Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorthesis

Fabian Bergfeld

Einführung und Betrieb des Pentaho Open Source Business Intelligence Systems am Beispiel eines mittelständischen Unternehmens

Fakultät Technik und Informatik
Department Informations- und Elektrotechnik

Faculty of Engineering and Computer Science
Department of Information and Electrical Engineering

Fabian Bergfeld

Einführung und Betrieb des Pentaho Open Source Business Intelligence Systems am Beispiel eines mittelständischen Unternehmens

Bachelorthesis eingereicht im Rahmen der Bachelorprüfung
im Studiengang Angewandte Informatik
am Department Informations- und Elektrotechnik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. Olaf Zukunft
Zweitgutachter : Prof. Dr. Stefan Sarstedt

Abgegeben am 27. Juli 2011

Fabian Bergfeld

Thema der Bachelorthesis

Einführung und Betrieb des Pentaho Open Source Business Intelligence Systems am Beispiel eines mittelständischen Unternehmens

Stichworte

Business Intelligence, Pentaho, Open Source, Reporting, ETL, OLAP

Kurzzusammenfassung

Diese Arbeit beschreibt die Einführung und den Betrieb des Pentaho Open Source Business Intelligence Systems zur Analyse auflaufender Geschäftsdaten zum Zwecke der Optimierung der Erfassung und Visualisierung. Dies geschieht am Beispiel eines mittelständischen Unternehmens.

Fabian Bergfeld

Title of the paper

Launching and operating the Pentaho Open Source Business Intelligence System in a medium-sized business environment

Keywords

Business Intelligence, Pentaho, Open Source, Reporting, ETL, OLAP

Abstract

Inside this report, it is described how to launch and operate the Pentaho open source business intelligence system to analyze various business information to optimize capturing as well as visualization. This takes place in a medium-sized business environment.

Inhaltsverzeichnis

1 Einführung

In heutigen Unternehmensumgebungen laufen ständig eine Vielzahl von Daten und Informationen auf, die gespeichert werden oder werden sollten. Die Anzahl steigt stetig, sowohl im Umfang als auch in der Geschwindigkeit. Das Spektrum reicht von einfachen Daten bis hin zu komplexen Kombinationen.
Diese Daten sind nicht nur zur betriebswirtschaftlichen Analyse des Unternehmens notwendig, sondern beinhalten u.a. komplizierte Informationen, die händisch nicht zu überblicken sind und analysiert werden müssen. Um anfallende Daten aus allen Unternehmensbereichen in einem zentralen System zu sammeln, die enthaltenen Informationen zu extrahieren und den Unternehmenserfolg darzustellen, bietet sich eine Technik an, die sich »Business Intelligence« nennt.
Mit Hilfe dieser Systeme lassen sich große Datenmengen aus annähernd beliebigen Datenquellen sammeln und in z.B. benutzerdefinierten Berichten dynamisch analysieren, um zielgerichtete Informationen aus ihnen zu gewinnen. Das Ziel der Business Intelligence ist, mit den gewonnen Informationen betriebswirtschaftliche Entscheidungen zu verbessern.
Diese Beschreibung zeigt, dass der Namensbestandteil »Intelligence« nicht wirklich den Begriff der Intelligenz meint. Unter Intelligenz versteht Wikipedia (Wikipedia, 2011d) :

> In der Psychologie ist Intelligenz ein Sammelbegriff für die kognitive Leistungsfähigkeit des Menschen, also die Fähigkeit, zu verstehen, zu abstrahieren, Probleme zu lösen, Wissen anzuwenden und Sprache zu verwenden.

Der Begriff bezeichnet also zusammengefasst die Leistungsfähigkeit, Zusammenhänge zu erfassen, diese herzustellen und zu verstehen.
Das Wort »Intelligence« im Zusammenhang der Business Intelligence erfordert ein anderes Verständnis, bei dem ein Hinweis auf die CIA (Central Intelligence Agency[1]) hilfreich sein kann. In diesem Fall beschreibt »Intelligence« das Sammeln und Auswerten von Informationsquellen, die häufig sehr umfangreich sein können. Diese verschiedenen Interpretationen des Begriffs zu berücksichtigen, hilft im Umgang mit der Thematik der Business Intelligence.

[1] Auslandsgeheimdienst der USA

1.1 Motivation

Für mittelständische Unternehmen ist es nicht ungewöhnlich, dass, wenn das Geschäft mit den Jahren wächst, die Struktur zur Erfassung und Erhebung von Unternehmensdaten nicht in gleicher Geschwindigkeit mitwächst. Konnte ein Unternehmen mit wenigen Beschäftigten und einem überschaubaren Kunden- oder Produkt-Stamm vor einigen Jahren noch Daten in Excel-Dateien speichern und trotzdem den Überblick behalten bzw. war ein Interesse an Anruf-Statistiken nicht gegeben, ist dies bei heutigen Wachstumsbedingungen so nicht mehr möglich und erfordert Neuerungen.

Möchte die Geschäftsführung Analysen der bestehenden Daten vorgelegt bekommen, die nur das Übertragen von auf Papier geführten Listen in eine vorbereitete Tabelle beinhalten, kann es je nach Anzahl der Listen zu einem regelmäßigen Zeitaufwand von mehreren Stunden kommen. Sollen diese Analysen wöchentlich durchgeführt werden, ist zu sehen, dass ein mit dieser Aufgabe betrauter Mitarbeiter auf einen Monat gesehen einen vollen Arbeitstag damit beschäftigt ist, diese Daten zu übertragen, selbst wenn er pro Woche nur zwei Stunden benötigt.

Durch den Einsatz von Business Intelligence Systemen ist es möglich, eine Analyseinfrastruktur aufzubauen, die mit überschaubaren Anpassungen in der Lage ist, nahezu beliebige Daten auszuwerten und auf verschiedene Arten darzustellen.

Das Ziel dieser Arbeit ist es, zu untersuchen, ob in einem vertretbaren Zeitfenster zu geringen Kosten ein solches System aufzubauen ist. Dieses soll mit tatsächlich anfallenden Daten eines ausgewählten Bereichs arbeiten und diese auf verständliche und den Anforderungen entsprechende Art aufbereiten und darstellen.

1.2 Die Gliederung

Diese Arbeit ist aufgeteilt in folgende Abschnitte, die hier kurz beschrieben werden.

Das zweite Kapitel widmet sich der Einführung in das Themengebiet, beinhaltet also die Erklärung von Grundbegriffen und Zusammenhängen. Neben der Erklärung, was unter »Business Intelligence « zu verstehen ist und warum Pentaho das System der Wahl ist, wird ebenfalls auf den Begriff »Open Source« eingegangen.

Kapitel drei widmet sich der Analyse und klärt die Fragen, welche Anforderungen an das Zielsystem bestehen und welche Probleme es zu lösen gilt.

Die Entwicklung eines lauffähigen Systems ist Thema des vierten Kapitels, das sich mit der Architektur, der Entwicklung von Tools zur Datenerfassung und schließlich der Verarbeitung der Daten durch das Business Intelligence Systems befasst.

Mit dem fünften Kapitel wird diese Arbeit nach einer Zusammenfassung, einem Fazit und einem Ausblick auf zukünftige Entwicklungen und Erweiterungen geschlossen.

2 Grundlagen

In diesem Kapitel werden benötigte Begriffe, Techniken und Zusammenhänge eingeführt, die eine Rolle in dieser Arbeit spielen. In Abschnitt 2.1 findet sich eine Erläuterung des Begriffs »Business Intelligence «, im darauf folgenden Abschnitt 2.2 eine Beschreibung der Eigenschaften von »Open Source«.

2.1 Business Intelligence

Business Intelligence beschreibt analytische Prozesse, die sowohl die Bereitstellung quantitativer und qualitativer Daten als auch die Aufdeckung relevanter Zusammenhänge und die Kommunikation der gewonnen Erkenntnisse zur Entscheidungsunterstützung umfassen (Köster, 2002).

Anders ausgedrückt werden unter Business Intelligence gebräuchlicher Weise analytische Konzepte und Werkzeuge zusammengefasst, deren Ziel es ist, vorhandene Daten dahingehend zu analysieren, aus ihnen neue Erkenntnisse zu gewinnen.

2.1.1 Zielsetzung

Das Hauptziel der Business Intelligence liegt in der zielgerichteten Aufbereitung von Daten und Informationen, um betriebswirtschaftliche Entscheidungen zu verbessern und zu erleichtern (Engels, 2010). Je früher es den Entscheidern von Unternehmen möglich ist, Informationen zu Trends, kritischen Entwicklungen oder auch vergleichsweise Trivialem wie dem Schrumpfen von Beständen eines Produkts oder Bauteils im Lager zu erhalten, desto schneller kann reagiert werden. Negative Auswirkungen auf den Betrieb können so gering gehalten werden oder im Idealfall gar nicht erst aufkommen.

Die Business Intelligence ist ein weitläufiges Feld, für das es zwar immer gleichbleibende dazuzuzählende Komponenten gibt, das jedoch aufgrund der Vielfältigkeit der Verständnismöglichkeiten schwierig abgegrenzt werden kann. Eine solche Abgrenzung bzw. Aufteilung in Kategorien stellt Mertens auf und fasst damit das Verständnis von Business Intelligence in sieben Kategorien zusammen (Mertens, 2002):

1. Business Intelligence als Fortsetzung der Daten- und Informationsverarbeitung: IV für die Unternehmensleitung
2. Business Intelligence als Filter für die Informationsflut: Informationslogistik
3. Business Intelligence = MIS[1], mit besonders schnellen/flexiblen Auswertungen
4. Business Intelligence als Frühwarnsystem (»Alerting«)
5. Business Intelligence = Data Warehouse
6. Business Intelligence als Informations- und Wissensspeicherung
7. Business Intelligence als Prozess: Symptomerhebung → Diagnose → Therapie → Prognose → Therapiekontrolle

Bei Betrachtung der in der Aufzählung verwendeten Begriffe zeigt sich eine Abgrenzung, die in der Hauptsache durch verwendete Bausteine erreicht wird. Es werden Daten und Informationen verarbeitet (vgl. 1.), gefiltert (vgl. 2.), ausgewertet (vgl. 3.), auf vorher festgesetzte Grenzwerte geprüft (vgl. 4.), gespeichert (vgl. 5. und 6.) und in einen Zusammenhang gebracht (vgl. 7.). Die Kernaufgaben eines Business Intelligence System sind es, Daten firmeninterner und externer Art so zusammenzufügen, dass es dadurch dem Management des Betriebs erleichtert wird, die richtigen Entscheidungen zu treffen.

2.1.2 Funktionsweise

Aus verschiedenen Datenquellen, die z.B. aus relationalen Datenbanksystemen, Log-Dateien, Dateien aus Tabellenkalkulationsprogrammen oder Textdateien bestehen können, die aber selbst noch nicht zum Business Intelligence System dazuzuzählen sind, werden die Daten eingelesen und weiterverarbeitet. Sind die zu verwendenden Daten nicht ausdrücklich auf die Verwendung in dem Business Intelligence System zugeschnitten, werden für diesen ersten Schritt häufig »ETL[2]«-Programme verwendet, auf die im Abschnitt 2.1.2.1 noch genauer eingegangen wird.

Im nächsten Schritt werden die gelesenen Daten in einem »Data Warehouse« gespeichert und können dort zur Analyse oder Darstellung durch das Business Intelligence System aufbereitet werden. Bei der Datenanalyse werden häufig drei Analyseverfahren unterschieden:

- Data Access (s. Abschnitt 2.1.2.2)

[1] Management Informations System
[2] Extract, Transform, Load

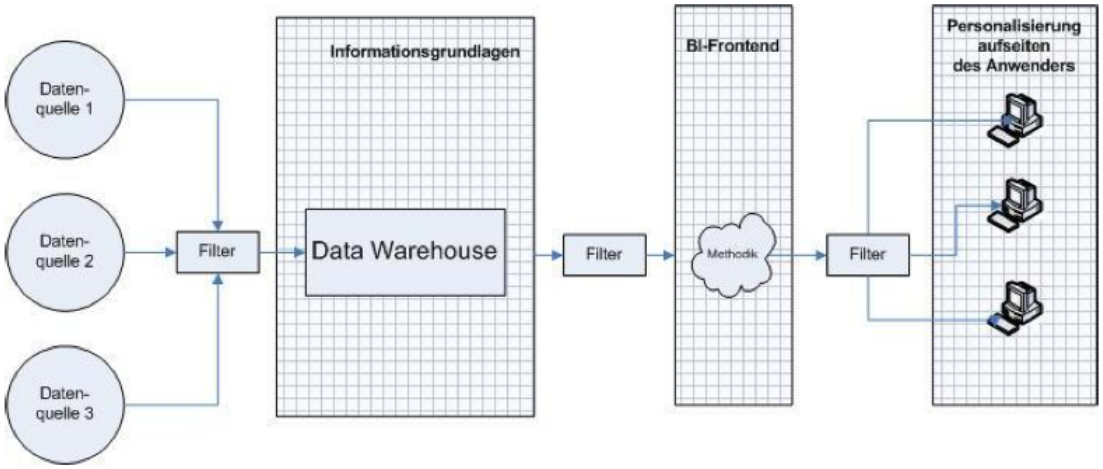

Abbildung 2.1: Typische Architektur von Business Intelligence Systemen (Reibold, 2010)

- OLAP[3] (s. Abschnitt 2.1.2.3)
- Data Mining (s. Abschnitt 2.1.2.4)

Durch diese Verfahren können Daten effizient und zielgerichtet dem Anwender dargestellt werden.

