstellt, so dass diese manuell zu identifizieren sind.

Die RFC-Bausteine stellen die zweite Technologie dar, die in dieser Arbeit zu Datenextraktion verwendet wurde. Mit der Durchführung der Vorbereitungsschritte wurden die Tabellen sowie die Relation zwischen den Datensätzen identifiziert. Die Extraktion erfolgt unter der Verwendung eines RFC-Bausteins, welcher als Parameter die Tabellen und die Relation zwischen diesen übergeben bekommt.

Die Informationen befinden sich nach der Extraktion in einem Repository. Dabei sind sie gleichzeitig in der Benutzeroberfläche visualisiert. Eine detaillierte Erläu-

terung der Bedienung der Benutzeroberfläche, von der Datenextraktion aus einem SAP-System bis hin zur Erzeugung einer Ereignislogdatei, ist erfolgt. Im nächsten Kapitel wird der Test der Anwendung an dem Beschaffungsprozess aus SAP ERP erläutert.

Kapitel 4

Fallstudie

In den vorangegangenen Kapiteln wurden die in dieser Arbeit entwickelten Anwendung vorgestellt. Um die Anwendung zu evaluieren, wird in diesem Kapitel das Vorgehen auf den Beschaffungsprozess aus einem SAP-System angewendet, wobei dazu jeder beliebiger Prozess aus SAP verwendet werden kann, wenn die dazugehörigen Tabellen bekannt sind. Dazu werden in einem SAP-System verschiedene Beispielinstanzen des Prozesses ausgeführt sowie die bereits vorhandenen Instanzen verwendet, um eine Menge an Daten für die Anwendung von Process-Mining-Analyse zu gewährleisten.

In dem ersten Kapitel wird der Beschaffungsprozess vorgestellt, welcher in dem SAP-System des IDES-Modellunternehmens integriert ist. Dazu gehört die Beschreibung der einzelnen Schritte des Prozesses mitsamt der Komponentensichtdiagramme. Das zweite Kapitel stellt die Datenerhebung und den Aufbau der Testreihen dar. Diesbezüglich ist insbesondere eine Durchführung einer Prozessinstanz in dem SAP-ERP-System des IDES-Unternehmens erläutert. Dazu wird ein Szenario beschrieben, in dem fehlendes Material für die Herstellung eines Rennrades beschafft wird. Im darauf folgenden Kapitel werden die Ergebnisse der Extraktion einer XES-Logdatei präsentiert. Diese Ergebnisse werden anschließend interpretiert.

4.1 Beschaffungsprozess

Ein Beschaffungsprozess in SAP ERP hat eine Vielzahl von Verknüpfungsstellen zu anderen Komponenten des Systems. Dieses Kapitel erläutert die einzelnen Schritte des Beschaffungsprozesses und stellt die Integrationspunkte mit den anderen Komponenten dar.

Der in dieser Arbeit ausgeführte Beschaffungsprozess findet im Kontext eines Szenarios statt. Abbildung 4.1 stellt den Prozess dar. Dieser fängt mit dem Ereignis der Bestellanforderung an, in Abbildung *Purchase Requisition* bezeichnet. Daraufhin wird die benötigte Ware in dem Ereignis *Purchase Order* bestellt und in dem Ereignis *Good Receipt* erhalten. Parallel zum Erhalten der Ware kann eine dazugehörige Rechnung eingehen, welche in dem Ereignis *Invoice Receipt* aufgenommen wird. Letztendlich wird der Prozess durch das Bezahlen der Rechnung in dem Ereignis *Payment* beendet.

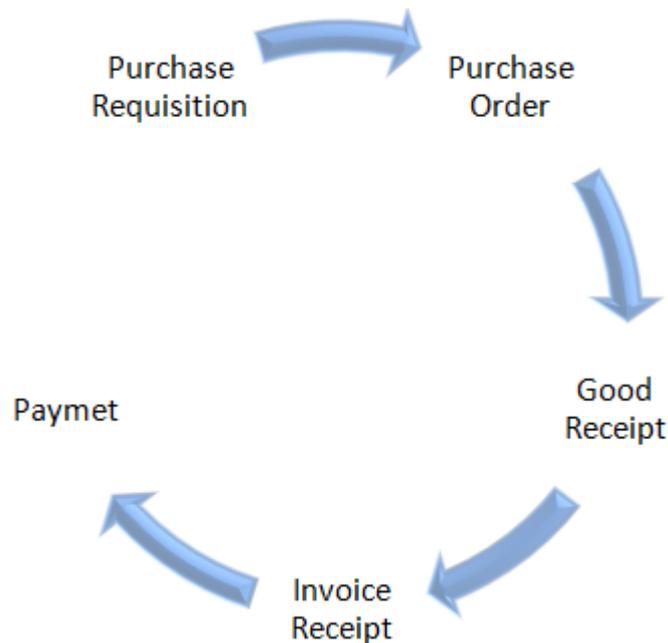


Abbildung 4.1: Beschaffungsprozess in IDES [SBg]

Das dazugehörige Szenario beschreibt die Herstellung eines Rennrades in dem IDES-Unternehmen. Zum Beginn der Herstellung wurden die Materialstammsätze in das System eingepflegt. Dazu gehören vor allem die Aufbaustruktur des Fahrrades sowie die dazugehörige Organisationsstruktur. Dabei beschreibt die Aufbaustruktur die Stücklisten und die Komponenten des Produkts. Die Organisationsstruktur beschreibt die Grunddaten, die mandantenweit gültig sind, die Produktionsplanung, welche beispielhaft im Werk 1000 in Hamburg stattfindet, sowie das Finanzwesen [W.98, M.08]. Dazu werden die organisatorischen Daten für jedes Element der Stückliste eingetragen. Die Schaltung des Rennrades ist insbesondere das Element, welches in dem Szenario in einem Fremdbeschaffungsprozess eingekauft wird.

Im nächsten Schritt wird der Lieferantenstammsatz eingetragen, welcher in dem Prozess die Bezugsquelle für die Schaltung darstellt. Dazu gehören vor allem die Einkaufsinfosätze und die dazugehörigen Konditionen und die Adresse des Kreditors sowie das Konto, von welchem die Rechnung beglichen wird [W.98, M.08]. Die einmalige Einpflege dieser Daten ermöglicht im weiteren Verlauf eine effiziente Beschaffung der fehlenden Materialien, weil sie für alle weiteren Bestellungen wiederverwendet werden.

Nachdem die Stammdaten eingetragen wurden, wird der Beschaffungsprozess für die Schaltung ausgeführt. In dem Szenario fängt der Prozess mit dem Anlegen der **Bestellanforderung** an. Abbildung 4.2 stellt einen möglichen Ablauf der Bestellanforderung dar. In dem Beispiel ist ein Komponentensichtdiagramm dargestellt, welches von SAP zur Verfügung gestellt wird. Es existieren in SAP zwei Möglichkeiten, um eine Bestellanforderung zu generieren, wobei das Ereignis in Abbildung *Create/process purchase requisitions* bezeichnet wird. Die erste Möglichkeit ist die manuelle. Dabei trägt ein zuständiger Mitarbeiter die erforderlichen Daten in das

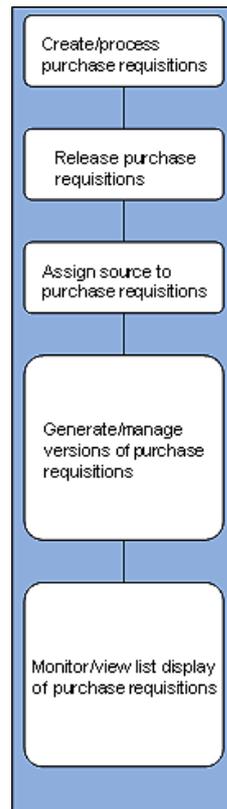


Abbildung 4.2: Geschäftsprozess: Bestellanforderung [SBg]

System ein. Dagegen ist die zweite eine automatische. Diesbezüglich wird die Anforderung über die Materialbedarfsplanung (MRP) generiert. Nachdem das Anlegen einer Anforderung ausgeführt ist, ist im nächsten Schritt die Anforderung freizugeben, bevor sie einer Bezugsquelle zugeordnet werden kann. Dazu wird das Ereignis *Release purchase requisitions* in einem SAP-System ausgeführt. Diesbezüglich muss eine Freigabestrategie definiert werden. Falls eine Freigabestrategie nicht definiert oder aktiviert ist, sind die Bestellanforderungen automatisch freigegeben [W.98, M.08]. Da für das Szenario in IDES eine Freigabestrategie nicht definiert wurde, sind die erzeugten Bestellanforderungen immer freigegeben. Im nachfolgenden Schritt *Assign source to purchase requisitions* kann die Zuordnung zu einer Bezugsquelle stattfinden. Dieser Vorgang ist sowohl manuell von einem Mitarbeiter als auch automatisch durch das System ausführbar. Des Weiteren kann die Zuordnung zu einer Bezugsquelle bereits beim Anlegen einer Bestellanforderung stattfinden, wenn der Lieferantenstammsatz des Kreditors angelegt ist. Letztendlich kann eine Bestellanforderung, in dem Prozessschritt *Monitor/view list display of purchase requisitions*, kontrolliert werden. Weiterhin sind jederzeit Änderungen von Bestellanforderungen von einem zuständigen Mitarbeiter ausführbar. Schließlich entspricht eine Bestellanforderung dem Bedarf, der für die Einkaufsabteilung freigegeben wird, wobei dadurch die Einkaufsabteilung aufgefordert wird, die ermittelte Menge eines Materials oder einer Dienstleistung zu beschaffen [W.98, M.08].

Ein weiterer Vorgang in dem Beschaffungsprozess ist die **Bestellung**. Dabei kann diese entweder in Verbindung oder ohne Verbindung zu einer Bestellanforderung, einer Anfrage oder einer anderen Bestellung erzeugt werden. Abbildung 4.3 stellt einen möglichen Bestellprozess dar. Das Ziel einer Bestellung ist es, einen Bedarf in

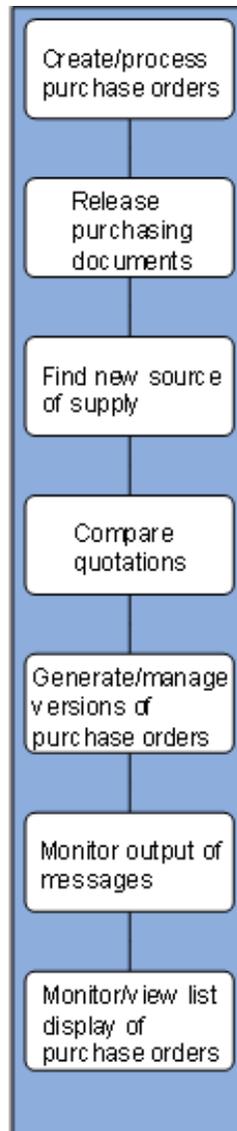


Abbildung 4.3: Geschäftsprozess: Bestellung [SBf]

eine Bestellung umzuwandeln [W.98, M.08]. Dabei beinhaltet diese infolge der Umwandlung die Informationen über die Menge, das Auslieferungsdatum, den Preis, die Zahlungsvereinbarungen oder den Service, welcher von einem Kreditor zur Verfügung gestellt wird. Dazu stellt eine Bestellung insbesondere den Bezugsbeleg für den Waren- und Rechnungseingang dar. Nachdem eine Bestellung durch das Ereignis *Create/process purchase orders* angelegt wird, wird diese entweder beim Anlegen freigegeben oder sie muss, wenn ein Freigabeprogramm vorhanden ist, im nächsten Schritt durch das Ereignis *Release purchasing documents* freigegeben werden. Ferner ist eine Bezugsquelle in dem Prozessschritt *Find new source of supply* zuzuordnen,

wenn dieser Vorgang nicht bereits bei der Bestellanforderung oder beim Anlegen der Bestellung durchgeführt ist. Dabei werden die Angebote der Kreditoren von einem zuständigen Mitarbeiter durch das Ereignis *Compare quotations* verglichen und zugeordnet. Letztendlich sind mehrere Versionen einer Bestellung in dem Vorgang *Generate/manage versions of purchase orders* generierbar. Das kann zur Folge haben, dass eine Bestellanforderung in mehrere Bestellungen gesplittet wird. Dadurch sind die weiteren Vorgänge des Beschaffungsprozesses ebenfalls aufgeteilt. Deshalb resultieren aus einer aufgeteilten Bestellung, die einen Bezug zu einer Bestellanforderung hat, mehrere Waren- und Rechnungseingänge sowie Kreditorenzahlungen. Die letzten beiden Ereignisse *Monitor output of messages* und *Monitor/view list display of purchase orders* werden zur Kontrolle des Bestellvorgangs von den Benutzern ausgeführt.

Falls das Material, welches zuvor bestellt ist, angeliefert wird, wird im SAP-ERP-System ein Wareneingangsbeleg zu dem **Wareneingang** erzeugt [W.98, M.08]. Dabei können die verspätet angelieferten Materialien angemahnt werden. Der Wareneingangsbeleg unterteilt sich in zwei Belege. Bei der Buchung des Wareneingangs in das Lager wird in dem Logistik-Modul ein Materialbeleg erzeugt. Zusätzlich wird ein Buchhaltungsbeleg bei den Vorgängen, die für die Materialbewertung relevant sind, im Finanzwesen-Modul erzeugt [W.98, M.08].

In SAP ERP existieren verschieden Arten von Bewegungen. Bewegungsarten, wie Wareneingang zur Bestellung, Rücklieferung an den Lieferanten, Freigabe aus Qualitätsprüfung und Zugang ohne Bestellung, können in dem System erfasst werden. Nachdem das Material eingegangen ist, ist das Konto des Kreditors in der Finanzbuchhaltung belastet und muss im weiteren Verlauf ausgeglichen werden.

Im nächsten Schritt erfasst und prüft die Rechnungsprüfung den **Rechnungseingang**. Bei Abweichung gibt das System eine Warnung aus und die Rechnung wird ggf. für die Zahlung gesperrt, so dass im Folgenden der zuständige Mitarbeiter den Fall weiter bearbeiten kann. Das Buchen der Rechnung beendet die Rechnungsprüfung. Daraufhin wird die Bestellentwicklung fortgesetzt und die **Kreditorenzahlung** kann vorgenommen werden. Die Kreditorenzahlung kann entweder manuell von einem zuständigen Mitarbeiter oder automatisch durch das Zahlungsprogramm ausgeführt werden [W.98, M.08].

Des Weiteren findet an dem Verknüpfungspunkt zwischen der Rechnungsprüfung und der Kreditorenzahlung die Integration zwischen der Materialwirtschaft und dem externen bzw. internen Rechnungswesen statt. Weiterhin existiert die Integration im Vertrieb, da bei der Erstellung einer Bestellanforderung diese einem Kundenauftrag zugeordnet werden kann. Ferner ist die Produktionsplanung durch die Bestandsanforderungen mit dem Beschaffungsprozess integriert.

4.2 Durchführung

Nachdem die einzelnen Schritte des Beschaffungsprozesses beschrieben wurden, wird in diesem Abschnitt die konkrete Durchführung des Prozesses in IDES vorgestellt.

Dazu gehört die Erläuterung zur Ausführung einer Beispielinstantz des Prozesses in dem SAP-System. Im Anschluss wird die Extraktion der Informationen sowie die Erzeugung einer XES-Logdatei mit der XSAPI-Anwendung beschrieben. Anschließend werden mehrere Prozessmodelle mit *ProM* erzeugt und analysiert.

Das Szenario fängt mit dem manuellen Anlegen einer Bestellanforderung an. Dazu wurde in dem *Easy Access Menu* des SAP-Systems die *Logistik* → *Materialwirtschaft* → *Einkauf* → *Banf* → *Anlegen* (ME51N)-Transaktion ausgeführt. Abbildung 4.4 stellt in (a) einen Ausschnitt der Benutzeroberfläche der Transaktion dar. Zusätzlich ist in dem unteren Bild ein Screenshot der EBAN-Tabelle, in welcher die Daten einer Bestellanforderung protokolliert werden, dargestellt. In dem Vorgang wird ei-



(a)

MANDT	BANFN	BNFPO	ERNAM	WERKS	MENGE	BADAT	UZEIT
926	0010015670	00010	BLYUFSHTEYN	1000	100,000	19.10.2011	00:00:00

(b)

Abbildung 4.4: Anforderung erstellen. Bild (a) zeigt das SAP-System-Screenshot. In (b) ist die EBAN-Tabelle dargestellt mit der, von dem Benutzer erzeugten Anforderung.

ne Anforderung für 100 Schaltungen des Rennrades für das Werk 1000 von einem Benutzer erzeugt, weil diese für die Herstellung des Rennrades nicht in dem Werk produziert werden können und eingekauft werden müssen. Nach der Speicherung des Belegs wird dieser in der Datenbank in der EBAN-Tabelle unter der generierten Nummer abgelegt, die in dem BANF-Feld gespeichert wird. In Abbildung 4.4 (b) ist die Nummer 0010015670 dargestellt, wobei diese den Beleg eindeutig identifiziert, weil das BANF-Feld das Primärschlüssel der EBAN-Tabelle ist. Dadurch ist der Datensatz eindeutig einer Bestellanforderung zugeordnet, die ihrerseits einem Belegfluss eindeutig zugeordnet werden kann.

Im nächsten Schritt wird eine Korrektur bzw. Änderung der Bestellanforderung durchgeführt. Anstatt 100 sollen 200 Schaltungen beschafft werden. Diesbezüglich ist die *Logistik* → *Materialwirtschaft* → *Einkauf* → *Banf* → *Ändern* (ME52N)-Transaktion zu wählen. Abbildung 4.5 stellt die Änderung der Bestellanforderung in dem SAP-System (a) und in den zugehörigen Tabellen (b) dar. Dabei zeigt das obere Bild die Oberfläche der Änderungstransaktion, wie sie von einem Benutzer in einem SAP-System ausgeführt wird. Ferner interagiert als Belegnummer in den zugehörigen Tabellen die Bestellanforderungsnummer. Falls ein anderer Beleg geändert wird, z.B. eine Bestellung, dann entspricht die Belegnummer der Bestellnummer. Abbildung (b) stellt die beiden Änderungstabellen in SAP dar. Die erste Tabelle ist



(a)

MANDANT	OBJECTCLAS	OBJECTID	USERNAME	UDATE	UTIME	
926	BANF	0010015670	BLYUFSHTEYN	19.10.2011	21:21:46	1

↓

MANDANT	OBJECTCLAS	OBJECTID	TABNAME	FNAME	VALUE_NEW	VALUE_OLD
926	BANF	0010015670	EBAN	MENGE	200.000	100.000

(b)

Abbildung 4.5: Anforderung ändern. Bild (a) zeigt das SAP-System-Screenshot. In (b) sind die CDHDR/CDPOS-Tabellen, mit den von den Benutzern erzeugten Änderungen, dargestellt.

die CDHDR-Tabelle. Diese beinhaltet den Änderungskopf. Dagegen stellt die zweite, die CDPOS-Tabelle, die Positionen der Änderung dar. In diesem Fall besteht die Bestellanforderung aus einem Datensatz. Die CDHDR-Tabelle beinhaltet dabei genau die Informationen, die für die Anwendung von *Process Mining* die Grundvoraussetzungen bilden. Dazu gehört die *OBJECTID*, mit der ein Geschäftsfall identifiziert werden kann, wobei in diesem Fall die Bestellanforderungsnummer gespeichert wurde. Weiterhin identifiziert das Feld *USERNAME* die ausführende Ressource. Letztendlich wird der Zeitstempel durch die Felder *UDATE* und *UTIME* repräsentiert. Die CDPOS-Tabelle (Tabelle 2 in Abbildung 4.5 (b)) stellt die zusätzlichen Informationen dar, die für weitere Process-Mining-Analysen verwendet werden können. Diesbezüglich bildet das Feld *TABNAME* die Tabelle ab, in der Informationen geändert sind. Weiterhin stellt die Position *FNAME* genau das Feld dar, welches geändert wurde.

In Bezug auf die Bestellanforderung wird die benötigte Ware von einem Mitarbeiter bestellt. Dabei werden in diesem Szenario ein Lieferant sowie die Einkaufinfosätze und die Konditionen bereits bei der Stammdatenpflege angelegt. Dadurch müssen diese Datensätze nicht noch einmal eingetragen werden und werden aus der Datenbank von dem System in den Bestellbeleg eingefügt. Um eine Bestellung mit Bezug zu einer Bestellanforderung auszuführen, ist die *Logistik* → *Materialwirtschaft* → *Einkauf* → *Bestellung* → *Anlegen* → *Lieferant/Lieferwerk bekannt* (ME21N)-Transaktion zu wählen. Abbildung 4.6 stellt die Oberfläche der Bestell-Transaktion dar. Ferner ist die Belegübersicht einzuschalten und *Selektionsvariante* → *Meine Bestellanforderungen* zu wählen, um die Bestellung einer Bestellanforderung zuzuordnen zu können. Nachdem die zuvor angelegte Bestellanforderung übernommen ist, werden die Daten aus der Anforderung in die Bestellung kopiert. Dazu sind die Schaltflächen in Abbildung 4.6 rot markiert. Anschließend werden die fehlenden Daten



Abbildung 4.6: SAP-System-Screenshot: Bestellung anlegen

von einem Mitarbeiter eingetragen. Dazu gehören die Einkaufsorganisation und der Lieferant. Bei der Speicherung der Bestellung generiert das System eine eindeutige Bestellnummer und legt die Informationen des Belegs in den EKKO/EKPO-Tabellen ab. Abbildung 4.7 stellt die beiden Tabellen mit der zuvor erzeugten Bestellung dar. Dabei enthält die EKKO-Tabelle den Belegkopf einer Bestellung, wobei diese in

MANDT	EBELN	AEDAT	ERNAM	VERKF
926	4500018470	26.10.2011	BLYUFSHTEYN	Hr. Schneider

1

MANDT	EBELN	EBELP	AEDAT	WERKS	MENGE	MEINS
926	4500018470	00010	26.10.2011	1000	200,000	ST

2

Abbildung 4.7: Screenshot: Bestellung in den EKKO/EKPO-Tabellen

Abbildung mit 1 gekennzeichnet wurde. Dazu gehören die Informationen über eine Bestellung, wie das Bestelldatum, die ausführende Person und der Verkäufer. Die EKPO-Tabelle stellt dagegen detaillierte Informationen aller Positionen einer Bestellung dar und ist mit 2 gekennzeichnet. Das Feld *EBELN* beinhaltet die Bestellnummer, welche eine Bestellung eindeutig in beiden Tabellen identifiziert. Dazu identifizieren in der EKPO-Tabelle die beiden Felder *EBELN* und *EBELP* eindeutig eine Position einer Bestellung. Dabei stellt *EBELP* die Position dar.