2.1.2.1 ETL

Ausgangsdaten aus verschiedenen Datenquellen liegen nur in Ausnahmefällen in Formaten vor, in denen sie sofort zur weiteren Verwendung in Business Intelligence Systemen zu verwenden sind. Nicht nur, wenn man z.B. händisch gepflegte Adressdaten verwenden möchte, ist die Wahrscheinlichkeit, dass diese nicht konsistente Schreibweisen beinhalten, ziemlich groß. Allein bei der Schreibweise der Straße sind die Variationsmöglichkeiten mit »...str«, »...str.«, »...strasse« und »...straße« nur die Spitze des Eisbergs. Dieses und weitere Probleme zu lösen, ist die Aufgabe von ETL-Programmen:
Quelldaten werden gelesen (Extract) und über konfigurierbare Filter auf Konsistenz geprüft, Doubletten können entfernt oder Fehler berichtigt werden (Transform). Auch können einzelne Spalten oder Werte ausgelesen werden, so dass es möglich ist, sich mit Hilfe von ETL-Programmen genau die benötigten Daten in genau der benötigten Form in die Zieldatenbank zu laden (Load).

[3] Online Analytical Processing

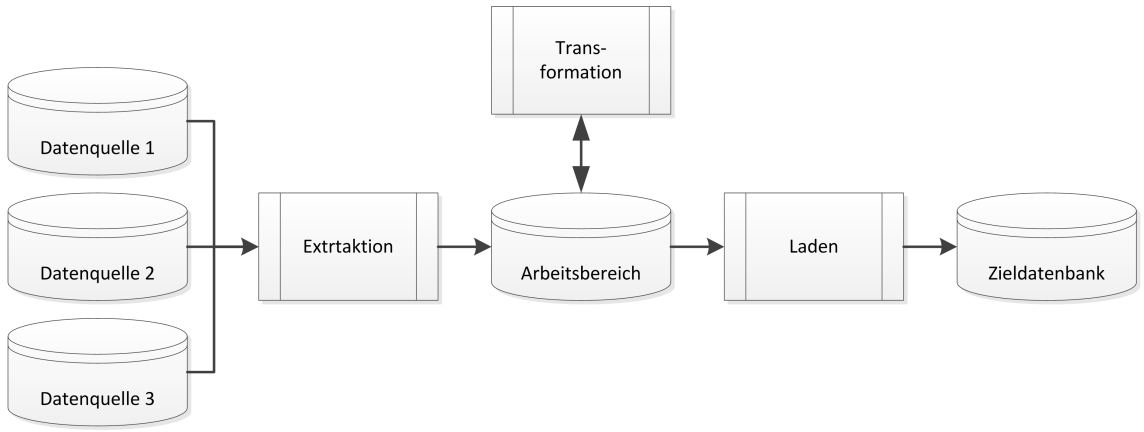

Abbildung 2.2: Abläufe beim ETL-Prozess - nach (Wikipedia, 2011b)

2.1.2.2 Data Access

Der Data Access sieht keine weitere Analyse der Daten vor, sondern beschränkt sich auf das Auslesen. Die Ergebnisse werden meist in recht statischen Reports dargestellt. »Recht statisch« deshalb, weil es möglich ist, generelle Attribute festzulegen, auf deren Basis der Bericht automatisch generiert wird. Dies können beispielsweise Zeiträume oder die Vorgabe einer Kategorie sein. Ist der Bericht jedoch einmal ausgegeben, besteht ohne eine Neugenerierung keine Möglichkeit, auf diesen Einfluss zu nehmen. Gleiches gilt im Wesentlichen für ein ebenfalls häufig genutztes Tool, das Dashboard, welches den Zweck hat, einen schnellen Überblick über relevante Daten zu ermöglichen. Verbreitet sind in diesem Zusammenhang auch Ampelfunktionen, bei denen Grenzwerte vorgegeben werden, so dass man durch ein Wechseln der Farbe von grün über gelb auf rot einen noch schnelleren Überblick über den Lagerbestand erhalten kann.

2.1.2.3 OLAP

OLAP ist ein bestimmtes multidimensionales Datenmodell zuzuordnen, das als Weiterentwicklung von Tabellenkalkulationsprogrammen aufgefasst werden kann (Heuer und Saake, 2000). Dieses Datenmodell wird häufig mit dem Begriff Datenwürfel bezeichnet, um es anschaulicher zu machen. Diese Technik dient dazu, große Datenmengen in Echtzeit zu analysieren. Im Gegensatz zu OLTP[4], bei dem während typischer Operationen Daten nicht nur dargestellt, sondern auch verändert werden, arbeitet OLAP ausschließlich lesend. Da häufig sehr große Datenmengen zu analysieren sind, ist es wichig, dass das System performant auf Benutzeranfragen reagiert. Daher werden die Daten meist in einem Data Warehouse

[4]Online Transaction Processing

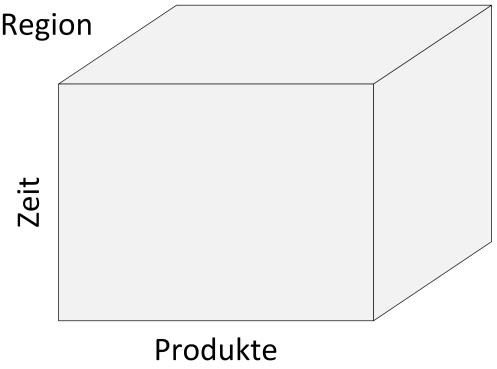

Abbildung 2.3: Multidimensionaler Datenwürfel

vorgehalten und nicht aus dem laufenden Firmenumfeld bezogen (Hyde, 2006).
Wie in Abb. 2.3 beispielhaft gezeigt, werden die Daten nach verschiedenen Dimensionen[5] organisiert. Die Abfragen an ein solches System werden u.a. durch Pivotierung[6] gelöst, indem z.B. die Spaltenreihenfolge oder der Ausschnitt verändert werden.

OLAP lassen sich folgende Standard-Operationen zuordnen (Manhart, 2008):

- Pivotierung: Nicht immer ist der volle Informationsgehalt eines Hypercubes notwendig, um die benötigte betriebswirtschaftliche Analyse fahren zu können. Häufig reicht auch ein zweidimensionaler Ausschnitt aus. Unter der »Pivotierung« versteht man das Verdrehen des Würfels um die eigene Achse, so dass eine andere Kombination aus zwei Achsen sichtbar wird.

- Roll-Up und Drill-Down: Sind Dimensionen zusätzlich noch in Hierarchien unterteilt (z.B. Lebensmittel $\rightarrow$ Obst, Gemüse), kann man durch die Operationen Roll-Up und Drill-Down innerhalb dieser Hierarchien navigieren.
  Beim Drill-Down steigt man von einem Darstellungsniveau auf die nächsttiefere und damit detailliertere Stufe.
  Der Roll-Up ist die dazu inverse Operation, die die Einzelwerte auf die nächsthöher gelegene Hierarchiestufe zusammenfasst und dadurch den Detailgrad verringert.

- Slice und Dice: Hierbei handelt es sich um Operationen, die der Selektion einzelner Daten dienen, also eine Filterfunktion ausüben.
  Beim Slice wird beim Anwenden auf einen dreidimensionalen Würfel eine Scheibe herausgeschnitten, so dass eine zweidimensionale Matrix entsteht, mit der weitergearbeitet werden kann. Generell ist die Anzahl der Dimensionen beim Ergebniswürfel

[5] Meist handelt es sich in der Praxis um einen Hypercube (Hyperwürfel), der über mehr als drei Dimensionen verfügt.

[6] Französisch: Drehachse. In Tabellenumgebungen häufig verwendet, um Daten anders darzustellen, ohne die Ursprungstabelle zu verändern (von Cube, 2003).

geringer als bei der Quelle.
Beim Dice wird ein mehrdimensionaler und detaillierterer Ausschnitt eines Hypercubes geliefert. Die Anzahl der Dimensionen des Ursprungswürfels bleiben erhalten, jedoch werden durch Anwendung von Filtern die dargestellten Elemente reguliert.

2.1.2.4 Data Mining

Als Data Mining[7] werden Analyseverfahren bezeichnet, deren Ziel es ist, Muster in einem Datenbestand zu erkennen. Änderungen im Verhalten von Kunden, die sich durch den Einsatz von solchen Techniken erkennen lassen, können helfen, den Betrieb daran anzupassen. Der Data-Mining-Prozess wird meist in vier Phasen aufgeteilt (Reibold, 2010):

1. Datenselektion: Dieser Punkt wird wiederum in zwei Schritten durchgeführt. Zuerst werden die relevanten Datensätze bestimmt und die relevanten Attribute festgelegt. Ist der optimale Aggregationsgrad zu diesem Zeitpunkt noch unbekannt, sollte der niedrigste verfügbare Wert ausgewählt werden (Wikipedia, 2011a). Sind die zu untersuchenden Daten gefunden, müssen diese für den weiteren Gebrauch aufbereitet werden. Bestenfalls sind sie konsistent in einem Data-Warehouse gespeichert, was in der Praxis bedauerlicherweise nicht die Regel ist, da sie dort häufig über historisch gewachsene Systeme und Datenbanken verstreut liegen. Häufig werden für den operativen Betrieb aktuell nicht benötigte Daten gar nicht in solchen Systemen gespeichert — daher ist es sinnvoll, weitere Datenquellen auf Relevanz zu prüfen, ob z.B. Daten in Papierform abgelegt sind, oder ob es relevante Daten gibt, die individuell auf Rechnern von Mitarbeitern in Dateien von Textverarbeitungs- oder Tabellenkalkulations-Programmen abgelegt sind. Ist das der Fall, gilt es zu prüfen, auf welche Weise sich die betreffenden Daten mit einbeziehen lassen.

2. Datenaufbereitung: In der Literatur wird dem Autor Peter Cabena die Aussage zugeschrieben, lediglich zehn Prozent des Zeitaufwandes im Data-Mining seien unmittelbar auf den Einsatz von Data-Mining-Methoden zurückzuführen, während 90 Prozent für Datenaufbereitung und Ergebnisbereinigung in Anspruch genommen würden.
Diese Aussage zeigt, wie wichtig und anspruchsvoll es ist, Daten per Bereinigung in Formate, Strukturen und Standards zu bringen, die maschinell weiterverarbeitet werden können. In der Praxis kommt es häufig zu semantischen Problemen, die durch unterschiedliche Schreibweisen oder Synonyme entstehen. Werden z.B. Adressdaten verarbeitet, kommt kommt es sehr oft zu Ungenauigkeiten: »...str.«, »...str«, »...straße« oder »...strasse«. Diese Inkonsistenzen müssen auf jeden Fall bereinigt werden.
Ein weiteres Beispiel für ein Phänomen, das häufig auftritt, sind redundant gespeicherte Datensätze, die es zu finden und die Doubletten zu löschen gilt.

[7] engl. für Daten schürfen

3. Mustererkennung: Diese Phase des Data-Mining-Prozesses lässt sich wieder in drei Unterphasen unterteilen:

   a) Modellspezifikation

   b) Modellevaluation

   c) Suche

   Die Modellspezifikation hat die Auswahl des Analyseverfahrens zum Ziel. Es ist darauf zu achten, diejenigen Verfahren zu finden und zu verwenden, die geeignet sind, den vorliegenden Problemtyp zu bearbeiten.
   Durch Modellevaluation wird geprüft, inwieweit ein entdecktes Muster den gesetzten Anforderungskriterien entspricht.
   Die Suche gibt dem Benutzer die Möglichkeit, selber mit dem System zu arbeiten.

4. Kommunikation: In dieser Phase werden die gefundenen Muster in eine weiterverwertbare Form gebracht. Da es sich meist um Computersysteme handelt, werden die Inhalte in eine formale Sprache überführt, die vom Zielsystem weiterverarbeitet werden kann.

2.2 Open Source

Der Begriff »Open Source« bezeichnet in erster Linie, dass der Quellcode der Software offen liegt und von jedermann einzusehen, zu verändern und weiterzugeben ist. Mittlerweile gibt es Abwandlungen dieser Vorgehensweise, so dass nicht nur Software, sondern auch andere Produkte als »Open Source« bezeichnet werden, wie z.B. Rezepte für Erfrischungsgetränke[8] oder das Konzept zum effizienten Bau eines Rechenzentrums, wie es Facebook im April 2011 veröffentlicht hat. [9] Diese Entwicklung ist zwar interessant, jedoch beschränkt sich diese Arbeit auf das traditionelle Verständnis von Open Source.