Die Bestellung wird in dem Szenario nachträglich geändert, weil die Bestellmenge noch einmal vergrößert werden soll. Dazu ist die *Logistik* → *Materialwirtschaft* → *Einkauf* → *Bestellung* → *Ändern* (ME22N)-Transaktion zu wählen. Abbildung 4.8 stellt die Oberfläche der Bestellung ändern-Transaktion dar. Falls die Lieferung auf-

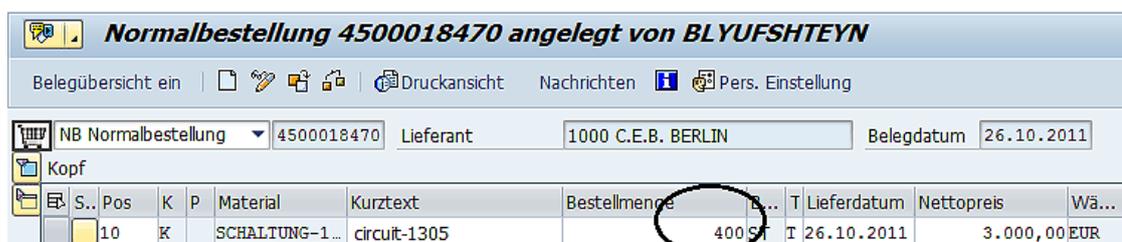


Abbildung 4.8: SAP-System-Screenshot: Bestellung ändern

grund der Änderung nicht eingehalten werden kann, warnt das SAP-System den

Benutzer [W.98, M.08]. Die Änderung von 200 Einheiten auf 400 Einheiten wird nach dem Speichern in der CDHDR- und CDPOS-Tabelle protokolliert. Abbildung 4.9 stellt die beiden Tabellen dar. Die obere, die CDHDR-Tabelle, enthält zwei Da-

MANDANT	OBJECTCLAS	OBJECTID	USERNAME	UPDATE	UTIME	CHANGE_IND
926	EINKBELEG	4500018470	BLYUFSHTEYN	26.10.2011	12:00:33	I
926	EINKBELEG	4500018470	BLYUFSHTEYN	26.10.2011	16:18:48	U

MANDANT	OBJECTCLAS	OBJECTID	TABNAME	FNAME	CHNGIND	VALUE_NEW	VALUE_OLD
926	EINKBELEG	4500018470	EKKO	KEY	I		
926	EINKBELEG	4500018470	EKPO	MENGE	U	400.000	300.000

Abbildung 4.9: Screenshot: Bestellung in den CDHDR/CDPOS-Tabellen ändern

tensätze. Der erste Datensatz beinhaltet in dem Feld *CHANGE_IND* den Eintrag *Insert* (I). Dieser Eintrag beschreibt den Datensatz, welcher bei der Erzeugung einer Bestellung in die Tabelle abgelegt wird. Dagegen beinhaltet der zweite Datensatz den Eintrag *Update* (U). Dieser bedeutet, dass der Beleg geändert wurde. Dazu sind die ausführende Ressource und der Zeitstempel in den Datensätzen gespeichert. Die untere, die CDPOS-Tabelle enthält detaillierte Informationen zu den Datensätzen. Dazu gehört die Tabelle, in welcher geändert bzw. eingefügt wurde. Weiterhin sind das betroffene Feld sowie der alte und der neue Wert des Feldes in der Tabelle enthalten.

Nachdem die Ware bestellt und eingegangen ist, bucht ein Mitarbeiter den Wareneingang. Dazu wird in dem Szenario angenommen, dass die Materialbestellung von dem Kreditor komplett geliefert wurde und keine Komplikationen entstanden sind. Um den Wareneingang zu buchen, ist die *Logistik* → *Materialwirtschaft* → *Bestandsführung* → *Warenbewegung* → *Wareneingang* → *Zur Bestellung* → *Bestell-Nr bekannt* (MIGO)-Transaktion zu wählen, wobei Abbildung 4.10 die Oberfläche der Transaktion darstellt. Weiterhin ist in dem linken Dropdown-Menü am oberen Bildrand

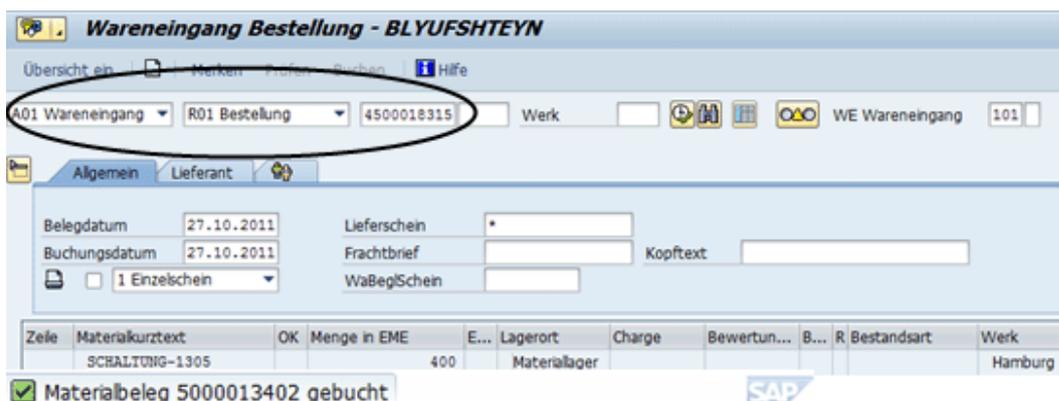


Abbildung 4.10: SAP-System-Screenshot: Wareneingang buchen

der Eintrag *A01 Wareneingang* und aus dem rechten Dropdown-Menü der Eintrag *R01 Bestellung* gewählt, weil der Wareneingang zu der bekannten Bestellung gebucht wird. Dazu wird die Bestellnummer in das Feld neben dem Dropdown-Menü eingegeben. Danach füllt das System die Datenfelder der Bestellung aus. Ferner

werden das Beleg- und das Buchungsdatum in den beiden Feldern *Belegdatum* und *Buchungsdatum* angegeben. Nach dem Speichern generiert das System eine Materialbelegnummer, welche den Datensatz eindeutig identifiziert und legt die Informationen in der EKBE-Tabelle, welche in Kapitel 3.2 erläutert wurde, ab. Abbildung 4.11 stellt die Tabelle dar. Das Feld *EBELN* stellt die Bestellnummer bezüglich

MANDT	EBELN	EBELP	VGABE	BELNR	MENGE	CPUDT	CPUTM	MATNR	ERNAM
926	4500018470	10	1	5000013402	400,000	27.10.2011	08:22:55	SCHALTUNG-1305	BLYUFSHTEYN

Abbildung 4.11: EKBE-Tabelle zur Wareneingang

des Wareneingangs dar. Dazu repräsentiert *EBELP* die Position der eingegangenen Bestellung. Das Feld *VGABE* ist bereits in 3.3.3 erläutert und beschreibt den Wareneingang. Weiterhin gibt *BELNR* die Materialnummer, welche von dem System beim Speichern des Belegs dem Wareneingangsbeleg zugewiesen wird. Der Zeitstempel ist durch die Vereinigung der beiden Felder *CPUDT*- und *CPUTM* dargestellt. Letztendlich ist die Ressource durch das Feld *ERNAM* repräsentiert.

Da die Bestellung im Lager eingetroffen ist, ist lediglich die Rechnung zu begleichen, um den Beschaffungsprozess abzuschließen. Dazu wird zuerst die eingegangene Rechnung von einem Benutzer in dem System erfasst. Diesbezüglich ist in *Easy Access Menu* die *Logistik* → *Materialwirtschaft* → *Logistik-Rechnungsprüfung* → *Belegerfassung* → *Eingangsbuchung hinzufügen* (MIRO)-Transaktion gewählt. Abbildung 4.12 veranschaulicht den Vorgang. Hinsichtlich der Zuordnung der Rechnung zu der

Abbildung 4.12: SAP-System-Screenshot: Rechnung erfassen

bezahlenden Filiale wird der Buchungskreis angegeben. Ferner werden die Grunddaten, wie das Rechnungsdatum, das Buchungsdatum und der Betrag eingepflegt. Im unteren Bildabschnitt ist zudem aus dem gewählten Dropdown-Menü-Eintrag *1 Bestellung/Lieferung* die Bestellnummer angegeben. Nach dem Sichern des Belegs fügt das System die Informationen in die EKBE-Tabelle ein. Abbildung 4.13 stellt die Tabelle der Bestellentwicklung dar. Der zweite Eintrag in der Tabelle beschreibt den Rechnungseingang. Im Gegensatz zum ersten Eintrag beinhaltet das Feld *VGABE*

MANDT	EBELN	EBELN	VBELN	BELNR	MENGE	CPUDT	CPUTM	MATNR	ERNAM
926	4500018470	101	5000013402	400,000	27.10.2011	08:22:55	SCHALTUNG-1305	BLYUFSHTEYN	
926	4500018470	102	5105608842	400,000	27.10.2011	11:18:28	SCHALTUNG-1305	BLYUFSHTEYN	

Abbildung 4.13: EKBE-Tabelle zum Rechnungseingang

den Eintrag 2. Dieser stellt das Ereignis *Invoice Receipt* dar, welches in 3.3.3 erläutert wurde. Ferner beinhaltet dieser Eintrag eine andere Belegnummer, welche in dem Feld *BELNR* abgelegt ist, wodurch der Datensatz eindeutig identifiziert wird. In Bezug zu einer Rechnung wird das Konto des Kreditors in dem System ausgeglichen. Dazu erfolgt in dem Szenario ein manueller Zahlungsausgang. In der Transaktion *Rechnungswesen* → *Finanzwesen* → *Kreditoren* → *Buchung* → *Zahlungsausgang* → *Buchen + Formulardruck* (F-58) wird dieser Vorgang ausgeführt. Abbildung 4.14 stellt die Oberfläche der Transaktion dar. Nach der Eingabe der

Abbildung 4.14: SAP-System-Screenshot: manuelle Zahlung

Grunddaten, dem dazugehörigen Buchungskreis, dem Zahlweg und der Hausbank wird die Zahlung von dem Benutzer erfasst. Anschließend sind der zu zahlende Betrag sowie die Kreditornummer anzugeben. Nach dem Ausführen der Operation durch *OP bearbeiten* wird die offene Position des Kreditors durch das System beglichen. Abschließend ist der Beleg mit der von dem System generierten Belegnummer in der Datenbank gespeichert und wird gleichzeitig gedruckt. Der Beleg wird in den EKBE- und BKPF-Tabellen protokolliert, wobei die erste Tabelle die Historie speichert. Die BKPF-Tabelle protokolliert alle von den Benutzern ausgeführten Bezahlungen. Da insbesondere die Ereignisse aus dem Finanzwesen-Modul gespeichert werden und diese für das Prozessmodell des Beschaffungsprozesses irrelevant sind, werden diese nicht in das Repositorium der XSAPI-Anwendung aufgenommen. Abbildung 4.15 stellt den protokollierten Beleg in der BKPF-Tabelle dar. Das Feld *BUKRS* beinhaltet den Buchungskreis, welcher für die Bezahlung verantwortlich ist. Der Zeitstempel entsteht durch die Verknüpfung des *CPUDT*- und des *CPUTM*-

MANDT	BUKRS	BELNR	CPUDT	CPUTM	USNAM
926	1000	1500002000	27.10.2011	14:43:47	BLYUFSHTEYN

Abbildung 4.15: Darstellung des Bezahlvorgangs in der BKPF-Tabelle

Feldes. Die Ressource wird durch das Feld *USNAM* abgebildet. Der Wert in dem BELNR-Feld identifiziert den Zahlbeleg.

4.3 Ergebnisse

Im Folgenden werden die Ergebnisse der Analyse einer XES-Logdatei mit dem *Prom* beschrieben und interpretiert, wobei die Logdatei mit der in dieser Arbeit entwickelten Anwendung aus einem SAP-ERP-System extrahiert wurde. Nach Darstellung der Evaluationsdaten folgt die Interpretation der Resultate in Bezug auf die erzeugten Geschäftsprozessmodelle und auf die untersuchte Fragestellung.

Diese Diplomarbeit wurde im Kontext des SAP-Systems der Universität Magdeburg bearbeitet. Entsprechend wurden als Evaluationsdaten Prozessausführungen eingesetzt, die an dem System während eines Beschaffungsprozesses ausgeführt wurden. Dabei sind sowohl die Evaluationsdaten eingesetzt, die bereits in der Datenbank des Systems vorhanden waren als auch die Evaluationsdaten, die selbstständig an dem System im Rahmen eines Szenarios ausgeführt wurden und in Kapitel 4.2 erläutert sind. Um möglichst repräsentative Ergebnisse zu erzielen und diese evaluieren zu können, wurde der Beschaffungsprozess insgesamt zehn Mal, zu den bereits vorhandenen Daten, ausgeführt. Diese Prozessausführungen enthalten die möglichen Ereignisausprägungen bestehend aus den Ereignissen die Bestellanforderung, die Bestellung, der Wareneingang, der Rechnungseingang und die Bezahlung, wobei die Ereignisse in Kapitel 4.1 erläutert wurden. Dazu unterscheiden sich die Prozessausführungen in der Reihenfolge und der Vollständigkeit der ausgeführten Ereignisse. Bezüglich der Reihenfolge wurden z.B. die Ereignisse der Wareneingang und der Rechnungseingang in den Prozessausführungen in verschiedenen Reihenfolgen ausgeführt. Die Vollständigkeit bezieht sich auf die Prozessausführungen, die z.B. nicht abgeschlossen sind oder anstatt bei der Bestellanforderung bereits bei der Bestellung anfangen.

Nachdem der Beschaffungsprozess in verschiedenen Ausprägungen ausgeführt wurde, wurden die zugehörigen Daten und die bereits vorhandenen Daten bezüglich des Beschaffungsprozesses aus dem Datenbanksystem des IDES-Modellunternehmens extrahiert. Dazu zeigt Tabelle 4.1 Informationen über die Anzahl der Einträge aus den für den Beschaffungsprozess verwendeten Tabellen aus dem Mandanten 926, welcher in IDES integriert ist. Die EBAN-Tabelle, die Informationen zu einer Bestellanforderung speichert, enthält 4304 Einträge, wobei hier die Einträge aus verschiedenen Werken gespeichert werden. Der Fokus der Evaluierung in dieser Arbeit liegt auf dem Beschaffungsprozess, welcher in dem Werk 1000 ausgeführt wurde. Deshalb beträgt die tatsächliche Menge der extrahierten Bestellanforderungen 1703 Einträge, weil

Table	# Records
EBAN	4304
EKKO	15068
EKPO	29218
EKBE	53360
CDHDR	25179
CDPOS	91219
BKPF	261278

Tabelle 4.1: Anzahl der Einträge in den Tabellen aus dem Beschaffungsprozess in SAP ERP

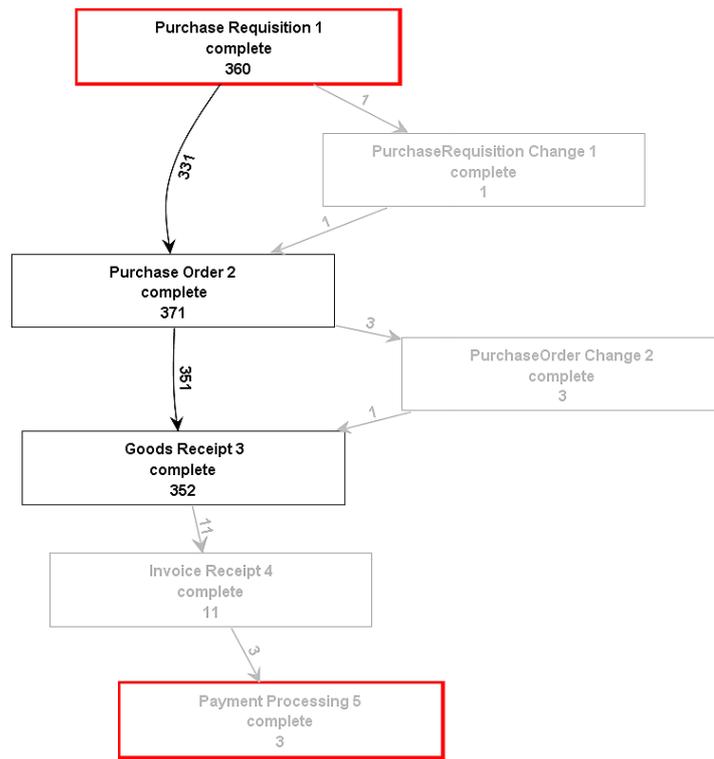
so viele Bestellanforderungen im Werk 1000 erzeugt wurden. Weiterhin enthält die BKPF-Tabelle nicht nur die Einträge aus dem Beschaffungsprozess, sondern auch aus anderen Geschäftsprozessen in dem Finanzwesen und Werken. Diese wurden deshalb für die Erzeugung des Prozessmodells des Beschaffungsprozesses nicht extrahiert. Dazu sind Bestellungen, die im Bezug zu einer Bestellanforderung erfolgt sind, sowohl in der EBAN- als auch in der EKKO/EKPO-Tabelle protokolliert. Diese Bestellungen wurden nach der Extraktion in dem Repositorium als Duplikatproblem betrachtet und dementsprechend entfernt. Tabelle 4.2 stellt die Anzahl der extrahierten Ereignisse aus den Tabellen dar. Das Ereignis *Create Purchase Requisition*

Activities	Occurrence
Create Purchase Requisition	1703
Change Purchase Requisition	18
Create Purchase Order	3562
Change Purchase Order	23
Goods Receipt	3299
Invoice Receipt	1505
Payment	41

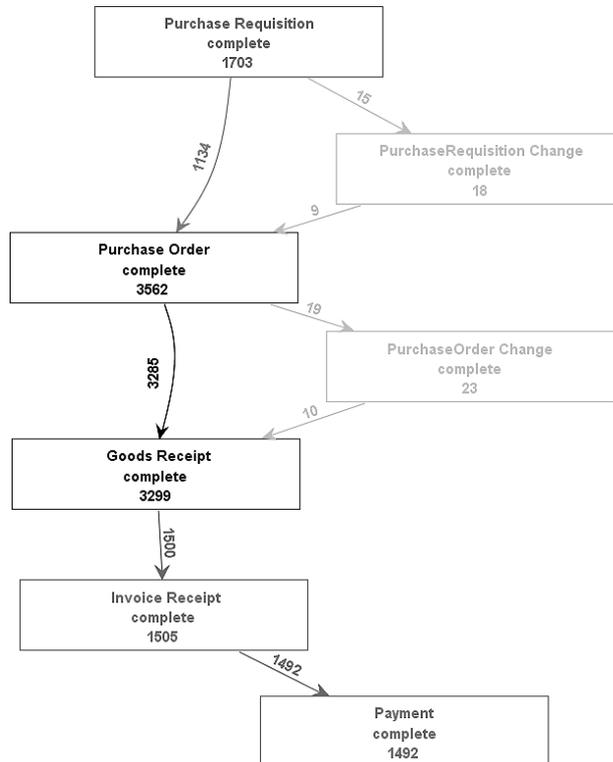
Tabelle 4.2: Anzahl der Aktivitäten im Beschaffungsprozess in IDES in dem Werk 1000

wurde aus der EBAN-Tabelle extrahiert. Die Ereignisse *Change Purchase Requisition* und *Change Purchase Order* stammen aus der CDHDR- und der CDPOS-Tabelle. Die Tabellen EKKO und EKPO protokollieren das Create Purchase Order-Ereignis. Wenn die bestellte Ware von einem Benutzer angenommen wird, resultiert dieser Vorgang in dem Goods Receipt-Ereignis, welches in der EKBE-Tabelle gespeichert wird. Falls eine Rechnung eingeht, wird das Invoice Receipt-Ereignis ebenfalls in der EKBE-Tabelle protokolliert. Letztendlich wird die Bezahlung der Rechnung, das Ereignis *Payment*, in der BKPF-Tabelle gespeichert.

Nach der Extraktion der Daten und der Erzeugung einer XES-Logdatei mit der XSAPI-Anwendung wurden diese mit dem *ProM* analysiert. Abbildung 4.16 stellt die gewonnenen Prozessmodelle mit dem *Heuristic Miner* gegenüber. Dabei reprä-



(a)



(b)

Abbildung 4.16: Gegenüberstellung der gewonnenen Prozessmodelle mit *Heuristic Miner*

sentiert ein Rechteck ein Ereignis. Weiterhin zeigt die Zahl in dem Rechteck, wie oft ein Ereignis aufgetreten ist. Dagegen repräsentieren die Zahlen an den Pfeilen die Anzahl der Ereignisübergänge. Ferner stellt Abbildung (a) das Geschäftsprozessmodell dar, welches unter Anwendung von *Business Application Programming Interfaces* (BAPIs) erzeugt wurde, wobei diese in 3.3.1 erläutert wurden. Dagegen entstand das Geschäftsprozessmodell in (b) unter Anwendung von Funktionsbausteinen, wobei diese in 3.3.1 dargestellt wurden. Die gleichen extrahierten Aktivitäten sowie die gleichen gewonnenen Prozessmodelle zeigen, dass für das letztendlich gewonnene Prozessmodell nicht relevant ist, welcher Extraktor verwendet wird.

Um die Prozessmodelle aus Abbildung 4.16 mit dem *ProM* zu erfassen, wurden zwei verschiedene XES-Dateien erzeugt. Die erste ist mit dem Modell generiert, welches BAPIs zur Extraktion in dem Extraktor der in dieser Arbeit entwickelten Anwendung verwendet. Diese Ereignislogdatei ist 300KB groß und enthält 1101 Ereignisse in 399 Spuren. Die zweite Datei wird mit dem Extraktor gewonnen, welcher im Modell die Funktionsbausteine integriert hat. Diese Datei ist 3 MB groß. Dabei wurden 11602 Ereignisse auf 4122 Spuren verteilt extrahiert.

Ferner wurden weitere Geschäftsprozessmodelle mit dem Werkzeug *ProM* erfasst. Abbildung 4.17 stellt ein Geschäftsprozessmodell des Beschaffungsprozesses dar, welches mit dem Fuzzy-Miner-Plugin erfasst wurde. Die Rechtecke mit den abgerun-

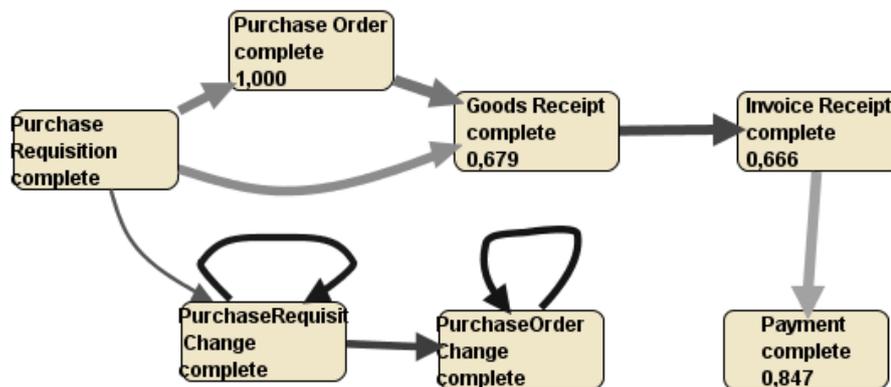


Abbildung 4.17: Prozessmodell mit *Fuzzy Miner*

deten Ecken repräsentieren die Ereignisse, wobei sie den Ereignisnamen, den Typ des Ereignisses und die Wahrscheinlichkeit beinhalten, mit welcher ein Ereignis ausgeführt wurde. Z.B. wird das Ereignis *Goods Receipt* mit der Wahrscheinlichkeit 0,679 von den Benutzern des Systems bei der Ausführung des Beschaffungsprozesses ausgeführt. Die Schleifen bei den Ereignissen *Purchase Requisition Change* und *Purchase Order Change* deuten darauf hin, dass die Ereignisse wiederholt ausgeführt wurden. Weiterhin unterscheiden sich die gerichteten Verbindungen in der Intensität ihrer Darstellung. Z.B. ist die Verbindung von *Purchase Requisition* nach *Goods Receipt* dicker als die Verbindung nach *Purchase Requisition Change*. Dadurch wird gezeigt, dass der Übergang mit der dickeren Verbindung öfter stattfindet als mit der dünneren. Des Weiteren wurde die XES-Logdatei analysiert, die mit dem Funktionsbausteinextraktor generiert wurde. Zur Erfassung des Geschäftsprozessmodells

in Abbildung 4.17 wurde die gleiche Anzahl der Ereignisse und der Spuren verwendet, wie in Abbildung 4.16. Hierbei sind die Unterschiede in den Modellen besonders in der Reihenfolge der aufgetretenen Ereignisse auffallend. Weiterhin ist die Reihenfolge zu dem in Kapitel 4.1 skizzierten Modell ebenfalls unterschiedlich. Der erste Unterschied ist durch die unterschiedlichen Algorithmen in den Plugins begründet, während der zweite dadurch resultiert, dass in Kapitel 4.1 ein idealer Beschaffungsprozess dargestellt ist, während in der realen Ausführung Fehler durch die Benutzer entstehen oder unerwartete Vorgänge ausgeführt werden.