2.2.1 Die Geschichte von Open Source

Die Geschichte quelloffener Software ist älter als das Label »Open Source«. Im Jahr 1997 veröffentlichte Eric Steven Raymond das Essay »Die Kathedrale und der Basar[10]«. Beeinflusst davon entschied Netscape ein Jahr später, den Quellcode seines kommerziell nicht

[8] http://www.open-cola.de/

[9] http://www.heise.de/open/meldung/Open-Compute-Project-Facebook-veroeffentlicht-Spezifikationen-fuer-effiziente-Rechenzentren-1224492.html

[10] Mittlerweile aktualisierte Fassung unter: http://www.catb.org/ esr/writings/cathedral-bazaar/cathedral-bazaar/

mehr erfolgreichen Browsers Netscape Navigator freizugeben, woraus später das Mozilla-Projekt entstand. Diese Freigabe war für viele Nutzer ein unglaublicher Schachzug. Offene Software war bisher für den Normalverbraucher völlig unbekannt.
Im Februar des gleichen Jahres wurde die OSI[11] durch Eric Raymond und Bruce Perens gegründet, die das Label »Open Source« verwalten, vermarkten und weniger ideologisch, sondern seriöser erscheinen lassen sollte. (OSI, 1998)

Wie bereits erwähnt, gibt es quelloffene Software schon länger als deren heute gebräuchliche Bezeichnung. Im Jahr 1969 entwickelten Ken Thompson und Dennis Ritchie die erste Version von UNIX in den AT&T Bell Telephone Laboratories. Eine kommerzielle Verwertung war AT&T damals nicht möglich, so dass Lizenzen der Software gegen eine geringe Gebühr ohne Anspruch auf Support oder Bugfixes Universitäten zugänglich gemacht wurden. Da mit Fehlerbehebungen oder Erweiterung an eigene Bedürfnisse nicht zu rechnen war, begann eine rege Entwicklungsarbeit, die das Usenet als Support-Netzwerk verwendete. Koordiniert wurde diese Entwicklung in der Hauptsache durch die University of Berkley, die auch selber entwickelte und UNIX unter dem Namen Berkley Software Distibution (BSD) vertrieb.
Anfang der 1980er wurde es AT&T möglich, im Softwarebereich tätig zu sein und UNIX wurde durch eine Anhebung der Lizenzkosten und Schließen des Quellcodes wieder verschlossen. Zu dieser Zeit wurde von Richard Stallman das GNU[12]-Projekt gegründet, dessen Ziel es war, ein UNIX-ähnliches, aber völlig freies Betriebssystem zu entwickeln.
Ende der 1980er bis Anfang der 1990er Jahre ermöglichten erste Internetprovider auch Personen außerhalb von Universitäten den Zugang zu Netzwerken wie dem Internet, so dass die Entwicklerzahlen von Open Source-Projekten steigen konnten. Etwa zur gleichen Zeit arbeitete Linus Torvald mit dem Betriebssystem Minix, welches die Basis zu Linux darstellt. Er rief die Netznutzer zur Mitarbeit am Quellcode auf, so dass nach kurzer Zeit nicht nur der Code von Minix vollständig ersetzt war, sondern auch weitere heute gebräuchliche Stücke Software entstanden, wie z.B. Apache und Samba.
Nun fand die bereits beschriebene Überführung des Netscape Navigators in die Sphäre der offenen Software statt. Ihm folgte die Ankündigung Corels, die Textverarbeitung Wordperfect nach Linux zu portieren und IBM kündigte Unterstützung und Vertrieb von Apache an. (OSI, 1998) (O'Reilly und Associates, 1999) (LWN, 1998)

Der Einsatz von Open Source Software ist eine ernstzunehmende Alternative zu kommerziell vertriebener Software. Durch die Offenheit des Quellcodes können individuelle Anpassungen an Funktion der Software vorgenommen werden. Steht eine große Anzahl Entwickler hinter dem Projekt, ist die Wahrscheinlichkeit groß, dass Sicherheitslücken nach ihrem Bekanntwerden u.U. schneller geschlossen werden, als dies bei kommerzieller Software der Fall sein würde. Das Vertrauen, dass die Software den eigenen Sicherheitsansprüchen genügt und über keine Hintertüren verfügt, die unautorisierten Personen Zugriff verschaffen könn-

[11] Open Source Initiative
[12] Rekursives Akronym: GNU's Not Unix

ten, kann ebenfalls steigen, wenn jeder die Möglichkeit hat, das Programm zu analysieren. Nicht zuletzt ist der Einsatz interessant, weil für die Software keine Lizenzkosten anfallen.
Es ist jedoch nicht so, dass Open Source Software nur in Kellern und Garagen entwickelt wird. Einige der großen Software-Projekte werden von bedeutenden börsennotierten Firmen vorangetrieben. Die IDE[13] Eclipse hat ihre Wurzeln bei IBM, die Office Suite Open Office gehört zu Oracle. Die Firmen machen mit Open Source Software nicht zwangsläufig keinen Profit oder sogar ein Verlustgeschäft. Durch die freie Verfügbarkeit wird eine hohe Verbreitung erreicht, so dass die Benutzer später mit Support- oder Wartungsverträgen zu ihnen zurückkehren, wodurch der teils immense Entwicklungsaufwand finanziert wird.

2.2.2 Definition

Damit eine Software den Begriff »Open Source« führen darf, reicht es nicht, dass ihr Quellcode einsehbar ist, es müssen folgende »10 Gebote« der Open Source Definition erfüllt sein (OSI, 1998) [Übersetzung von (O'Reilly und Associates, 1999)]:

1. »Freie Weiterverbreitung: Die Lizenz darf niemanden im Verkauf oder in der Weitergabe der Software als Teil einer aus verschiedenen Quellen zusammengesetzten Softwaredistribution einschränken. Die Lizenz darf keinerlei Lizenzgebühren oder andersartige Beiträge verlangen.«

2. »Quellcode: Das Programm muss den Quellcode beinhalten und sowohl die Verbreitung als Quellcode als auch in kompilierter Form gestatten. Wird ein Teil des Produkts nicht mit Quellcode verbreitet, so muss auf eine Möglichkeit, den Quellcode gebührenfrei aus dem Internet downzuloaden, ausdrücklich hingewiesen werden. Der Quellcode muss in einer Form zur Verfügung gestellt werden, in der ein Programmierer ihn verändern kann. Ein absichtlich verwirrend geschriebener Quellcode ist nicht erlaubt. Ebenso sind Zwischenformen, wie die Ausgabe eines Präprozessors oder eines Übersetzers, verboten.«

3. »Auf dem Programm basierende Werke: Die Lizenz muss die Veränderung des Programms, auf dem Programm basierende Werke sowie deren Verbreitung unter den gleichen Lizenzbedingungen gestatten.«

4. »Die Unversehrtheit des Originalcodes: Die Lizenz darf die Verbreitung von modifiziertem Quellcode nur dann einschränken, wenn sie die Verbreitung von sogenannten Patchdateien in Verbindung mit dem Originalcode gestattet, damit das Programm vor der Benutzung verändert werden kann. Die Lizenz muss ausdrücklich die Verbreitung von Software erlauben, die mit verändertem Quellcode erstellt wurde. Die Lizenz darf

[13] Integrated Development Environment

allerdings von auf dem Programm basierenden Werken verlangen, einen von der Originalsoftware verschiedenen Namen oder eine andere Versionsnummer zu tragen.«

5. »Keine Diskriminierung von einzelnen Personen oder Gruppen: Die Lizenz darf keinerlei Personen oder Personengruppen diskriminieren.«
6. »Keine Einschränkungen für bestimmte Anwendungsbereiche: Die Lizenz darf niemanden in der Benutzung des Programms in einem bestimmten Einsatzgebiet einschränken. Sie darf beispielsweise nicht die kommerzielle Nutzung oder die Benutzung in der Genforschung verbieten.«
7. »Verbreitung der Lizenz: Die zum Programm gehörigen Rechte müssen für jeden gelten, der das Programm erhalten hat, ohne dass eine weitere Lizenz beachtet werden muss.«
8. »Die Lizenz darf nicht für ein bestimmtes Produkt gelten: Die zum Programm gehörigen Rechte dürfen nicht davon abhängen, dass das Programm Teil einer bestimmten Softwaredistribution ist. Wird das Programm außerhalb einer solchen Distribution genutzt oder verbreitet, so gelten für den Benutzer dieselben Rechte, die in der Originaldistribution gewährt werden.«
9. »Die Lizenz darf andere Software nicht beeinträchtigen: Die Lizenz darf keine andere Software einschränken, die zusammen mit der lizensierten Software verbreitet wird. Die Lizenz darf beispielsweise nicht verlangen, dass jegliche Software, die auf demselben Datenträger verbreitet wird, Open Source-Software sein muss.«
10. Die Lizenz muss technologie-neutral sein: Keine Bestimmung der Lizenz darf eine Technologie oder eine Schnittstelle ausschließen.

(OSI, 1998)

»Open Source« ist als Fundament zu verstehen, auf dem Software entwickelt wird, das vorsieht, dass der Quelltext jedem für jeglichen Einsatzzweck unter Beachtung der aufgeführten zehn Punkte zur Verfügung steht.

2.2.3 Open Source Lizenzen

Open Source Software unterliegt jedoch nicht keinerlei Bestimmungen. Auch in diesem Bereich gibt es Lizenzen, die festlegen, was mit der Software geschehen darf. Bis heute wurden von der OSI fast 70 Lizenzen als konform zu den gestellten Regeln anerkannt.[14] Diese lassen sich am besten in drei Kategorien einteilen (Kleijn, 2006b):

[14] http://www.opensource.org/licenses/alphabetical

1. Starkes Copyleft
2. Schwaches Copyleft
3. Kein Copyleft

Der Begriff »Copyleft« erscheint zunächst befremdlich, kennt man doch meist nur den des Copyrights, bei dem ausschließlich der Urheber bzw. der Rechteinhaber einer Software über die Verbreitung bestimmt. Er wurde maßgeblich von Richard Stallmann mit absichtlicher Anlehnung an den Copyright-Begriff geprägt und besagt, dass Änderungen und Weiterentwicklungen einer Open Source-Software nur unter der gleichen Lizenz als weiterhin freie Software weitergegeben werden dürfen.

Art der Lizenz	Starkes Copyleft	Schwaches Copyleft	Kein Copyleft
Kombinationsmöglichkeit mit proprietärer Software	Keine Einbindung in proprietären Code möglich	Statisches und dynamisches Linken von Code mit proprietärer Software möglich. Eigen-Entwicklungen dürfen als proprietäre Software weitergegeben werden	Keine Vorgaben. Der gesamte Code darf auch als proprietäre Software weitergegeben werden
Beispiel-Lizenz	GPL	LGPL, MPL	BSD, Apache

Tabelle 2.1: Open Source Lizenz-Kategorien (nach (Kleijn, 2006b))

Die obige Tabelle zeigt für GPL[15] an, dass die Einbindung in proprietären Code ausgeschlossen ist. Dies kann im Firmenumfeld unangenehme Auswirkungen haben. Innerhalb eines Softwareprojektes ist es nicht ungewöhnlich, auf vorhandene Bibliotheken zurückzugreifen. Ebenfalls nicht ungewöhnlich ist, dass Firmen häufig kein besonders stark ausgeprägtes Interesse daran haben, selbstentwickelte und u.U. hochspezialisierte Betriebssoftware öffentlich zugänglich zu machen. Sie ist - obwohl Open Source - in der Handhabung im Firmenumfeld unter bestimmten Voraussetzungen schwierig. Die Lizenz erfordert jedoch, dass selbst Software, die nur unter GPL stehende Bibliotheken aufruft, diese Lizenz erbt. Aufgrund dieses Problems gibt es noch die abgeschwächte Variante LGPL[16], die diesen Zugriff ausdrücklich erlaubt und keine Copyleft-Forderungen an die aufrufende Software stellt (Wikipedia, 2011c).

[15] General Public License
[16] GNU Lesser General Public License

Bei einigen Softwareprodukten gibt es eine weitere Vorgehensweise: das »Dual-Licensing«. Bei diesem Modell wird eine Software wahlweise unter zwei Lizenzen angeboten. So kann eine kommerzielle Lizenz mit einer offenen kombiniert werden, so dass der Anwender entscheiden kann, welche Version er einsetzen möchte. Der grundsätzliche Funktionsumfang unterscheidet sich i.d.R. zwischen beiden Versionen nicht. Zwei Gründe sprechen für die Variante mit der kommerziellen Lizenz: Support und Zertifizierung (Diedrich, 2006). Beides Gründe, die im Unternehmensumfeld für den Einsatz der kommerziellen Version sprechen, jedoch für Privatanwender meist irrelevant sind, so dass diese das Produkt trotzdem weiterhin kostenfrei einsetzen können. Ein weiterer Punkt, der natürlich von den verwendeten Lizenzen abhängt, ist die Kopplung mit eigenen kommerziellen Anwendungen. Wie bereits beschrieben, ist dies durch die GPL nicht möglich, ohne dass der Zwang entsteht, den eigenen Code offenlegen zu müssen. Dies ist bei der kommerziellen Version von der Datenbanksoftware MySQL, die unter Dual-Lizenz angeboten wird, beispielsweise möglich. (Diedrich, 2006)

2.3 Softwarelösungen im Bereich Business Intelligence

Business Intelligence Software ist über Jahre hinweg die Domäne großer Firmen gewesen, wie SAP, Oracle, Cognos[17], Microsoft und vielen mehr. Diese hatten die finanziellen Möglichkeiten und die Infrastruktur, solch komplexe Software zu entwickeln. Die Open Source-Gemeinde setzt bei der Entwicklung freier Alternativen bestehende Komponenten ein, wie beispielsweise als Datenspeicher die Datenbanken MySQL oder PostgreSQL. Im Laufe der Jahre wurden verschiedene in der Business Intelligence gebräuchliche Komponenten einzeln entwickelt:
Für den ETL-Prozess gibt es die Tools Kettle, als OLAP-Server Mondrian, als Reporting-Engine gibt es JFreeReport und für den Data-Mining-Prozess kann das Weka-Projekt verwendet werden. Diese Komponenten stellen kombiniert im Wesentlichen die Pentaho Business Intelligence Suite dar.
Pentaho ist nicht die einzige Anlaufstelle für Open Source Business Intelligence Lösungen. Jedoch ist die Pentaho-Suite eine der wenigen Lösungen, die viele Komponenten der Business Intelligence unter einem Dach vereint und als Komplettlösung zur Verfügung stellt. Eine Lösung, die ähnlich wie Pentaho aufgestellt ist, findet sich in der Business Intelligence Lösung von Jaspersoft[18]. Neben der kommerziellen Version, die Support seitens des Herstellers beinhaltet, gibt es eine freie Sammlung der Tools, die in der JasperForge-Community[19] angeboten wird.

[17] 2008 von IBM übernommen
[18] http://www.jaspersoft.com/
[19] http://jasperforge.org/

Daneben überwiegen Tools, die auf Teilaspekte spezialisiert sind.
Um ETL-Aufgaben zu lösen, reicht die Spanne von sehr einfachen Tools wie z.B. Scriptella[20] oder Octopus[21] zu sehr mächtigen wie CloverETL[22] oder Kettle (Manageability.org, 2009). Auch einzelne OLAP-Server stehen quellenoffen zur Verfügung. Die Bekanntesten sind das mittlerweile in der Pentaho-Suite integrierte Mondrian-Projekt und die Datenbanklösung für mehr Übersicht in Excel Palo[23] vom Hersteller Jedox[24] aus Freiburg. Das Weka-Projekt, das von der Universität aus Waikato in Neuseeland zur Verfügung gestellt wird, bietet eine Sammlung von Algorithmen zum Data-Mining-Prozess.
Der Artikel »Business Intelligence mit Open Source« von Alexandra Kleijn (Kleijn, 2006a) stellt eine Reihe von Lösungen vor und ermöglicht so einen ersten Überblick über Open Source Lösungen innerhalb der Business Intelligence Landschaft.

Software	Suite	ETL	OLAP	Reporting	Lizenz
Pentaho CE	•	•	•	•	GPLv2
Jaspersoft CE	•	•	•	•	GPL
Palo Suite CE	•	•	•	•	GPL
JFree Report				•	LGPLv2.1
Modrian			•		EPLv1
Kettle		•			LGPLv2.1
Scriptella		•			APL
Octopus		•			GPL
Cloveretl		•			LGPL

Tabelle 2.2: Übersicht einiger Open Source Business Intelligence Software

Für die Umsetzung des Projekts fällt die Wahl der Software auf die Pentaho Business Intelligence-Suite, weshalb im Folgenden auch näher auf diese eingegangen wird. Auch wenn andere Lösungen technisch ebenfalls den Anforderungen genügen, verspricht Pentaho CE eine umfangreichere Dokumentation als es bei Jaspersoft CE der Fall ist (Pientka, 2008).

[20] http://scriptella.javaforge.com/
[21] http://octopus.objectweb.org/
[22] http://www.cloveretl.com/
[23] http://www.palo.net/de/
[24] http://www.jedox.com/

2.4 Pentaho

Die Pentaho Business Intelligence Suite ist eine von der Pentaho Corporation[25] seit 2004 existente Zusammenstellung aus Softwarelösungen der üblichen Bereiche ETL, OLAP, Reporting und Data Mining, die vollständig in Java entwickelt ist. Sie wurde 2007 von Infoworld zu den zehn wichtigsten Business-Lösungen im Open Source Bereich gezählt.[26]

Durch die Kombination der wichtigsten Tools, die im modularen Aufbau der Suite ineinandergreifen, ist Pentaho eine ernstzunehmende Konkurrenz gegenüber kommerziellen Lösungen etablierter Hersteller. Deren größte Nachteile sind sowohl die Lizenzkosten — die gerade für kleinere Unternehmen einen tiefen Einschnitt ins Budget bedeuten können — als auch die eigene Geschlossenheit. So ist eine individuelle Anpassung an eigene Firmensysteme i.d.R. nicht selbst vorzunehmen, sondern muss — wenn möglich — kostspielig beim Hersteller dazugekauft werden. Pentaho vereint die für die meisten Business Intelligence Anforderungen geeignete Software in einer Suite, aus der die einzelnen Komponenten zu wählen sind, die für den eigenen Einsatzzweck benötigt werden. Diese sind vollständig anpassbar und erweiterbar und werden von dem Business Intelligence Server als gemeinsame Plattform gesteuert.
Die Suite ist sowohl in einer freien Version[27], als auch in einer kommerziellen »Enterprise«-Version erhältlich. Im Funktionsumfang unterscheiden sich beide Versionen nicht, jedoch bietet die Enterprise-Version nicht nur einen durch Pentaho geleisteten Support, sondern es sind auch weitere Softwarekomponenten enthalten, die das Aufsetzen und Einrichten der Suite erleichtern.

Die Pentaho Business Intelligence Suite besteht aus folgenden Komponenten (Held und Klose, 2007):

- Pentaho BI-Server: Diese stellt die gemeinsame Kontrollinstanz dar und ist eine Eigenentwicklung Pentahos. Enthalten sind Web-Frontends für die Administration und die Nutzung der weiteren Komponenten.
- Pentaho Reporting: Basierend auf dem Projekt JFreeReport lassen sich automatisch Berichte auf Basis von durch das System oder anderweitig erhobenen Daten erstellen.
- Pentaho Data Integration: Auf Basis des Projekts Kettle beinhaltet dies eine leistungsfähige ETL-Lösung
- Pentaho Analysis: Zu diesem Oberbegriff gehören einzelne Komponenten für OLAP auf Basis von Mondrian und für Data Mining auf Basis Wekas

[25] http://www.pentaho.com/
[26] http://www.computerwoche.de/software/software-infrastruktur/1873958/index9.html
[27] http://community.pentaho.com/

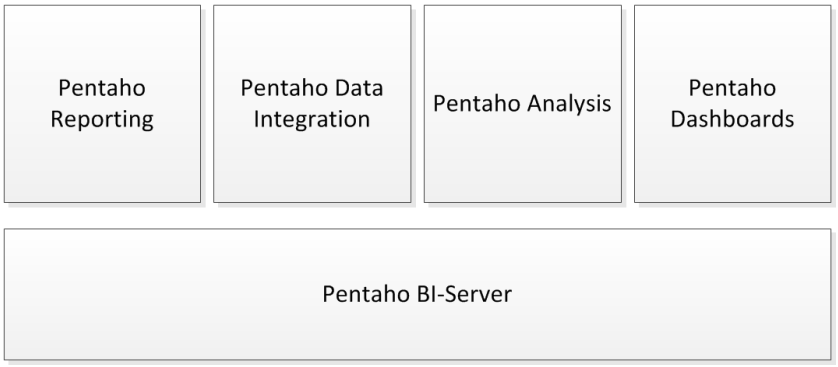

Abbildung 2.4: Modularer Aufbau der Pentaho Suite (nach (Held und Klose, 2007))

- Pentaho Dashboards: Dienen als Übersichtsseite, auf der statisch aktuelle Informationen angezeigt werden.

Die Solution Engine ist das Herzstück der Architektur, die die einzelnen Komponenten der Suite zusammenhält und die Operationen prozessorientiert steuert. Business Intelligence Prozesse werden durch Action Sequences in XML dargestellt, in denen die durchzuführenden Aktionen definiert sind. Eine einfache Action Sequence kann das Auslesen einer festgelegten Datenquelle mit anschließender Verarbeitung und Speicherung der Daten im Data Warehouse sein. Eine solche Aufgabe ist bei Bedarf innerhalb des Systems mit Hilfe des Schedulers so einzurichten, dass sie automatisch in vorbestimmten Zeitintervallen ausgeführt wird und das Ergebnis beispielsweise per Email an die zuständige Person geschickt wird. Diese Sequences sind sehr mächtig und können um unzählige Attribute erweitert werden, wie z.B. dass vor dem Abruf von Informationen aus einer Datenquelle erst geprüft wird, ob diese überhaupt existiert und ob diese seit dem letzten Durchlauf verändert wurde, so dass auch sehr komplexe Aufgaben in Action Sequences abgebildet werden können.

Die in der Pentaho Suite verwendeten Technologien sind sehr offen. So finden JSP, HTML, SOAP, CSS, AJAX, J2EE, JDBC u.a. Verwendung. Diese Offenheit führt in Kombination mit dem konsequent modularen Aufbau dazu, dass es möglich ist, beliebige Komponenten der Suite gegen andere auszutauschen. Statt Mondrian kann eine beliebige MDX-kompatible OLAP-Komponente verwendet werden oder es kann statt Kettle JasperETL[28] eingesetzt werden. Die Kehrseite dieser Flexibilität ist eine eingeschränkte Übersicht. So gibt es kein wirklich zentrales und allumfassendes Admin-Tool, stattdessen wirken Konfiguration und Module etwas fragmentiert.

Der Pentaho Business Intelligence Server setzt einen J2EE-fähigen Webserver voraus. Es gibt zwei Versionen zum Download. Eine, die vorkonfiguriert ist und ihren eigenen bereits

[28] http://jasperforge.org/projects/jasperetl

eingerichteten Application-Server mitbringt oder eine, die zur komplett manuellen Einrichtung bestimmt ist.

3 Analyse

Dieses Kapitel befasst sich mit den Anforderungen an das System und Möglichkeiten zur Umsetzung (Abschnitt 3.1). In den Abschnitten 3.2 und 3.3 werden zunächst die Zielsetzung des zu entwickelnden Systems beschrieben, sowie dessen formale Anforderungen definiert.

3.1 Anwendungsszenarien

Dieser Abschnitt stellt dar, wie Business Intelligence-Systeme die Informationsgewinnung innerhalb eines Unternehmens unterstützen können, das sowohl telefonischen, als auch persönlichen Kundenkontakt hat und Verträge über Telekommunikations- und anderen Dienstleistungen mit diesen hat.

3.1.1 Auswertung von Telefon- und Besucherstatistiken

Für ein Unternehmen ist die Möglichkeit der Planung einer der wichtigsten Aspekte überhaupt. Es gilt, relevante Faktoren möglichst präzise vorhersagen und z.B. präzise beantworten zu können, welche Anzahl an Kunden zu welcher Uhrzeit zu erwarten ist. Auf Grundlage dieser Werte ist es möglich, die Anzahl der Mitarbeiter, die voraussichtlich für die jeweiligen Abteilungen erforderlich sind, zu optimieren. Zusätzlich lässt sich durch Veränderungen dieser Daten ablesen, ob eine Werbekampagne oder ein Angebot, das zu diesem Zeitpunkt geschaltet oder kommuniziert wurde, zu einem Anstieg der Kundenkontakte führt.

3.1.2 Auswertung von Lagerbeständen

Die meisten Unternehmen, die über ein Lager für die zu verkaufenden Produkte verfügen, werden Daten darüber speichern, wie viele Produkte durch Lieferungen ihren Weg in das Lager hineinfinden. Auch der Verkauf wird i.d.R. durch irgendeine Art von Kassen- oder Abrechnungssystem protokolliert. Möchte man nun möglichst in Echtzeit einen Überblick über verfügbare Mengen eines oder mehrerer Produkte im Lager haben und ist das Lager nicht

bloß ein kleines direkt einsehbares Regal, setzt dies voraus, dass beide Daten verknüpft werden.

3.1.3 Weitere mögliche Anwendungen

Die Anzahl möglicher Einsatzgebiete von Business Intelligence-Systemen ist nahezu unzählbar. Sei es, dass Statistiken darüber erstellt werden sollen, wie viele Kunden in einem gewissen Zeitraum einen Vertrag abgeschlossen, geändert oder gekündigt haben oder ob die Frage Klärung bedarf, an welchen Tagen am meisten Kaffee getrunken wird. Es gilt die Regel, dass alle Informationen, die irgendwie irgendwo gespeichert werden, ausgewertet werden können.

3.2 Zielsetzung

Das Ziel dieser Arbeit ist es, ein System zu entwickeln, das im ersten Schritt in der Lage ist, die in 3.1.1 beschriebenen Daten sowohl zu erheben als auch auszuwerten. Aus den Telefon- und Besucherdaten sollen getrennt verschiedenen Berichte (Reports) generiert werden, die eine tagesgenaue, wöchentliche, monatliche, quartalsweise und jährliche Auswertung ermöglichen.
Darüber hinaus soll das Business Intelligence-System komplett und funktionstüchtig aufgesetzt werden, so dass für später folgende Erweiterungen bereits eine Infrastruktur geschaffen ist, die erweitert und weiterverwendet werden kann.

3.3 Anforderungen

Im Rahmen dieser Arbeit wird ein System entwickelt, das die Zielsetzung umsetzt und zeigen soll, dass es einem mittelständischen Unternehmen möglich ist, eine solche Software unter beschränktem Einsatz von Zeit und Geld einzuführen und den Informationsstand anzuheben.

Im Folgenden werden die funktionalen Anforderungen an das System getrennt nach System- und Benutzeranforderungen beschrieben, anschließend sodann die nichtfunktionalen Anforderungen.
Benutzeranforderungen beschreiben die Funktion aus Sicht des Endanwenders, also des Mitarbeiters, der das System entweder mit Daten versorgt oder die vom System generierten Informationen abruft. Die Systemanforderungen beinhalten eine genaue Beschreibung

der Beschaffenheit des Systems aus Sicht des Entwicklers oder des Administrators. Zuletzt werden nichtfunktionale Anforderungen nach DIN 66272 geklärt.

3.3.1 Benutzeranforderungen

Die Abb. 3.1 beschreibt anhand eines Zustandsdiagramms den typischen Ablauf der Speicherung eines Datensatzes innerhalb des Systems. Zur Speicherung ist es erforderlich, dass dem Mitarbeiter ein beliebiger Webbrowser zur Verfügung steht, der über LAN[1] mit dem Business Intelligence-Server verbunden ist (R1)[2]. Über den Browser wird ein kleines webbasiertes Tool aufgerufen, das dem Mitarbeiter verschiedene Stichpunkte anbietet, warum sich ein Kunde meldet (R2). Nachdem das Anliegen des Kunden, dessen Kontaktaufnahme persönlich im Servicecenter oder telefonisch erfolgen kann, bearbeitet ist, wird vom Mitarbeiter der Kundenkontakt passend durch einen Klick auf den entsprechenden Eintrag kategorisiert (R3). Ist dies geschehen, hat er auf der darauffolgenden Seite, die ihm die Speicherung bestätigt, die Möglichkeit, den Eintrag zu löschen und nochmal eine andere Kontaktursache auszuwählen (R4). Wird auf »Zurück« geklickt, ist der Datensatz gespeichert und das Tool steht für die Speicherung des nächsten Datensatzes zur Verfügung.