Im Vergleich dazu ist in Abbildung 4.18 ein Prozessmodell dargestellt, welches ebenfalls mit dem Fuzzy-Miner-Plugin generiert wurde. Bezüglich dieses Prozessmodells

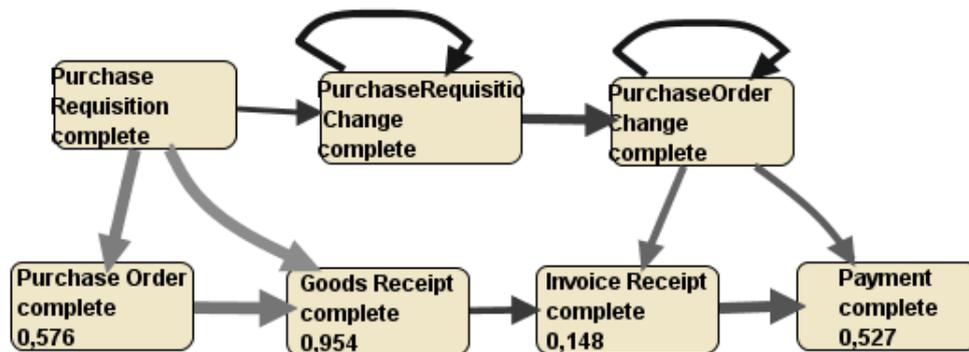


Abbildung 4.18: Prozessmodell mit *Fuzzy Miner* und zeitlichem Intervall

wurde ein zeitliches Intervall benutzt, welches die Anzahl der Daten in der Datei auf 1141 Ereignisse in 408 Spuren reduziert. Das Prozessmodell veranschaulicht, wie sich die Anzahl der extrahierten Ereignisse auf das Resultat auswirkt. Das Ereignis *PurchaseOrder Change* folgt dem Ereignis *PurchaseRequisition Change*, das bedeutet, dass nachdem eine Bestellanforderung geändert wurde, die dazugehörige Bestellung geändert wird. Dabei ist die Bestellung laut dem Modell im aktuellen Zustand noch nicht erzeugt und dadurch kann keine Änderung dieser stattfinden.

Ein weiteres Prozessmodell wurde mit dem *Genetic Miner* erfasst. Dieses Modell stellt genau das Geschäftsprozessmodell des Beschaffungsprozesses dar, welches in Kapitel 4.1 beschrieben und in Kapitel 4.2 in dem SAP-System ausgeführt wurde. Abbildung 4.19 zeigt das gewonnene Modell. Die Rechtecke stellen die Ereignisse dar, wobei die gerichteten Verbindungen die Ereignisübergänge zeigen. Das Geschäftsprozessmodell fängt mit einer Bestellanforderung an, welche durch das Ereignis *Purchase Requisition* in Abbildung gezeigt ist. Daraufhin kann die Bestellanforderung durch das Ereignis *PurchaseRequisition Change* geändert werden oder die Ware wird im Ereignis *Purchase Order* im Bezug zu der Bestellanforderung bestellt. Falls die Bestellanforderung geändert wird, kann im Folgenden die Ware durch das Ereignis *Purchase Order* bestellt werden. Nach der Bestellung wird diese in dem Ereignis *PurchaseOrder Change* geändert oder die Ware wird durch das Ereignis *Goods Receipt* von einem Mitarbeiter angenommen. Des Weiteren kann nach der Bestellung die Rechnung für die bestellte Ware eingehen, wobei dieser Vorgang durch das Ereignis *Invoice Receipt* in Abbildung dargestellt wurde. Nach der Änderung einer Bestellung kann die Ware, wie nach der regulären Bestellung, erhalten

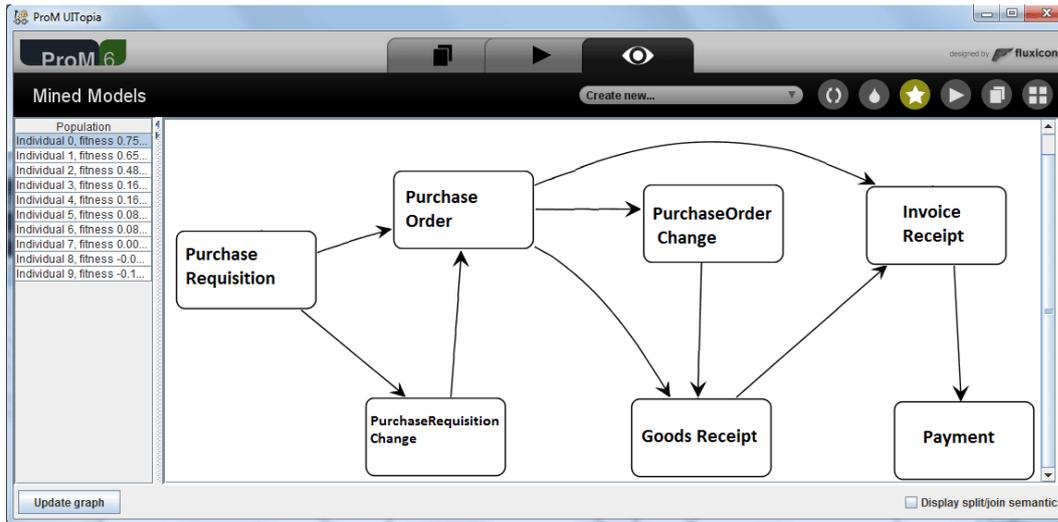


Abbildung 4.19: Prozessmodell mit *Genetic Miner*

werden. Falls die Ware angenommen wird, endet der Beschaffungsprozess mit dem Ereignis *Payment*, wenn davor die Rechnung erhalten wurde. Unabhängig von dem Erhalt der Ware kann die bereits angenommene Rechnung von einem Mitarbeiter in dem Vorgang *Payment* bezahlt werden.

Ferner beruht der *Genetic Miner* auf evolutionären Algorithmen und verwendet Individuums- und Fitness-Funktionen, um die Modelle zu generieren. Dabei ist ein Individuum ein mögliches Prozessmodell. Die Fitness-Funktion gibt an, wie gut das Individuum die Logdatei in dem gewonnenen Prozessmodell wiedergibt [Aal07]. Bereits bei null Individuen erreicht das Plug-in eine Fitness von 0,7 und das Geschäftsprozessmodell stellt die in Abschnitt 4.2 erläuterte Durchführung des Prozesses dar.

Des Weiteren wurde ein Geschäftsprozessmodell des Beschaffungsprozesses mit dem Business Process Modeling Notation-Plugin (BPMN-Plugin) erfasst, wobei Abbildung 4.20 das Modell darstellt. Dabei repräsentiert ein Rechteck mit abgerundeten Ecken ein Ereignis. Der Workflow wird durch die gerichteten Verbindungen zwischen den Ereignissen gezeigt. Weiterhin werden in BPMN Konnektoren verwendet, um die Ereignisse miteinander zu verbinden, wobei in Abbildung die Ereignisse durch den XOR-Konnektor verbunden wurden. Das Modell zeigt deutlich, dass in dem Beschaffungsprozess Prozessverzweigungen vorhanden sind. Ein Beispiel einer Prozessverzweigung in dem Beschaffungsprozess ist das Ereignis *Purchase Requisition*, wobei dadurch eine Bestellanforderung erzeugt wird. Nach dem Erzeugen einer Bestellanforderung wird entweder das Ereignis *Purchase Requisition Change* ausgeführt, welches eine Änderung einer Bestellanforderung darstellt oder das Ereignis *Purchase Order*, welches eine Bestellung im Bezug zu einer Bestellanforderung darstellt. Dabei verdeutlicht der erste Konnektor die Entweder-Oder-Beziehung nach dem Ereignis der Bestellanforderung.

Das erfasste Geschäftsprozessmodell stellt das in Kapitel 4.1 beschriebene Modell eines Beschaffungsprozesses dar, wobei dieses ein Syntaxfehler aufweist, welches auf die Fehler in dem Algorithmus des BPMN-Plugins zurückzuführen ist. Der Syntax-

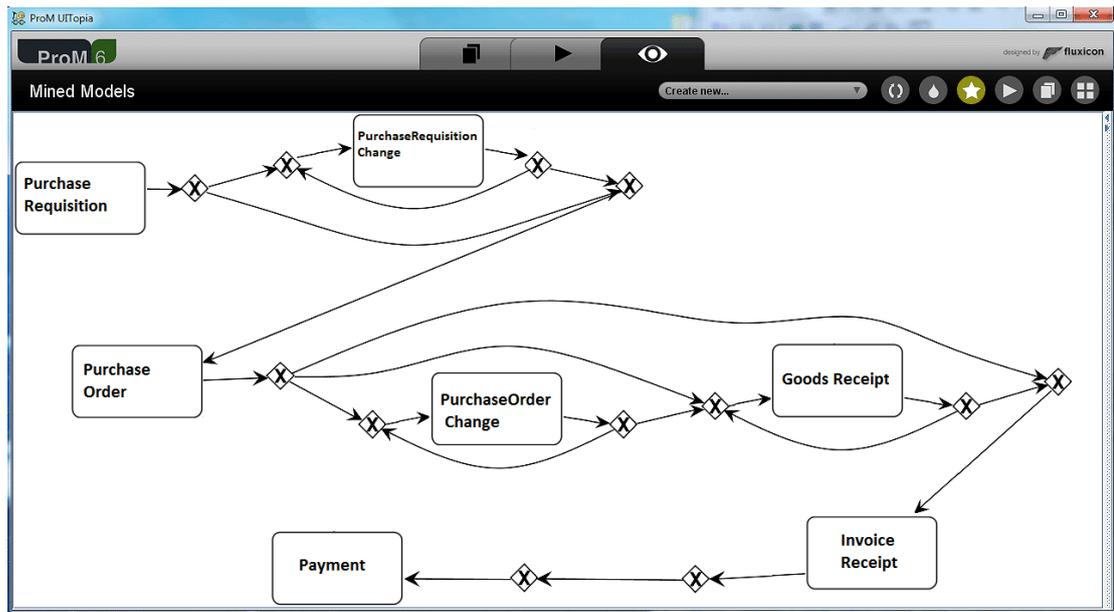


Abbildung 4.20: Prozessmodell mit *BPMN*

fehler tritt nach dem Ereignis *Invoice Receipt* auf, welches den Erhalt einer Rechnung repräsentiert. Anschließend folgen zwei XOR-Konnektoren, wobei das Workflow des Beschaffungsprozesses mit dem Ereignis *Payment*, welches das Bezahlen einer Rechnung darstellt, abgeschlossen wird. Dabei stellen die beiden XOR-Konnektoren den Syntaxfehler dar, weil in BPMN zwei Arten von XOR-Konnektoren existieren, wobei die existierenden Konnektoren nicht die Konnektoren in Abbildung 4.20 darstellen. Der erste Konnektor entspricht dem XOR-Split-Konnektor, welcher eine eingehende und zwei ausgehenden Verbindungen hat und der zweite dem XOR-Join-Konnektor, wobei dieser zwei eingehende und eine ausgehende Verbindung aufweist. Abbildung 4.21 stellt in (a) den XOR-Split-Konnektor und in (b) den XOR-Join-Konnektor dar. Hier wird deutlich, dass die beiden Konnektoren immer drei Verbindungen auf-

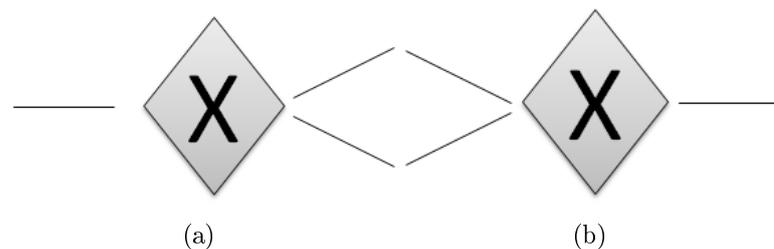


Abbildung 4.21: XOR-Konnektor in BPMN. Bild (a) zeigt den XOR-Split-Konnektor. In (b) ist der XOR-Join-Konnektor dargestellt.

weisen. Dagegen sind die Konnektoren in Abbildung 4.20 mit einer Eingang und einer Ausgangs-Verbindung dargestellt.