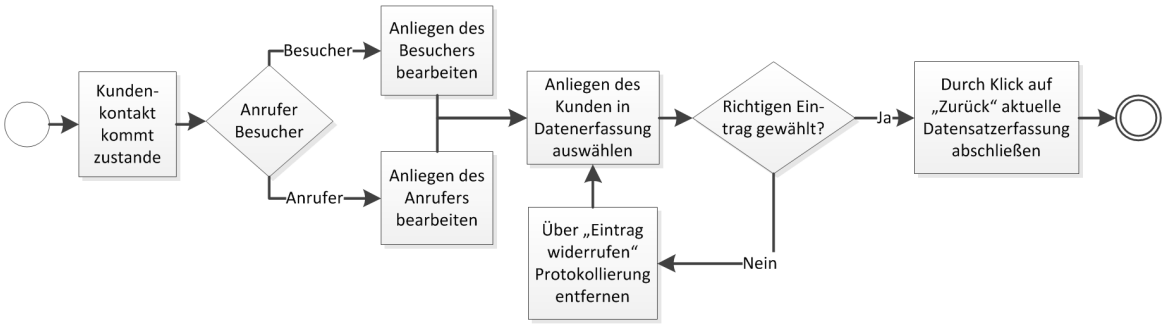

Abbildung 3.1: Zustandsdiagramm aus Anwendersicht

Der Mitarbeiter, der für die Generierung der Berichte auf Basis der über das Erfassungstool erhobenen Daten zuständig ist, meldet sich am Business Intelligence-Server mit seinen individuellen Zugangsdaten an (R5), wählt aus den verfügbaren Berichten (Tages-, Wochen-, Monats-, Quartals- und Jahres-Bericht) den gewünschten aus (R6), gibt den zu bearbeitenden Zeitraum an (R7), wählt, ob Besucher oder Anrufer ausgewertet werden soll (R8) und legt das Ausgabeformat des Berichtes (PDF oder HTML) (R9) fest. Der angeforderte Bericht wird generiert und dem Mitarbeiter angezeigt, der ihn speichern und ggf. ausdrucken kann (R10).

[1] Local Area Network
[2] Durchnummerierte Anforderungen (Request)

Die gespeicherten Daten bleiben im System bestehen, so dass Berichte beliebig oft erzeugt werden können (R11).

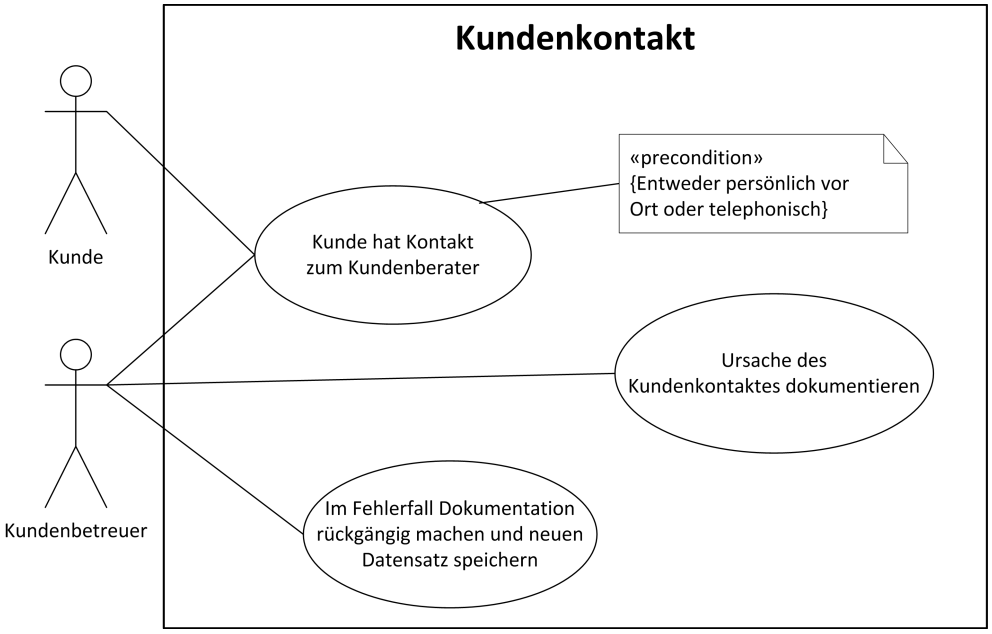

Abbildung 3.2: Anwendungsfalldiagramm: Kundenkontakt

3.3.2 Systemanforderungen

Das System besteht in erster Iterationsstufe aus zwei Hauptkomponenten:

- Die Business Intelligence-Suite
- Ein eigenes webbasiertes Tool zur Datenerfassung der Telefon- und Besucherdaten

Das Tool zur Datenerfassung wird physikalisch von dem gleichen Rechner, der auch die Business Intelligence-Software beheimatet, durch den dort aufgesetzten Apache Webserver zur Verfügung gestellt und nutzt eine MySQL-Datenbank als Speicher. Es handelt sich dabei um eine PHP-Webapplikation, die aus ihrer Datenbank die zur Verfügung stehenden Kontaktursachen lädt, diese graphisch aufbereitet und in einem übersichtlichen Layout anzeigt. Wird ein Eintrag per Klick ausgewählt, wird automatisch ein Eintrag in der Datenbank erstellt. Dieser beinhaltet folgende Daten:

- Id: Einzigartige Id als Bezeichner dieses Eintrags
- Datum

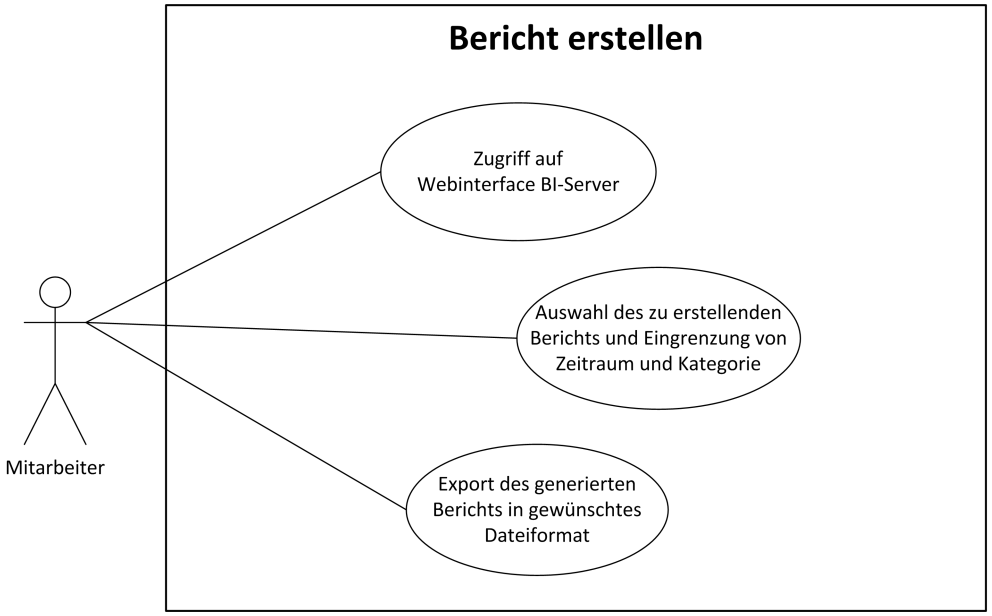

Abbildung 3.3: Anwendungsfalldiagramm: Bericht erstellen

- Uhrzeit
- IP: IP-Adresse des auslösenden Mitarbeiters, um auswerten zu können wie viele Mitarbeiter beteiligt waren.
- Ursachen-Id: Id der gewählten Ursache

Das Tool stellt keine Anforderung an Personalisierbarkeit seitens des Anwenders. Schriftgröße und Farben werden so gewählt, dass der Kontrast so hoch wie möglich ist und dass alle Informationen zu jeder Zeit ohne die Notwendigkeit, den Bildschirminhalt zu scrollen, auf eine Bildschirmseite passen. Dadurch wird ein einheitliches Erscheinungsbild erreicht und vermieden, dass es bei individuellen Einstellungen zu »Unfällen« kommt, die die Funktion beeinträchtigen.

Die zweite Komponente - das Business Intelligence-System - benötigt für die Auswertungen Zugriff auf die Datenbank, in der das Erfassungstool die Datensätze speichert. Schreibrechte sind nicht erforderlich, da die Daten nur lesend verarbeitet werden.
Für die Erstellung von Berichten werden entsprechende Berichtsvorlagen durch den Report Designer auf dem Server entwickelt, mit den benötigten Zugriffsrechten versehen und in das System eingepflegt, so dass diese für den dafür zuständigen Mitarbeiter ausführbar sind. Ihm wird die Möglichkeit gegeben, den Bericht über folgende einstellbare Attribute seinen Anforderungen anzupassen:

- Anfangsdatum der Auswertung: Bei der tageweisen Analyse wird jeder Tag, zu dem Daten gespeichert sind, angeboten. Beim Wochenbericht ist es immer der Montag, beim Monatsbericht der Monatserste usw.
- Enddatum: Nur bei der tageweisen Analyse. Ansonsten ist der Zeitraum automatisch eine Woche, ein Monat, ein Quartal oder ein Jahr.
- Kategorie: Auswahlmöglichkeit, ob Besucher oder Anrufer analysiert werden sollen. Die Auswahl, beide Kategorien zu mischen, ist nicht vorgesehen.
- Weitere Eingrenzung des Ziels: Da auch Kunden für Kooperationspartner anrufen oder das Servicecenter besuchen, ist es wünschenswert, die Kundenkontakte, die durch diese entstehen, gesondert ausgeben zu können.

3.3.3 Nichtfunktionale Anforderungen

Nichtfunktionale Anforderungen beschreiben keine sichtbaren Funktionen, sind aber genauso wichtig für den erfolgreichen Betrieb einer Software wie es die funktionalen Anforderungen sind. Sie beschreiben Eigenschaften des Systems global, ohne sich auf Details der Komponenten zu beschränken. Dadurch werden die Rahmenbedingungen des Systems, unter denen es arbeiten muss, definiert.
In diesem Abschnitt werden die nichtfunktionalen Anforderungen in Anlehnung an die in der DIN 66272 beschriebenen Hauptqualitätsmerkmalen erfasst.

Zuverlässigkeit

Das Qualitätsmerkmal »Zuverlässigkeit« bestimmt die Fähigkeit eines Systems, seine Aufgaben auf gleichbleibendem Niveau über ein bestimmtes Zeitintervall auszuführen (RN1)[3].

Zuverlässigkeit ist für das angestrebte System eine sehr wichtige Eigenschaft. Da es sich um keinen Prototypen handelt, der seine Leistungsfähigkeit zu Messzwecken im Labor zur Schau stellen soll, sondern das System sich im Produktiveinsatz behaupten soll, ist es wichtig, dass der ordnungsgemäße Betrieb gewährleistet ist.

Benutzbarkeit und Aussehen

Dieses Merkmal beschreibt die optische Aufbereitung der Software und die Schwierigkeit, die Funktionen der angestrebten Benutzergruppe zur Verfügung zu stellen. Es geht um Verständlichkeit, Erlernbarkeit und Bedienbarkeit (RN2).

[3] Nichtfunktionale Anforderung

Dieses Qualitätsmerkmal muss zweigeteilt gesehen werden, das heißt getrennt bezogen auf das Erfassungstool und das Business Intelligence-System.
Für das Erfassungssystem hat die Benutzereffizienz Vorrang gegenüber der graphischen Aufbereitung. Um zu erreichen, dass bei möglichst vielen Kundenkontakten ein Datensatz erstellt wird, ist es wichtig, dass die Auswahl intuitiv, schnell und problemlos vonstatten geht. Eine konstante Anordnung der Auswahlelemente sorgt dafür, dass die Positionen der Punkte nach einer nicht zu vermeidenden, aber möglichst kurz zu haltenden Lernphase im Gedächtnis bleiben und nicht jedes Mal langwierig der passende Punkt gesucht werden muss.
Aussehen und Benutzbarkeit des Business Intelligence-Systems sind zum großen Teil durch die zugrundeliegende Pentaho-Software vorgegeben. Eine Anpassung wäre natürlich möglich, jedoch ist nicht abzusehen, dass dieser Schritt notwendig ist. Die Oberfläche zum Erstellen der Berichte wird nur von wenigen Mitarbeitern benötigt, die an der sich bislang über Jahre bewährten Oberfläche nichts auszusetzen hatten.

Leistung und Effizienz

Unter diesem Merkmal sind Punkte wie Antwortzeiten, Ressourcenbedarf und Wirtschaftlichkeit zu verstehen (RN3).

Antwortzeiten sind für die Akzeptanz des Erfassungstools sehr wichtig. Benötigt das System mehrere Sekunden, um einen Datensatz zu speichern, würde der Anteil der kategorisierten Kundenkontakte aller Voraussicht nach sinken. Insofern muss das Anlegen eines Datensatzes so verzögerungsfrei wie möglich ablaufen. Weitaus weniger relevant ist die Laufzeit, die das System zur Generierung von den teils sehr umfassenden Berichten benötigt. Natürlich ist es erforderlich, die auftretenden Wartezeiten im Rahmen zu halten, aber Verzögerungen von einigen Sekunden sind hinnehmbar. Durch Änderung der verwendeten Hardware – auf die im nächsten Kapitel eingegangen wird – ist es natürlich möglich, bestehende Verzögerungen zu verringern. Bei diesem Merkmal gilt es jedoch, einen Kompromiss zu finden, der die Antwortzeiten im Rahmen hält und bei dem der Ressourcenbedarf die Grenzen der einzusetzenden Hardware nicht übersteigt, so dass die Wirtschaftlichkeit gegeben bleibt.

Wart- und Änderbarkeit

Der Aufwand, der betrieben werden muss, um ein System zu warten oder Änderungen an diesem durchzuführen, wird durch dieses Qualitätsmerkmal bezeichnet (RN4).

Sehr sinnvoll ist es, bei der Entwicklung von Software auf dieses Qualitätsmerkmal zu achten. Es gibt verschiedene Ansatzpunkte, die diese Eigenschaften begünstigen. Beispielsweise kann durch die Verwendung von Design Patterns[4] eine hohe Abgrenzung von

[4] Entwurfsmuster

Teilen der Software erreicht werden, so dass jede Komponente eine definierte Aufgabe hat, diese wiederverwendbar und bei Bedarf austauschbar ist.
Bei der Entwicklung des Erfassungstools ist es wichtig, gleich miteinzubeziehen, dass sich Kategorien ändern können, einige wegfallen oder neue dazukommen werden. Wenn es dafür notwendig ist, an mehreren Stellen Quellcode zu verändern, ist dieses kaum effizient möglich. Daher ist es sinnvoll, die Darstellung der Auswahlpunkte so dynamisch wie möglich zu machen, also dafür zu sorgen, dass so viel wie möglich direkt aus Datenbankvorgaben generiert wird.

Portier- und Übertragbarkeit

Dieses Qualitätsmerkmal ist ein Indikator für den Aufwand, das System von einer Software- oder Hardwareumgebung in eine andere zu übertragen. (RN5)

Da das Erfassungstool eine reine Webapplikation ist, die sowohl unter Linux, OS X, Windows als auch mit mobilen Endgeräten verwendet werden kann, ist dieses Qualitätsmerkmal für das Erfassungstool gegeben. Dieses gilt auch für die Pentaho-Software, die ebenfalls sowohl für Linux, Windows und OS X zur Verfügung steht.

4 Design und Realisierung

Dieses Kapitel beschreibt das Vorgehen, um das zu entwickelnde Zielsystem den Anforderungen an dieses anzupassen. Zunächst beinhaltet der Abschnitt 4.1 Architektur, Hard- und Software-Ausstattung des Zielsystems. Der Abschnitt 4.2 beschreibt die Entwicklung des Tools zur Datenerfassung, Abschnitt 4.3 geht auf die Analyse der erhobenen Daten ein, wobei ein Einblick in die Erstellung von Reports und auch das beispielhafte Einlesen von Daten mittels Kettle gewonnen wird. Im letzten Abschnitt des Kapitels wird ein erstes Fazit gezogen.

4.1 Architektur des Zielsystems

Sowohl die Auswertung der erhobenen Daten als auch die Erhebung der Daten selbst soll webbasiert über einen zentralen Server gesteuert vonstatten gehen, so dass bei der Entwicklung des Zielsystems nur in dem Maße auf Eigenschaften der Client-Rechner zu achten ist, dass die Systeme in Verbindung mit den clientseitig verwendeten Webbrowsern und Bildschrimauflösungen lauffähig und benutzbar sind.

Zur Strukturierung der Funktionalität wird eine Schichtenarchitektur verwendet. Eine solche hat mehrere Vorteile:
Durch die geringe Kopplung der Schichten wird eine hohe Kohäsion erreicht. Dies führt zusammen mit der geringen Komplexität der Abhängigkeit dazu, dass Wiederverwendbarkeit der einzelnen Module — z.B. in anderen Projekte — und Austauschbarkeit gegen andere Komponenten vereinfacht werden.

Die zur Anwendung kommenden Schichten werden im Folgenden erläutert, um einen Überblick des Zielsystems zu verschaffen (s. Abb. 4.1).

Datenhaltung (1)

Diese Schicht dient der Datenhaltung. Die auszuwertenden Daten werden hier über das Erfassungs-Tool gespeichert und zur späteren Auswertung gelesen.

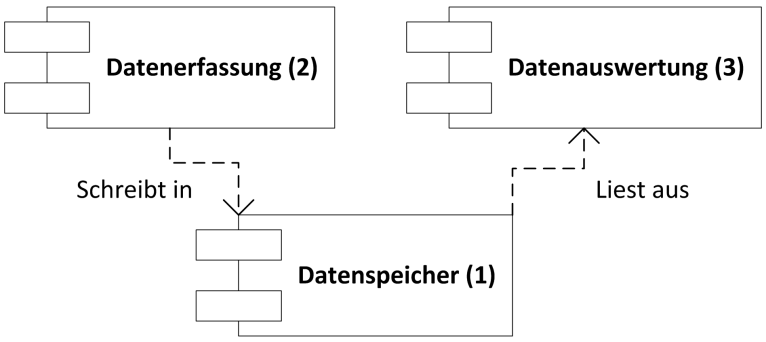

Abbildung 4.1: Komponentendiagramm der Schichtenarchitektur des Zielsystems

Datenerfassung (2)

Hier findet die Erfassung von Daten statt. Über ein geeignetes Frontend wird dem Benutzer die Möglichkeit gegeben, die auflaufenden Kundenkontakte kategorisiert zusammen mit weiteren Informationen wie einem Zeitstempel und der eigenen IP-Adresse an Schicht 1 zur dortigen Speicherung zu übergeben.

Datenauswertung (3)

Die durch Schicht 2 in Schicht 1 gespeicherten Daten werden in der Auswertungsschicht eingelesen und über das Pentaho Web-Frontend dem zuständigen Mitarbeiter zur weiteren Bearbeitung zur Verfügung gestellt. Dieser hat die Möglichkeit, verschiedene Berichte aus den Daten generieren zu lassen, wobei sowohl der Zeitraum der zur Auswertung berücksichtigten Daten als auch die Kategorie und das Ausgabeformat wählbar ist.

4.1.1 Verwendete Hardware

Für die Umsetzung des Projekts steht ein eigener Server zur Verfügung, der alle Elemente der Software beheimaten soll. Die Anforderungen an diesen sind überschaubar.
Es handelt sich um ein Gerät mit folgenden Eigenschaften:

- Prozessor: Intel Pentium 4 @ 1,8 GHz
- RAM: 1,28 GB
- HDD: 20 GB
- LAN: 10/100 MBit/sek

Weitere Eigenschaften der Hardware sind für den späteren Betrieb weitestgehend irrelevant. Soll auf dem Server auch Entwicklungsarbeit stattfinden, ist es sinnvoll, mehr Arbeitsspeicher einzusetzen, um ausreichend Spielraum zur Verfügung zu haben. Im laufenden Betrieb ist aber ein Wert um 1 GB ausreichend.

4.1.2 Verwendete Software

Die benötigte Software selbst steht sowohl für Windows, als auch für Mac OS X und Linux zur Verfügung. Da kein Mac verfügbar ist, der Einsatz von Windows nur Lizenzkosten mit sich brächte und es logisch erscheint, bei der Einführung eines Systems auf Open Source Basis auch ein solches Betriebssystem einzusetzen, wurde ein solches auch ausgewählt. Es ist problemlos möglich, das System auch im Nachhinein in eine andere Umgebung zu portieren (RN5).
Folgende Komponenten werden verwendet:

- Betriebssystem: Linux, Distibution Ubuntu 10.10 »Maverick Meerkat«[1]
- Datenbank: MySQL 5[2]
  - phpMyAdmin[3]
  - MySQL GUI Tools[4]
- Webserver: Apache[5] mit PHP 5[6]
- Texteditor mit Syntaxhighlighting: TextWrangler[7]
- Business Intelligence-Server: Pentaho Business Intelligence Server and Suite 3.6[8]
- Pentaho Report Designer 3.6.1[9]
- Pentaho Data Integration (Kettle) 4.0.1[10]

All diese Software ist entweder Open Source oder kostenfrei nutzbar, so dass für das System keinerlei Kosten für Softwarelizenzen entstehen.

[1] http://www.ubuntu.com/
[2] http://www.mysql.com/
[3] http://www.phpmyadmin.net/home_page/index.php
[4] http://dev.mysql.com/downloads/gui-tools/5.0.html
[5] http://www.apache.org/
[6] http://php.net/
[7] http://www.barebones.com/products/textwrangler/
[8] http://community.pentaho.com/
[9] http://reporting.pentaho.com/
[10] http://kettle.pentaho.com/

4.2 Datenbeschaffung

Um Daten auszuwerten, müssen diese in einem ersten Schritt ins System eingelesen oder – falls es bisher keine Erfassung gibt – erhoben werden. Da es in der ersten Phase der Einführung darum geht, Daten über Anrufer und Besucher des Servicecenters zu sammeln, ist als Erstes zu untersuchen, ob solche Daten bisher gespeichert werden und — falls ja — ob diese für den späteren Verlauf zu berücksichtigen sind.
Im konkreten Fall wurden zwar über viele Monate Daten über Besucher- und Anrufvorkommen erhoben und ausgewertet, jedoch geschah dies per Hand auf Formularblättern, deren Inhalte zur Auswertung wiederum per Hand in eine Excel-Tabelle übertragen wurden. Diese Daten zu übernehmen, wäre technisch zwar möglich, jedoch sprechen zwei Gründe dagegen, diesen Aufwand zu treiben:

1. Aus den bisherigen Daten wurden bereits elektronische Berichte erstellt, die zur Auswertung benötigt wurden, weshalb eine erneute Aufbereitung nicht notwendig ist.
2. Aufgrund der neuerlich möglichen elektronischen Erhebung von Daten ist eine weit größere Aufsplittung der zu unterscheidenden Kategorien erwünscht. Die bisherige Kategorisierung ist also nicht deckungsgleich mit der kommenden.

Daher ist in diesem Fall keine Übernahme der bestehenden Daten notwendig und es kann mit einem frischen Bestand begonnen werden.

4.2.1 Entwicklung des Logging-Tools

Das Logging-Tool ist eine kleine Webapplikation, die von dem auf dem aufgesetzten Server bereitgestellten Apache Web-Server zur Verfügung gestellt wird. Die MySQL-Datenbank-Zugriffe erfolgen über PHP von einem beliebigen Rechner, der über Zugriff auf das Firmennetzwerk verfügt (R1).

4.2.1.1 Das Datenbankschema

Bevor die Entwicklung dieses Tools beginnen kann, ist die Erstellung eines Datenbankmodells erforderlich, in welchem die zu speichernden Daten abgebildet werden können. Dafür sind bereits zwei Tabellen ausreichend.

Tabelle »reasons«

Diese Tabelle beinhaltet neben der ID, die Primärschlüssel ist, die Informationen über die zu speichernden Kategorien (R2).

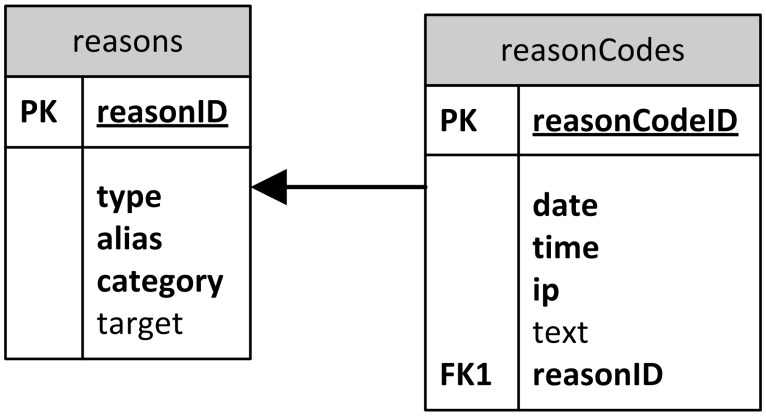

Abbildung 4.2: Datenbankschema des Logging-Tools

Die Spalte »type« speichert eine datenbankinterne Bezeichnung der Kategorie, wie beispielsweise »willy_kaufReceiverHD«.
Die Spalte »alias« beinhaltet eine aussagekräftige Beschreibung. Im Fall des verkauften HD-Receivers ist diese »Kauf Kabelreceiver HD«.
Unter »category« wird gespeichert, ob es sich um einen Besucher oder einen Anrufer handelt.
Die Spalte »target« beschreibt, wer Ziel der Kontaktaufnahme ist. In diesem Beispielfall ist der Inhalt »willy.tel«. Es könnte aber auch der Name einer in Kooperation stehenden Firma, für die Dienstleistungen übernommen werden, enthalten sein.

Tabelle »reasonCodes«

In dieser Tabelle werden die eigentlichen Kundenkontakte gespeichert. Neben der ID, die wieder den Primärschlüssel darstellt, wird in den Spalten »date« und »time« ein Zeitstempel gespeichert, wann dieser Datensatz angelegt wurde.
In der Spalte »ip« wird die IP des Rechners gespeichert, durch den der Datensatz in die Datenbank geschrieben wird, um in der späteren Auswertung Informationen darüber zu erhalten, wie viele Personen an dem jeweiligen Tag die gespeicherten Kundenkontakte hatten.
Falls eine gegebene Kategorie nicht in der Lage ist, das Kundenanliegen ausreichend genau abzubilden, wird dem Benutzer in dieser Spalte »text« die Möglichkeit gegeben, zu vorgegebenen Kategorien eigene Anmerkungen zu speichern. Die Spalte »reasonID« dient als Fremdschlüssel, durch den der Zusammenhang zu einer Kategorie hergestellt wird.

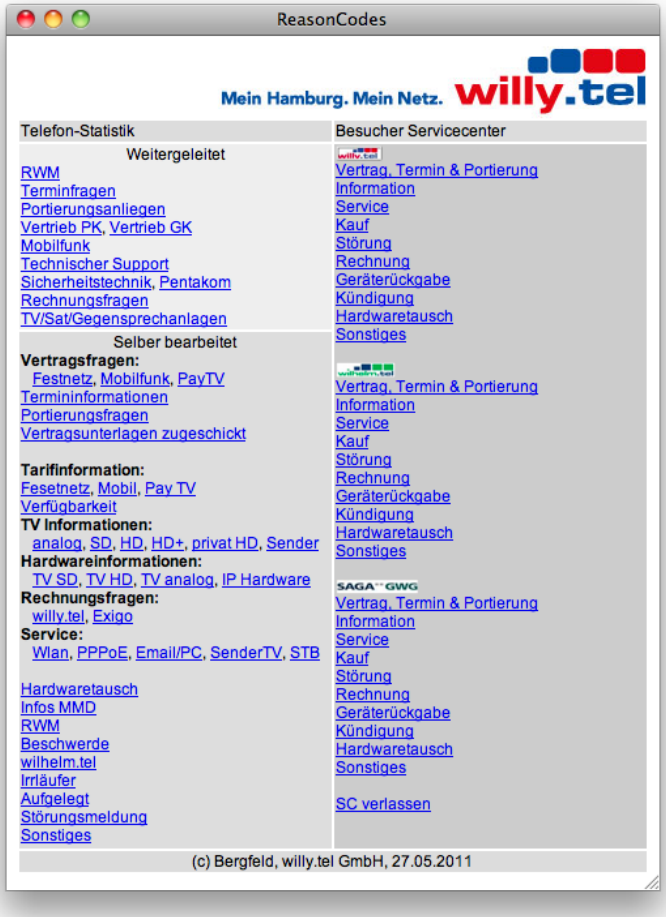

Abbildung 4.3: Screenshot des Logging-Tools

4.2.1.2 Das Logging-Tool

Dieses Tool ist die Benutzerschnittstelle zur Datenbank. Damit es nicht notwendig ist, jederzeit ein normales Browserfenster oder einen Tab geöffnet zu haben, wird dieses als Popup-Fenster umgesetzt (s. Abb. 4.3).