Schließlich wurde das Geschäftsprozessmodell des Beschaffungsprozesses in SAP ERP mit dem Alpha-Algorithm-Plugin erfasst, welches in Petri-Netze-Notation dar-

gestellt wird. Abbildung 4.22 zeigt das Modell, wobei der Prozess in einem Startzustand beginnt, welcher durch den Kreis mit der Zahl 1 gezeigt wird und einem Endzustand endet, welcher durch den Kreis dargestellt wird, der ausschließlich eingehenden Kanten hat. Aus dem Startzustand gehen zwei Kanten aus, die eine zu der

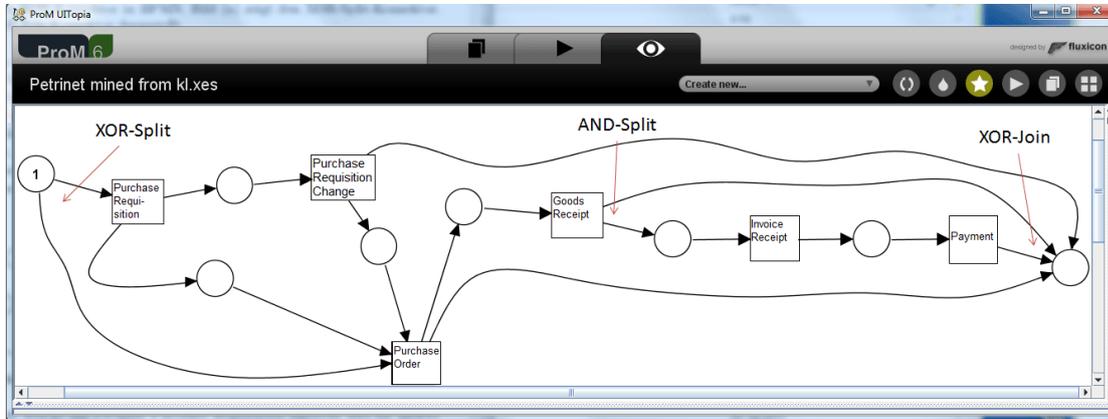


Abbildung 4.22: Prozessmodell mit Alpha-Algorithm-Plugin

oberen Transition, welche das Ereignis *Purchase Requisition* darstellt und die andere zu der unteren Transition *Purchase Order*, wobei durch die beiden Kanten ein XOR-Split-Konnektor dargestellt wird. Für das Geschäftsprozessmodell des Beschaffungsprozesses bedeutet das, dass dieses entweder mit einer Bestellanforderung oder mit einer Bestellung anfängt. Ein weiterer Konnektor in dem Geschäftsprozessmodell ist der XOR-Join-Konnektor, welcher z.B. in dem Endzustand gezeigt wird, wobei hier vier Kanten eingehen. Das bedeutet, dass ein Beschaffungsprozess entweder nach der Ausführung einer Bestellung, nach dem Begleichen einer Rechnung, nach dem Erhalt der bestellte Ware oder nach einer Änderung einer Bestellanforderung beendet wurde. Diese vier Fälle des Beschaffungsprozesses wurden in dieser Arbeit in dem SAP-ERP-System ausgeführt. Z.B. wurde ein Prozess ausgeführt, welcher durch eine Bestellanforderung startet und im Bezug darauf eine Bestellung ausgeführt wird, wobei keine weiteren Schritte des Beschaffungsprozesses ausgeführt wurden.

4.4 Fazit

In Kapitel 4 wurde die Validierung des Prototyps dargestellt. Dazu ist zuerst der Beschaffungsprozess in SAP ERP erläutert worden und es wurde anschließend dargestellt, wie dieser in einem SAP-System ausgeführt werden kann. Im Anschluss sind mehrere XES-Logdateien mit der in dieser Arbeit entwickelten Anwendung extrahiert und mit dem *ProM* analysiert worden. Dabei wurden mit verschiedenen Plugins des Werkzeugs unterschiedliche Geschäftsprozessmodelle des Beschaffungsprozesses erfasst.

In der Evaluation wurde zuerst betrachtet, wie sich die Modelle durch die beiden Extraktionsverfahren, dem mit den BAPIs- und dem mit dem Funktionsbaustein,

unterscheiden. Hier wurde festgehalten, dass für das Prozessmodell nicht relevant ist, welcher Extraktor verwendet wird, weil dabei das gleiche Geschäftsprozessmodell erfasst werden konnte (siehe Abbildung 4.16).

Weiterhin wurden die Prozessmodelle nach der Anzahl der extrahierten Ereignisse evaluiert. Diesbezüglich zeigten sich erhebliche Unterschiede bei den generierten Modellen. Bei den Modellen mit 1000 Ereignissen wurden Fehler in der Logik festgestellt, weil diese von dem Beschaffungsprozess abweichen, welcher in Kapitel 4.1 beschrieben und in Kapitel 4.2 in dem SAP-System ausgeführt wurde. Dies führt zu der Erkenntnis, dass bei der Analyse der Geschäftsprozessmodelle bereits Kenntnisse in den analysierten Geschäftsprozessen vorhanden sein müssen, weil sonst die Fehler nicht erkannt werden können.

Abschließend lässt sich festhalten, dass nicht alle der ausprobierten Plugins in *ProM 6* das Geschäftsprozessmodell des Beschaffungsprozesses darstellen, welcher in Kapitel 4.1 beschrieben und in Kapitel 4.2 ausgeführt wurde. Bereits beim Inspizieren der XES-Logdatei mit nur einer Spur ist das Prozessmodell sichtbar. Trotzdem wird das Prozessmodell nicht von allen Plugins erkannt. Insbesondere das Heuristic-Miner-Plugin und das Genetic-Miner-Plugin haben in allen Fällen das beschriebene Geschäftsprozessmodell des Beschaffungsprozesses aus der XES-Logdatei extrahieren können.

Kapitel 5

Zusammenfassung und Ausblick

5.1 Zusammenfassung

Das vorrangige Ziel dieser Arbeit bestand in der Implementierung eines Prototypen, welcher Process-Mining-Analysen auf Log-Daten aus SAP-Systemen ermöglicht, wobei die Log-Daten aus den Daten bestehen, welche bei den Ausführungen der Geschäftsprozesse in SAP ERP protokolliert werden. Dazu wurde der Verlauf dieser Arbeit in drei Teile unterteilt. Im ersten Teil wurde die Vorbereitungsphase erläutert, wobei dazu der Beschaffungsprozess und der Schaden-/Leistungsabwicklungsprozess für Lebensversicherung betrachtet wurden. In dieser Phase wird der Prototyp an das SAP-System angepasst. Dabei werden die Tabellen für den zu analysierenden Geschäftsprozess identifiziert und anschließend der Belegfluss zwischen den Daten aus den Tabellen hergestellt. Im Gegensatz zu der Arbeit von Martijn van Giessel wurde diskutiert, wie genau nur die Tabellen, die für die Rekonstruktion des Belegflusses notwendig sind, in der Vorbereitungsphase identifiziert werden können. Hier ist festgehalten worden, dass in der Vorbereitungsphase mehrere Quellen zur Identifikation der benötigten Informationen benötigt werden. Dazu gehören vor allem die SAP-Referenzmodelle, aber auch das *Easy Access Menu* und die Mitarbeiter des Unternehmens können einen Einblick in die tatsächlich ausgeführten Vorgänge ermöglichen, um anschließend die dazugehörigen Tabellen identifizieren zu können. Der Einfluss der Vorbereitungsphase ist für das ganze Vorgehen entscheidend, weil hier definiert wird, wie detailliert der Geschäftsprozess untersucht werden wird und wie die zugehörige Informationen identifiziert werden. Letztendlich wurde das Analyseziel festgehalten und dementsprechend wurden in der Extraktionsphase die Informationen aus SAP ERP extrahiert. Diesbezüglich bestand das Ziel darin, das Geschäftsprozessmodell des Beschaffungsprozesses aus dem SAP-ERP-System des IDES-Unternehmens mit Process Mining-Methoden zu analysieren.

Zu dem zusätzlichen Ziel dieser Arbeit gehört die Anbindung an ein SAP-System, welches mit dem entwickelten Prototyp erreicht wurde. Im Gegensatz dazu wurden die bisherigen Arbeiten von Martijn van Giessel, Segers und Buijs auf den Daten ausgeführt, die aus SAP manuell extrahiert wurden. Deshalb bestand der zweite Teil dieser Arbeit in dem Aufbau einer Anbindung des Prototyps an das SAP-System.

Dabei galt es zu untersuchen, welche Schnittstellentechnologien von SAP bereitgestellt werden und welche sich für diese Arbeit eignen. Zwei der Technologien wurden in dieser Arbeit angewendet, weil sie insbesondere von SAP für die Anbindung von Fremdsystemen entwickelt wurden und sie die Extraktion von den Informationen ermöglichen, welche von den Business-Objekttypen erzeugt werden und in den zugehörigen Tabellen gespeichert werden.

Die erste Schnittstellentechnologie, die in dem Prototyp integriert wurde, sind die BAPIs. Diese eignen sich dadurch besonders gut, da sie die SAP-Business-Objekte abbilden, aus welchen ein Prozessschritt aufgebaut wird. Dagegen bestand die Schwierigkeit darin, dass die BAPIs eine grafische Verbindung in Business-Objekt-Modellen untereinander aufweisen. Diese ist jedoch nicht durch z.B. Primärschlüssel abgebildet. Die Verbindung musste für jedes Objektpaar einzeln in den Modellen identifiziert werden. Letztendlich wurden die BAPIs, laut den Informationen aus der Vorbereitungsphase für den Beschaffungsprozess in SAP ERP, miteinander verknüpft und bilden dadurch den Extraktor, wobei diese für jeden beliebigen Prozess aus SAP ERP verknüpft werden können.

Die zweite Technologie stellten die Funktionsbausteine dar. Diese wurden auch in den Adaptern des Aris PPM Werkzeuges benutzt, wobei dort für jede Komponente ein Adapter benutzt wird. Im Gegensatz zu Aris PPM wurde letztendlich eine Logdatei erzeugt, die mit Process-Mining-Methoden analysiert werden kann. Des Weiteren werden in dieser Arbeit die Funktionsbausteine für die Extraktion der Informationen aus der Datenbank eines SAP-Systems benutzt. Ein Funktionsbaustein wurde wie die BAPIs in einem Extraktor integriert. Dieser extrahierte die Daten aus den Tabellen, die in der Vorbereitungsphase für den Beschaffungsprozess identifiziert wurden, in ein Repositorium, wobei hier jeder beliebige Prozess aus SAP ERP benutzt werden kann. In dem Repositorium findet zuerst die Bereinigung der Daten statt und anschließend wird der Belegfluss unter Benutzung der Primärschlüssel hergestellt, wobei im Gegensatz zur BAPIs die Verbindung zwischen den Tabellen und so die Rekonstruktion des Belegflusses aus einem SAP-System, wie in Kapitel 3.2 beschrieben wurde, entnommen werden kann.

In dem dritten Teil der Arbeit bestand das Ziel in der Anbindung des Prototyps an das *ProM*. Dazu ist die XES-Logdatei verwendet worden, die den Input für das *ProM* darstellt. Diese beinhaltet einen Prozess, die zugehörigen Spuren und die Ereignisse. Für die Rekonstruktion der Geschäftsprozessmodelle wurden den Ereignissen Attribute zugewiesen. Dazu gehörten die Ressource, der Transitionstyp, der Zeitstempel und die Bezeichnung.

Nach der Erzeugung einer XES-Logdatei wurde diese mit *ProM 6* analysiert, wobei zu diesem Zweck die verschiedenen Plugins der Anwendung benutzt wurden. Abschließend ist der Verlauf evaluiert worden. Diesbezüglich sind zuerst die Vorgänge in dem Beschaffungsprozess erläutert und anschließend in dem SAP-System des IDES-Unternehmens ausgeführt worden. Letztendlich sind die Ergebnisse der Analyse in Form von Geschäftsprozessmodellen mit *ProM* dargestellt worden und es wurde auf die Probleme und Fehler hingewiesen.

Mit dem Prototyp wird ein ETL-Prozess umgesetzt. Die Anwendung extrahiert Daten aus einem SAP-System sowie transformiert und lädt die Daten in eine XES-Logdatei. Insgesamt ist es keine triviale Aufgabe, *Process Mining* in einem datenzentrischen System, wie das SAP-System es ist, auszuführen, wobei in dieser Arbeit gezeigt worden ist, welche Schritte dazu notwendig sind und wie diese ausgeführt werden können.

5.2 Diskussion

Im Laufe dieser Arbeit wurde eine Logdatei aus einem SAP-ERP-System extrahiert, wobei die Logdatei die Informationen beinhaltet, die während einer Prozessausführung von dem SAP-System in einer Datenbank protokolliert wurden. Anschließend wurden die Geschäftsprozessmodelle des Prozesses mit dem *ProM* aus der Logdatei erfasst. Dabei ergaben sich Schwierigkeiten bei dem SAP-System einerseits und bei dem *ProM* andererseits. Des Weiteren haben sich Potenziale bei der Gewinnung des Geschäftsverständnisses in dieser Arbeit gezeigt.