Die Gliederung der Kategorien erfolgt in einem einfachen Muster: Links werden Daten über Anrufer gespeichert, in der rechten Spalte Daten über Besucher des Servicecenters. Wird ein Link angeklickt, wird ein entsprechender Eintrag in der Datenbank abgelegt (R3), worüber der Benutzer eine Bestätigung erhält. Außerdem erhält er an dieser Stelle die Möglich-

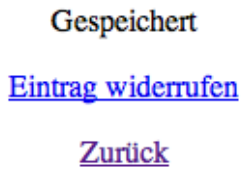

Abbildung 4.4: Screenshot des Logging-Tools nach dem Speichern

keit, den Eintrag rückgängigzumachen (R4), falls er die falsche Kategorie ausgewählt hat (s. Abb. 4.4).

Die Darstellung direkt auf der Startseite des Tools sollte sowohl übersichtlich als auch effizient zu nutzen sein. Diese Ansprüche erfordern das Treffen einer Auswahl, welche Punkte direkt auszuwählen sind und welche erst auf tieferen Ebenen verfügbar werden. So ist es wichtig, einen Kompromiss zu finden, zwischen der Vermeidung der Überladung der Oberfläche und der Anforderung, die benötigten Klickzahlen möglichst gering zu halten, damit es nicht erforderlich ist, mehrere Ebenen tief in eine Menüstruktur abzutauchen, um eine häufig verwendete Kategorie zu finden (RN2).
Im vorliegenden Fall ist die Menütiefe auf zwei Ebenen reduziert, so dass mit maximal zwei Mausklicks jede Kategorie erreicht werden kann. Verbirgt sich auf der Startseite statt der direkten Auswahl zur Speicherung in der Datenbank ein Oberbegriff, wie z.B. bei »Besucher« → »Information«, gelangt man zu Kategorien. Diese unterliegen zwar einer weiteren Hierarchie, die jedoch nicht durch eine noch tiefere Verschachtelung durch Hinzufügen weiterer Seiten, sondern über optische Einrückung der weiteren Punkte dargestellt werden. Dies ist in Abb. 4.5 zu sehen.

Die HTML-Dateien des Logging-Tools basieren auf einem einfachen Template, das die Darstellung definiert und dafür entwickelt wurde. Auf den Seiten ist jeweils eine Tabelle mit zwei Spalten eingefügt, die die Links beinhaltet. Diese Links, die über ein an den Link zum ausführenden PHP-Script übergebenes Argument den aktuellen Datensatz in die Datenbank schreiben, sind hart im HTML-Code gespeichert. Die Entscheidung, die Oberfläche des Tools statisch statt dynamisch zu entwickeln, hat zwei Gründe:

1. Ist das Tool initial mit Daten gefüllt, werden in Zukunft nur geringe Änderungen notwendig sein. Das Hinzufügen oder Umbenennen von einigen wenigen Kategorien ist sehr einfach möglich (RN4).

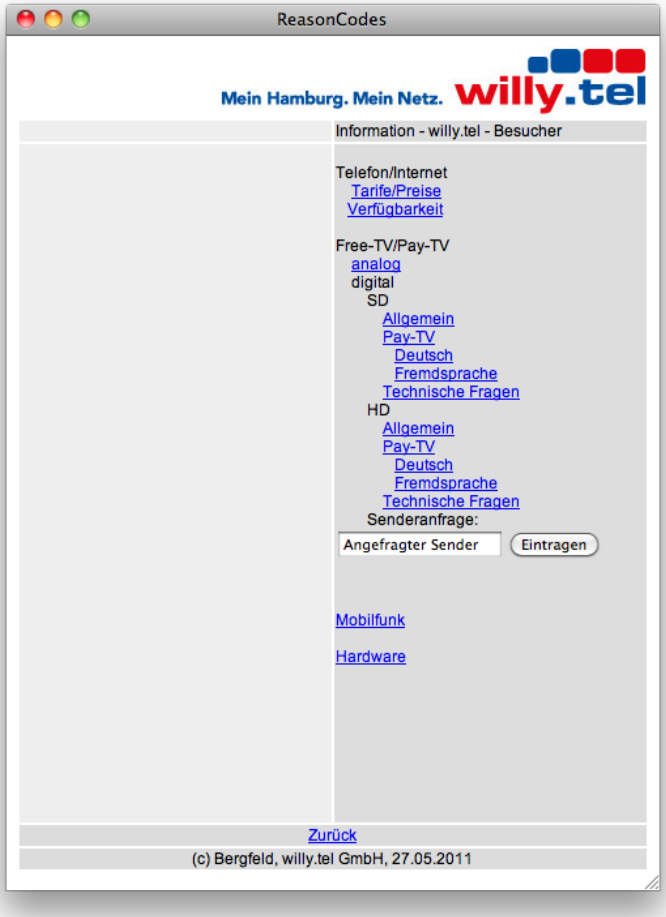

Abbildung 4.5: Unterkategorie »Informationen«

2. Eine statische Darstellung ist performanter als eine dynamisch erzeugte (RN3), da vom Server keine Verarbeitung von Datenbankabfragen notwendig ist.

Aus diesen Gründen ist die statische Umsetzung unter diesen Bedingungen die sinnvollere Wahl. Soll das Tool auf dynamische Generierung umgestellt werden, ist dies mit verhältnismäßig geringem Aufwand möglich. Der Tabelle »reasons« müssten zwei weitere Spalten hinzugefügt werden. Eine zur Angabe der Kategorie, in der das Element dargestellt werden soll und eine zweite, um die Reihenfolge zu bestimmen. Dann wäre es auch möglich, ein Admin-Tool zu entwickeln, mit dem das Hinzufügen und Ändern von Kategorien ohne jeglichen Kontakt mit Quellcode möglich wäre.

4.2.2 Möglichkeiten von ETL

Es ist die Ausnahme, dass die Freiheit besteht, die auszuwertenden Daten mit einem selbstentwickelten Tool in genau der Form zu speichern, die für die Aufbereitung erforderlich ist. Die Regel ist es eher, in die Situation zu kommen, Daten aus verschiedenen Quellen zu erhalten, die in unterschiedlicher Form gespeichert sein können, die dann trotzdem effizient verarbeitet werden müssen.
Für solche Fälle gibt es leistungsfähige Tools, die es beherrschen, diese Daten in drei Schritten in eine verwertbare Form zu bringen:

1. Extract: Die Daten werden aus der gegebenen Quelle eingelesen. Eine Zwischenspeicherung ist nicht notwendig, es ist ausreichend, Leserechte zu besitzen, so dass dieser Vorgang nicht destruktiv verläuft.
2. Transform: Es werden Regeln aufgestellt, wie mit den Daten verfahren werden soll. Von der Änderung des Zeichensatzes, über komplexe Ersetzungen von Schreibweisen, bis hin zu Sortierungen gibt es wenig Unmögliches.
3. Load: Die gelesenen/transformierten/bereinigten Daten werden in dem benötigten Format für die Weiterverarbeitung gespeichert. Empfehlenswert ist die Speicherung in einem RDBMS, jedoch ist es auch möglich, diese in CSV- oder Excel-Dateien zu exportieren.

Zur Pentaho Business Intelligence-Suite gehört die Komponente »Pentaho Data Integration« (PDI), die aus dem Kettle-Projekt hervorgegangen ist. Neben einer übersichtlichen und einfach zu verwendenden grafischen Oberfläche bringt diese mehr als 100 Mappingobjekte mit, mit deren Hilfe Eingaben, Ausgaben und Transformationen umgesetzt werden können. Außerdem besteht die Möglichkeit, die Komponente durch Plugins den eigenen Anforderungen anzupassen.

Die PDI-Komponente besteht aus mehreren Programmteilen:

- Spoon: Die grafische Oberfläche, mit der sich einfach Transformationen entwerfen lassen.
- Pan: Dieses Programm ermöglicht das Ausführen der zuvor mit Spoon erstellten Transformationen.
- Kitchen: Dieses Tool dient dazu, Aufgaben zeitgesteuert über eine Batch zu starten.
- Chef: Eine Hilfe, komplexe Datenbankoperationen in Jobs zusammenzufassen und automatisiert ablaufen zu lassen.
- Carte: Ein Web-Server, der es erlaubt, aus der Ferne den Zustand der PDI-Prozesse abzufragen.

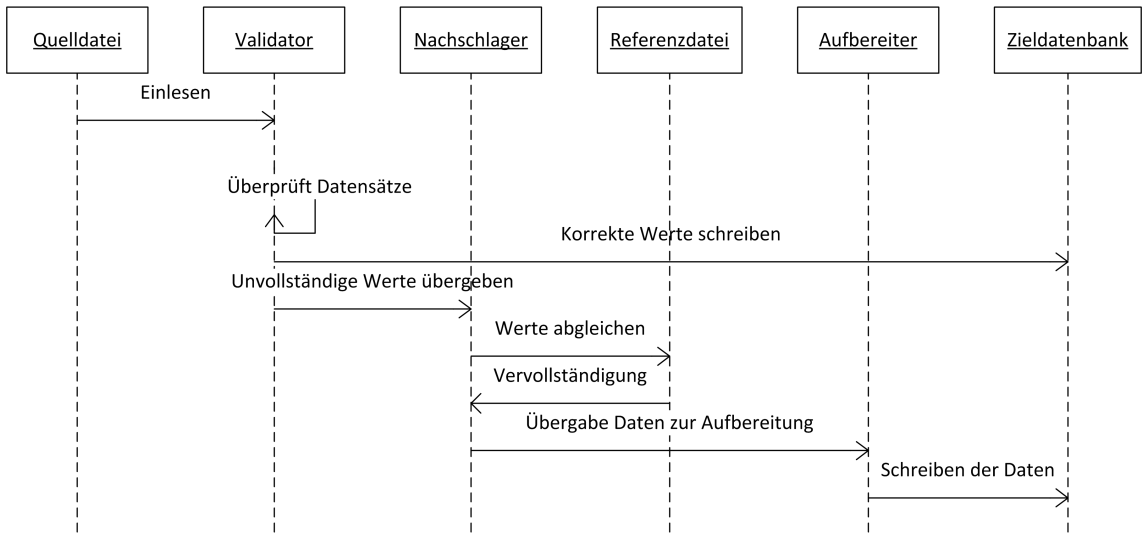

Abbildung 4.6: Sequenzdiagramm des Beispielablaufs

Im Folgenden soll ein Beispiel zeigen, wie eine Transformation in PDI erstellt wird und welche Schritte hierfür notwendig sind. Dieses besteht aus mehreren Schritten:

1. Einlesen einer Textdatei
2. Filtern von Datensätzen
3. Laden der Datensätze in eine relationale Datenbank
4. Validieren/Nachladen fehlender Daten
5. Abschluss und Ausführung der Transformation

Als Datenquelle dient hier eine Textdatei mit kommagetrennten Werten (CSV), die fiktive Kundendaten enthält (s. Tabelle 4.1).

Name	Vorname	Strasse	PLZ	Ort	Tel.
Mustermann	Max	Musterstr. 1	12345	Musterstadt	401234567
Musterfrau	Marianne	Beispielstr. 9a	23456	Beispielhausen	4312345678
Meier	Fritz	Ostergasse 44	34567	Osterdorf	45623456789
Schulze	Frieda	Bauernallee 89		Bauerndorf	304567890

Tabelle 4.1: Beispieldaten zur PDI-Transformation

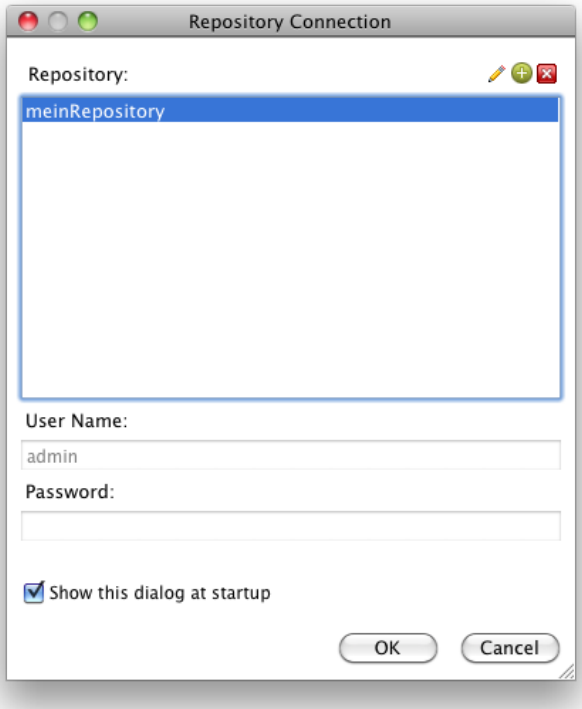

Abbildung 4.7: Spoon – Anmeldemaske am Repository

Der Ablauf der auzuführenden Transformation ist in Abb. 4.6 dargestellt.

Die erstellten Transformationen werden in einem Repository, also einem Workspace, gespeichert, zu dem beim Start des Designers eine Verbindung hergestellt werden muss, das durch ein Passwort geschützt sein kann. Dieses kann sowohl im Dateisystem als auch in einem Datenbanksystem gespeichert werden. Für den Einstieg empfiehlt sich hier das Speichern im Dateisystem, da man es sich dadurch ersparen kann, für die ersten Schritte extra eine Datenbank hochzufahren. Im späteren Produktivbetrieb sollte nicht zuletzt wegen erhöhter Sicherheit bei Parallelzugriffen die Datenablage in einer Datenbank favorisiert werden. Entscheidet man sich, das Repository in einer Datenbank anzulegen, fordert Spoon einen als Nächstes auf, eine Datenbankverbindung einzurichten. PDI ist — genau wie der restliche Teil der Pentaho Business Intelligence-Suite — nicht sehr wählerisch, was Datenbanksysteme angeht und ist damit in der Lage, mit annähernd allen verfügbaren Systemen, wie Oracle, MySQL, AS/400, MS Access, MS SQL DB2, SQLite und vielen weiteren, umgehen.