Bei dem SAP-System sind die Schwierigkeiten während der Vorbereitungsphase entstanden. Bezüglich dieser Phase sind Informationen aus verschiedenen Bereichen des SAP-Systems eingeflossen, um weitere Informationen zu erhalten, welche zu der Identifizierung der relevanten Tabellen führten. Diese Tabellen protokollieren die Daten, welche während der Ausführung eines Geschäftsprozesses erzeugt werden. Das Problem trat auf, weil die Informationen aus vielen unterschiedlichen Quellen stammen und nicht alle Informationen tatsächlich für ein bestimmtes System gelten. Z.B. protokolliert die EKBE-Tabelle die Historie eines Einkaufsbeleges, wie in Kapitel 3.3.3 in Abbildung 3.15 gezeigt wurde, wobei auch das Ereignis der Bezahlung einer Ware hier gespeichert wird. Dabei wird in dem IDES-System in dieser Tabelle das Ereignis der Bezahlung nicht gespeichert, sondern ausschließlich in der BKPF-Tabelle der Finanzwesen-Komponente. Deshalb sollten in zukünftigen Arbeiten die Quellen genau untersucht werden und nur die in der Vorbereitungsphase verwendet werden, welche für das System gelten mit dem gearbeitet wird.

Eine weitere Schwierigkeit ergab sich bei der Erfassung von Geschäftsprozessmodellen mit dem *ProM*. Dazu wurden mehrere Plugins benutzt, wobei die benutzten Plugins, außer das Heuristic-Miner-Plugin und das Genetic-Miner-Plugin, erfassten nicht das erwartete und in Kapitel 4 erläuterte Geschäftsprozessmodell des Beschaffungsprozesses. Des Weiteren sind Fehler in den Modellen gefunden worden, wie in Kapitel 4.3 gezeigt wurde. Diesbezüglich sollten in den zukünftigen Arbeiten die Plugins des *ProMs* untersucht werden und ausschließlich die Plugins benutzt werden, die fehlerfrei das Geschäftsprozessmodell erfassen.

Das Potenzial bietet in einem Process-Mining-Projekt das Geschäftsverständnis, welches sich mit dem *Process Mining* gewinnen lässt. Neben dem Wissen über die in einem Unternehmen bei der Ausführung der Geschäftsprozesse tatsächlich stattgefundenen Ereignisse kann mit den Process-Mining-Methoden auch weiteres Wissen über die Geschäftsprozesse gewonnen werden, wie bereits in Kapitel 2.2 erläutert wurde. Die während dieser Arbeit gewonnene Erfahrung zeigt, dass durch die An-

wendung von Process-Mining-Methoden auf ein SAP-ERP-System die Geschäftsprozesse des Systems sich verbessern und optimieren lassen.

5.3 Ausblick

Diese Arbeit zeigt deutlich, dass *Process Mining* generell im Zusammenhang mit einem SAP-System ausführbar ist. Dennoch kristallisierten sich einige Schwierigkeiten heraus, die in den zukünftigen Arbeiten behandelt werden sollten. Dazu gehört die Vorbereitungsphase. Die Vorbereitungsphase wird für jeden Prozess aus SAP ERP ausgeführt, wobei diese bisher nicht automatisiert wurde und deshalb manuell durchgeführt werden muss. Diesbezüglich ist der Suchraum für die zu einem Prozess zugehörigen Objekte, Tabellen und deren Verknüpfung untereinander sehr groß und nicht immer zugreifbar, weil z.B. die Referenzmodelle, die in dem *Business Navigator* in SAP ERP integriert waren, SAP-seitig in das *Business Process Repository* eingefügt und im *Solution Manager* integriert wurden. Dabei ist der *Solution Manager* ein Werkzeug, das bei SAP zusätzlich erworben werden kann. Dieser Suchraum sollte in weiteren Arbeiten untersucht und angepasst werden, um die Prozessschritte eines Geschäftsprozesses in SAP identifizieren zu können und infolgedessen die zugehörigen Tabellen aus einem Datenbanksystem eines SAP-ERP-Systems erhalten zu können. Des Weiteren sollten die Metadaten untersucht werden, welche im *Business Object Repository* (BOR) für die Business-Objekttypen (BOs) und die dazugehörigen BAPIs gespeichert werden, wobei ein BOR ein Repositorium für die Speicherung dieser Objekte und der dazugehörigen Ereignisse ist. Die Metadaten spezifizieren die BOs und die BAPIs und geben deshalb einen Überblick darüber, welche BOs und BAPIs in BOR vorhanden sind und wie sie miteinander zu einem neuen Geschäftsprozess verknüpft werden. Dadurch gelingt die Extraktion der Daten aus einem SAP-ERP-System, ohne die BOs und die dazugehörigen BAPIs oder die dazugehörigen Tabellen vorab identifizieren zu müssen.

Ferner sollte das *Customizing* des Prototyps z.B. in eine XML-Datei ausgelagert werden, so dass dies auch durch einen Analysten ohne Programmierkenntnisse durchgeführt werden kann. Dazu sollten zur Extraktion der Daten in dem Extraktor weiterhin die Funktionsbausteine verwendet werden.

Eine Online-Analyse ist für die weiteren Forschungsarbeiten ebenfalls interessant. Diesbezüglich sollte der Prototyp um die Möglichkeiten der inkrementellen Updates erweitert werden. Diese Erweiterung wird die Analyse deutlich flexibler und dynamischer machen.

Zusätzlich zu den strukturellen Erweiterungen sollten in der Anwendung die Filterfunktionen erweitert werden, um die Daten nach verschiedenen Indikatoren analysieren zu können. Außer dem zeitlichen und dem örtlichen Intervall sind Filterfunktionen denkbar, welche die Daten z.B. nach der Menge oder der Ausführungsdauer des Prozesses filtern.

Die Daten, die von dem Prototyp extrahiert werden, ermöglichen die Erfassung eines Geschäftsprozessmodells mit dem *ProM*. Das Werkzeug *ProM* mit der großen Anzahl an Plugins bietet weitere Analysen an. Dazu sind weitere Attribute in der

XES-Logdatei notwendig. Diese werden in SAP - in den Tabellen - protokolliert und können durch eine Erweiterung von dem Prototyp ebenfalls extrahiert werden.

Anhang A

BPMN zur Schaden- /Leistungsabwicklungsprozess

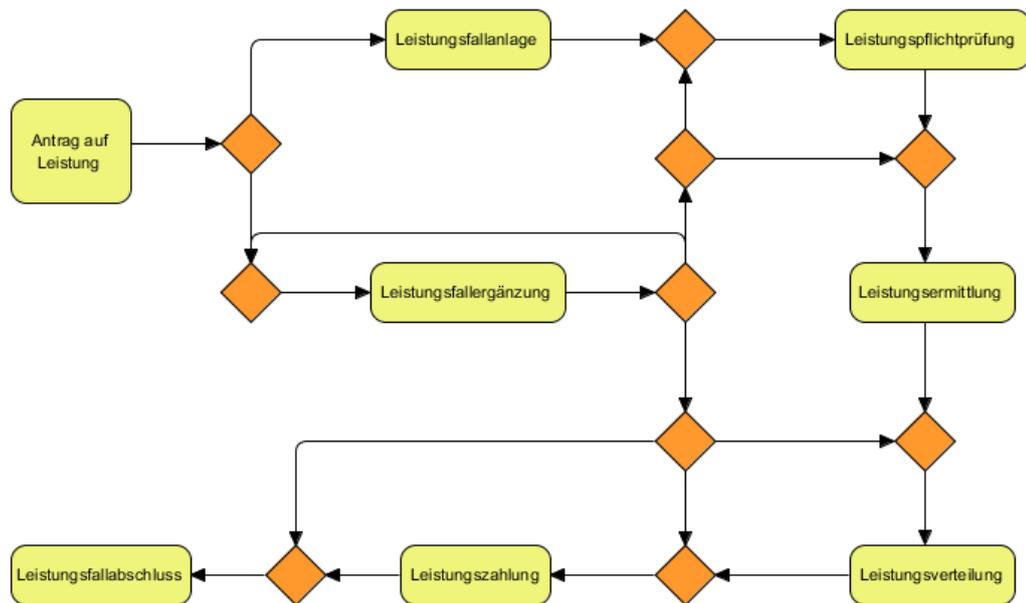


Abbildung A.1: Die Rechtecke zeigen die Ereignisse, wobei die Rauten die XOR-Konnektoren darstellen

Anhang B

Schaden- /Leistungsabwicklungsprozess

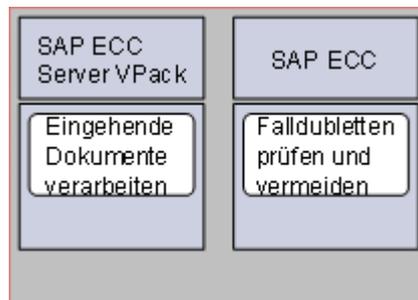


Abbildung B.1: Antrag auf Leistung



Abbildung B.2: Leistungsfallanlage



Abbildung B.3: Leistungspflichtprüfung



Abbildung B.4: Leistungsermittlung



Abbildung B.5: Leistungsverteilung



Abbildung B.6: Leistungszahlung



Abbildung B.7: Leistungsfallabschluss

Anhang C

Weitere Entwicklungsobjekte

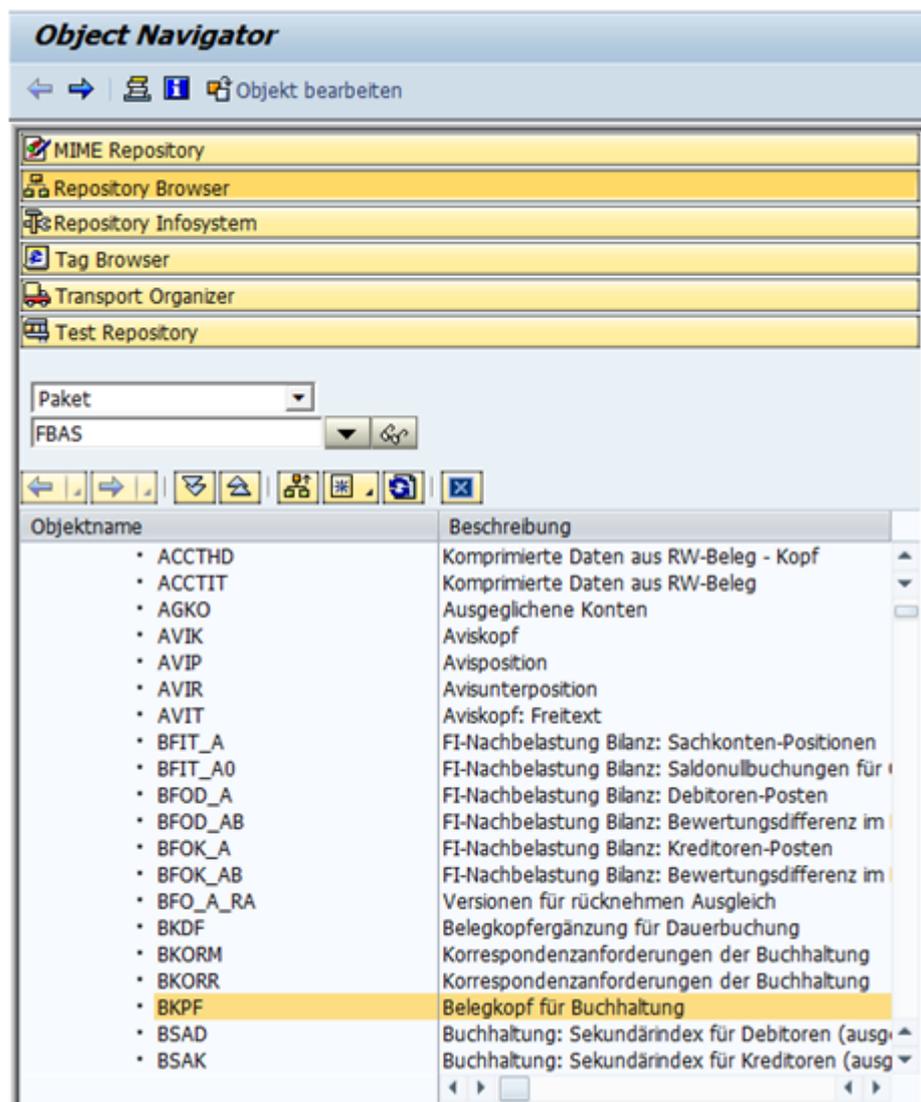


Abbildung C.1: Entwicklungsobjekt Bezahlung in *Object Navigator*

Abbildungsverzeichnis

1.1	Positionierung von Process Mining [AWM11]	2
1.2	Screenshot: SAP-System-Benutzeroberfläche	4
1.3	Drei-Schritte-Methode von Buijs [J.10]	6
1.4	Eine Abbildung in: EVS ModelBuilder [J.10]	7
1.5	Extraktion von Prozessdaten [She05]	8
1.6	EPK-Ausschnitt zu Auftragsabwicklung aus ARIS PPM	9
2.1	Ein Beispiel einer Unternehmensstruktur in SAP [SBc]	13
2.2	Transaktion in SAP ERP	14
2.3	Ereignisdaten in SAP ERP	16
2.4	Ein Ereignismodell nach van der Aalst et al. [AWM11]	17
2.5	Das XES-Metamodell [C.09]	18
2.6	Ein Ausschnitt einer XES-Logdatei	19
2.7	Analyse mit <i>Heuristic Miner</i>	20
2.8	Der Aufbau einer Logdatei angewendet auf SAP ERP	21
2.9	Beispiel:Belegfluss	21
2.10	<i>Easy Access Menu</i>	23
2.11	Struktur eines SAP-Referenzmodells [SB11a]	24
2.12	Wertschöpfungskette des Beschaffungsprozesses	24
2.13	Bestellanforderung und Bestellung	25
2.14	Wertschöpfungskette des Schaden-/Leistungsabwicklungsprozesses	25
2.15	<i>Object Navigator</i>	26
3.1	Projektziel	27
3.2	Tabellenverknüpfung	29
3.3	Fremdschlüsselbeziehung EKPO - EKBE	29
3.4	Dreischichtige SAP-Architektur [M.09]	30
3.5	Ausschnitt aus der RSEG-Tabelle	31
3.6	Der Aufbau der Komponenten in SAP ERP [SB11b]	33
3.7	Zugriffsmöglichkeiten auf BAPIs [SB11b]	34
3.8	SAP JCo-Architektur [SB11b]	35
3.9	ETL-Prozess	36
3.10	Dreischichtige Prototyparchitektur	37
3.11	Vorgehensweise: Identifikation von BAPIs	38
3.12	Extraktion mittels BAPIs	39
3.13	Herstellung eines Belegflusses mit den BAPIs	40

3.14	Belegfluss in den Tabellen in SAP	42
3.15	Einkaufshistorie in SAP	43
3.16	Änderungsarten eines Objektes	44
3.17	Struktur eines Ereignislog in UML-Notation [C.09]	45
3.18	Ein Ausschnitt der generierten XES-Logdatei	46
3.19	Benutzeroberfläche der Anwendung	47
3.20	Verbindungsaufbau mit einem SAP-System	48
3.21	Filter-Tab	49
3.22	Visualisierung der Daten	50
4.1	Beschaffungsprozess in IDES [SBg]	53
4.2	Geschäftsprozess: Bestellanforderung [SBg]	54
4.3	Geschäftsprozess: Bestellung [SBf]	55
4.4	Anforderung erstellen. Bild (a) zeigt das SAP-System-Screenshot. In (b) ist die EBAN-Tabelle dargestellt mit der, von dem Benutzer er- zeugten Anforderung.	57
4.5	Anforderung ändern. Bild (a) zeigt das SAP-System-Screenshot. In (b) sind die CDHDR/CDPOS-Tabellen, mit den von den Benutzern erzeugten Änderungen, dargestellt.	58
4.6	SAP-System-Screenshot: Bestellung anlegen	59
4.7	Screenshot: Bestellung in den EKKO/EKPO-Tabellen	59
4.8	SAP-System-Screenshot: Bestellung ändern	59
4.9	Screenshot: Bestellung in den CDHDR/CDPOS-Tabellen ändern	60
4.10	SAP-System-Screenshot: Wareneingang buchen	60
4.11	EKBE-Tabelle zur Wareneingang	61
4.12	SAP-System-Screenshot: Rechnung erfassen	61
4.13	EKBE-Tabelle zum Rechnungseingang	62
4.14	SAP-System-Screenshot: manuelle Zahlung	62
4.15	Darstellung des Bezahlvorgangs in der BKPF-Tabelle	63
4.16	Gegenüberstellung der gewonnenen Prozessmodelle mit <i>Heuristic Miner</i>	65
4.17	Prozessmodell mit <i>Fuzzy Miner</i>	66
4.18	Prozessmodell mit <i>Fuzzy Miner</i> und zeitlichem Intervall	67
4.19	Prozessmodell mit <i>Genetic Miner</i>	68
4.20	Prozessmodell mit <i>BPMN</i>	69
4.21	XOR-Konnektor in BPMN. Bild (a) zeigt den XOR-Split-Konnektor. In (b) ist der XOR-Join-Konnektor dargestellt.	69
4.22	Prozessmodell mit Alpha-Algorithm-Plugin	70
A.1	Die Rechtecke zeigen die Ereignisse, wobei die Rauten die XOR- Konnektoren darstellen	77
B.1	Antrag auf Leistung	78
B.2	Leistungsfallanlage	79
B.3	Leistungspflichtprüfung	80
B.4	Leistungsermittlung	81
B.5	Leistungsverteilung	81

B.6	Leistungszahlung	82
B.7	Leistungsfallabschluss	82
C.1	Entwicklungsobjekt Bezahlung in <i>Object Navigator</i>	83

Literaturverzeichnis

- [Aal03] AALST, W. M.; Weijters W. d.: Process Mining: A research agenda. In: *Department of Technology Management, Eindhoven University of Technology P.O Box 513, NL-5600 MB, Eindhoven, The Netherlands.* (2003)
- [Aal05] AALST, W. M. d.: A Meta Model for Process Mining Data. In: *In Proceedings of the Caise Workshops (2005)*, S. 309–320
- [Aal07] AALST, W. M.; de Medeiros A. K.; Weijters A. d.: Genetic process mining. In: *Department of Technology Management, Eindhoven University of Technology P.O Box 513, NL-5600 MB, Eindhoven, The Netherlands.* (2007)
- [Aal10] AALST, W. M. d.: XES, XESame, and ProM 6. In: *Technische Universiteit Eindhoven, Department of Mathematics and Computer Science P.O. Box 513, 5600 MB Eindhoven, The Netherlands (Juni 2010)*
- [AWM11] AALST W. M. van d.: *Process Mining: Discovery, Conformance and Enhancement of Business Processes.* Springer, 2011
- [C.09] C., Günther: XES Standard Definition. In: *Fluxicon Process Laboratories (November 2009)*
- [E.06] E., Cook J.: *Process Discovery and Validation through Event Data Analysis*, University of Colorado, Department of Computer Science, Diss., 2006
- [G.00] G., Versteegen: *Projektmanagement mit dem Rational Unified Process.* Springer, 2000
- [GM04] GIESSEL M. van: *Process Mining in SAP R/3*, Eindhoven University of Technology, Diplomarbeit, 2004
- [H.06] H., Forsthuber: *Praxishandbuch SAP-Finanzwesen.* SAP Press, 2006
- [I.07] I., Segers: *Investigating the application of process mining for auditing purposes*, Eindhoven University of Technology, Diplomarbeit, 2007

- [J.07] J., Ingvaldsen J.; G.: Preprocessing Support for Large Scale Process Mining of SAP Transactions. In: *BPM 2007 Workshops, LNCS 4928, S. 30-41* (2007)
- [J.10] J., Buijs: *Mapping Data Sources to XES in a Generic Way*, Eindhoven University of Technology, Diplomarbeit, 2010
- [M.08] M., Maassen A.; Frick D.; Gadatsch A.; S.: *Grundkurs SAP ERP, Geschäftsprozess-orientierte Einführung mit durchgehendem Fallbeispiel*. Vieweg, S. 150-240, 2008
- [M.09] M., Wegelin M.; E.: *SAP-Schnittstellenprogrammierung*. Galileo Press, 2009
- [P.01] P., Wenzel: *Technische Integration von SAP-Systemen, eine Einführung inklusive Customizing, ABAP-4, Accelerated SAP (ASAP), Projektssystem (PS)*. Vieweg, S. 177-248, 2001
- [SAPa] SAP: *Über die SAP AG*,
<http://www.sap.com/corporate-de/our-company/history.epx>,
Abruf: 23.11.2011
- [SAPb] SAP.INFO: *Den ERP-Kern ohne Eingriffe erweitern, Teil 1*,
<http://de.sap.info/den-erp-kern-ohne-eingriffe-erweitern-teil-1/12036>, Abruf: 23.11.2011
- [SBa] SAP-BIBLIOTHEK, (Online): *Allgemeine Einführung in die BAPIs (CA-BFA)*, http://help.sap.com/saphelp_46c/helpdata/de/61/f3f0371bc15d73e10000009b38f8cf/frameset.htm, Abruf: 23.11.2011
- [SBb] SAP-BIBLIOTHEK, (Online): *Business Blueprint*,
http://help.sap.com/saphelp_sm32/helpdata/de/2a/62c33af63ae93ae10000000a11402f/frameset.htm, Abruf: 23.11.2011
- [SBc] SAP-BIBLIOTHEK, (Online): *Corporate Structure*,
http://help.sap.com/saphelp_47x200/helpdata/en/dd/af9fec708b11d194da00a0c930e0da/frameset.htm, Abruf: 23.11.2011
- [SBd] SAP-BIBLIOTHEK, (Online): *Definition des Business-Objekt-Modells*,
http://help.sap.com/saphelp_nw70/helpdata/de/43/1433a0752104c1e10000000a11466f/content.htm, Abruf: 23.11.2011
- [SBe] SAP-BIBLIOTHEK, (Online): *IDES, Das R/3 Modellunternehmen*,
http://help.sap.com/saphelp_46b/helpdata/de/a9/fc4f35dfe82578e10000009b38f839/frameset.htm, Abruf: 23.11.2011
- [SBf] SAP-BIBLIOTHEK, (Online): *Processing Purchase Orders in ERP*,
http://help.sap.com/scenarios_bus2008/helpdata/en/da/203d418f578839e10000000a1550b0/frameset.htm, Abruf: 23.11.2011

- [SBg] SAP-BIBLIOTHEK, (Online): *Processing Purchase Requisition in ERP*, http://help.sap.com/scenarios_bus2008/helpdata/en/da/203d418f578839e10000000a1550b0/frameset.htm, Abruf: 23.11.2011
- [SBh] SAP-BIBLIOTHEK, (Online): *Schaden-/Leistungsabwicklung für Lebensversicherung*, http://help.sap.com/scenarios_bus2008/helpdata/de/48/ab0e78b00d3079e10000000a42189b/frameset.htm, Abruf: 23.11.2011
- [SB11a] SAP-BIBLIOTHEK, (Online): *Modellierungsobjekte des R/3-Referenzmodells*, 2011. http://help.sap.com/saphelp_45b/helpdata/de/d3/56e290415011d189ec0000e81ddfacc/content.htm, Abruf: 23.11.2011
- [SB11b] SAP-BIBLIOTHEK, (Online): *SAP-Bibliothek*, 2011. <http://help.sap.com>, Abruf: 23.11.2011
- [Sch04] SCHEER, IDS: *IDS Scheer stellt Version 3.2 ihrer Process Mining Software ARIS PPM vor*, 2004. <http://de.sap.info/ids-scheer-stellt-version-3-2-ihrer-process-mining-software-aris-ppm-vor/1740>, Abruf: 23.11.2011
- [She05] SHEER, IDES: *Aris Process Performance Manager 4.0*. In: *White Paper* (Oktober 2005)
- [W.98] W., Hagen: *SAP-R-3-Grundlagen für Anwender*. Sybex, 1998
- [WJ09] WERF J., Bozkaya M.; Gabriels J. d.: *Process Diagnostics: a Method Based on Process Mining*. In: *Department of Technology Management, Eindhoven University of Technology P.O* To appear in the proceedings of International Conference on Information, Process, and Knowledge Management (eKNOW) (2009)

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet sowie Zitate kenntlich gemacht habe.

Dortmund, den 18. Dezember 2011

Igor Blyufshteyn