Zu bearbeitende Aufgaben werden in Spoon in Transformationen abgebildet, so dass für jede Aufgabe eine solche erstellt wird. Auf der linken Seite unter »Design« sind die Befehle,

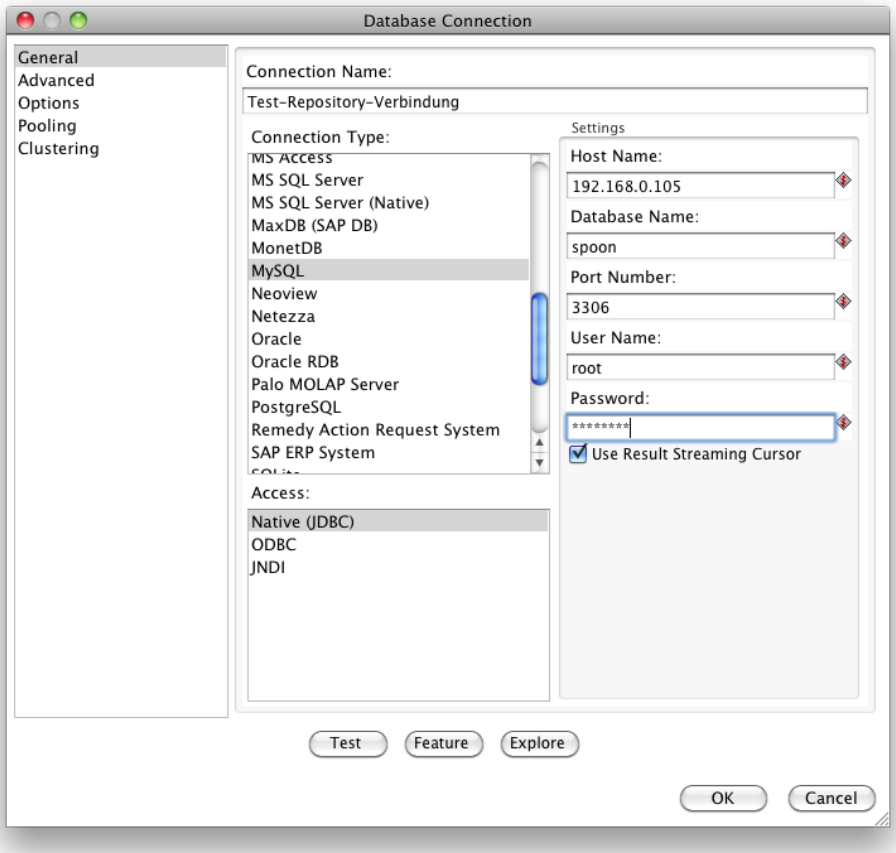

Abbildung 4.8: Herstellen einer Datenbankverbindung

die zur Verfügung stehen, aufgelistet. Hinter dem Eintrag »Input« ist der Befehl »Text File Input« zu finden. Dieser wird per Drag and Drop auf die Arbeitsfläche gezogen. Über einen Doppelklick auf das neue Element sind die Eigenschaften zu erreichen. Das Setzen einer aussagekräftigen Bezeichnung bietet sich an, hier »Einlesen der Kundendaten«.

Über die Schaltfläche »Browse« wird die einzulesende Datei ausgewählt und dem Arbeitsschritt hinzugefügt. Unter der Registerkarte »Content« sind bei Bedarf weitere Regeln für den Umgang mit der Datei anzugeben, wie den Seperator, der in diesem Fall auf Komma gesetzt wird. Außerdem wichtig ist die Angabe, ob Zeilenumbrüche in UNIX- oder DOS-Format erwartet werden. Die Registerkarte »Fields« bietet über die Schaltfläche »Get Fields« die Möglichkeit, das erste Mal aus der Datei zu lesen, so dass ersichtlich wird, ob die Einstellungen korrekt vorgenommen wurden und die Datei verarbeitet werden kann.

Die einzulesende Datei ist in diesem Fall unvollständig, in einem Eintrag fehlt ein Wert für die Postleitzahl. In einer solch kleinen Quelldatei ist das kein Problem, hätte diese aber mehrere

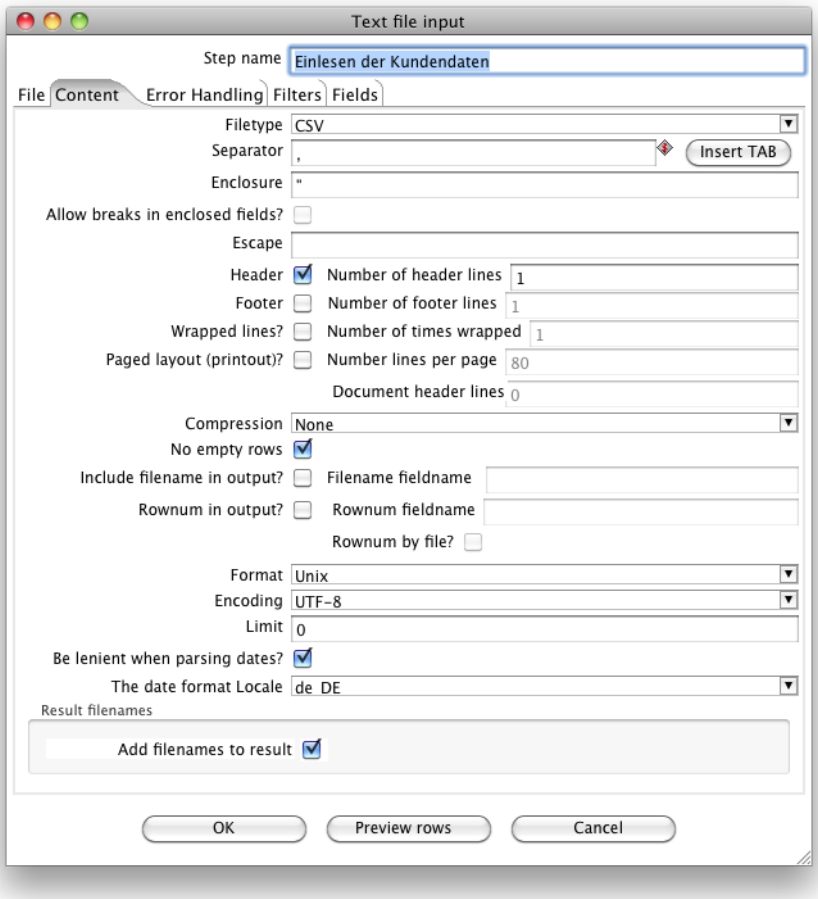

Abbildung 4.9: Eigenschaften Text file input

tausend Einträge, wäre der Zeitaufwand beträchtlich. Daher ist es erforderlich, die Transformation so anzupassen, dass sie in der Lage ist, das Problem selbstständig zu lösen. Dafür wird ein weiterer Schritt hinzugefügt: »Filter Rows«. Auf diesem Wege werden die Datensätze auf Bedingungen geprüft, anhand derer die weitere Verfahrensweise bestimmt wird.
Beide Schritte werden durch das Klicken mit gedrückter Shift-Taste auf den ersten und anschließende Ziehen und Loslassen des Mauspfeils auf den zweiten verbunden. Es erscheint ein Pfeil. Diese Verbindung wird »Hop« genannt.

In dem Dialogfeld, das sich nach einem Doppelklick auf den Arbeitsschritt öffnet, wird dieser Schritt wieder umbenannt und es wird festgelegt, wie er sich verhalten soll. Da die Spalte »PLZ« unvollständig ist, wird diese in dem linken Feld ausgewählt. Als »Function« wird »IS NOT NULL« gewählt. Für jeden Datensatz wird das Feld »PLZ« auf NULL geprüft, was

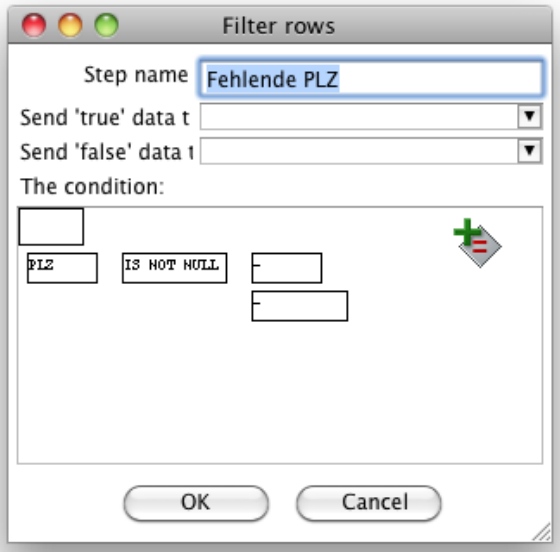

Abbildung 4.10: Eigenschaften Filter Rows

entweder zu true oder zu false evaluiert. Die Felder »Send true/false data to step« bleiben vorerst leer.

Die vollständigen Daten können direkt in die Datenbank geschrieben werden.
Für diese Operation wird ein weiterer Arbeitsschritt benötigt: »Table Output«. Dieser wird auf die Arbeitsfläche gezogen und über einen Hop mit dem Filter-Schritt verbunden. Es erscheint ein Auswahlfeld, ob dieser Weg bei true oder false gegangen werden soll. Es wird »true« ausgewählt.
In den Eigenschaften des Schrittes ist die Datenbankverbindung zur Zieldatenbank zu konfigurieren. Im Feld »Connection« wird die Verbindung ausgewählt, in das Feld »Target table« die Zieltabelle. Falls diese noch nicht existiert, lässt sie sich komfortabel durch einen Klick auf die Schaltfläche »SQL« generieren.
Der Schritt zum Schreiben in die Datenbank ist vollständig.

Die Vervollständigung der unvollständigen Daten ist der nächste Schritt. Dazu ist eine weitere Datei erforderlich, die zu Ortsnamen die passenden Postleitzahlen bereitstellt. Diese wird genau wie das Einlesen der Quelldaten über den »Text file input« gelesen.
Zusätzlich wird ein weiterer Schritt benötigt, der »Stream Lookup«, der auf die Arbeitsfläche gezogen wird und je einen Hop von dem neuen Schritt zum Postleitzahlenabgleich und den False-Arm des Schrittes »Fehlende PLZ« bekommt. Per Doppelklick werden die Eigenschaften festgelegt: Das Auswahlmenü »Lookup step« wird mit »Ort <-> PLZ« gefüllt und es wird

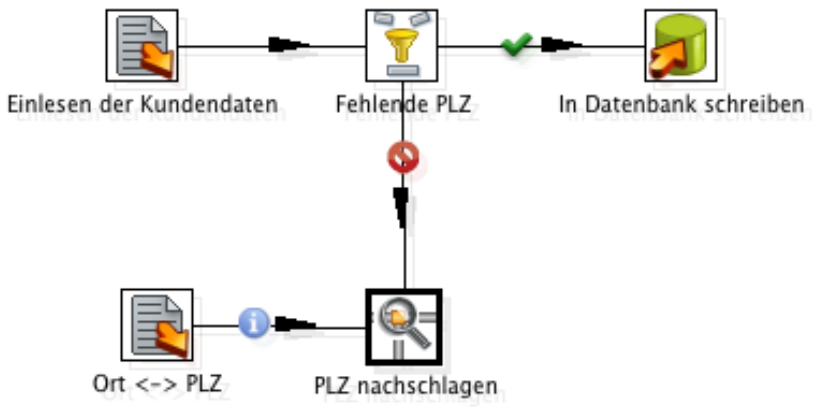

Abbildung 4.11: Transformation mit Arbeitsschritten und Hops

bei »Field« »Ort« ausgewählt. Schließlich werden über die Schaltfläche »Get lookup fields« die abzugleichenden Felder ausgelesen.

Bisher werden die korrekten Daten schon in die Datenbank geschrieben und die unvollständigen aussortiert und gegen eine Referenzdatei geprüft. Jedoch werden diese noch nicht in die Zieldatenbank geschrieben. Dies folgt im nächsten Schritt. Es wird ein Schritt »Select Values« hinzugefügt, der per Hop vom Schritt »PLZ nachschlagen« und zu Schritt »In Datenbank schreiben« verbunden ist. In den Eigenschaften werden per Klick auf »Get fields to select« die verfügbaren Feldnamen ausgelesen und der Eintrag »PLZ_1« aus der Verifikationsdatei an die Stelle des originalen Felds »PLZ« gebracht, das genau wie »Ort_1« entfernt wird. Unter der Registerkarte »Meta-data« werden Feldname und Typ angepasst, womit die Einrichtung des letzten Schrittes abgeschlossen ist.

Damit ist die erstellte Transformation bereit, ausgeführt zu werden. Es gibt drei verschiedene Möglichkeiten, dies zu tun:

1. Lokale Ausführung: Wird die Transformation so ausgeführt, wird sie lokal auf dem Rechner durchgeführt. Das setzt voraus, dass alle Daten, die verwendet werden, auch lokal zur Verfügung stehen. Diese Methode eignet sich gut zum Testen und wird im Praxiseinsatz nur selten Gebrauch finden, da die meist aufwändige Entwicklung von Transformationen nicht auf einen Produktivsystem durchgeführt werden sollte.
2. Remote Ausführung: Im Praxiseinsatz ist daher meist diese Variante zu finden. Die Transformation wird vom Client direkt auf dem Data Integration Server ausgeführt.
3. Cluster Ausführung: In sehr großen Umgebungen ist die Ausführung auch verteilt auf mehrere Cluster möglich.

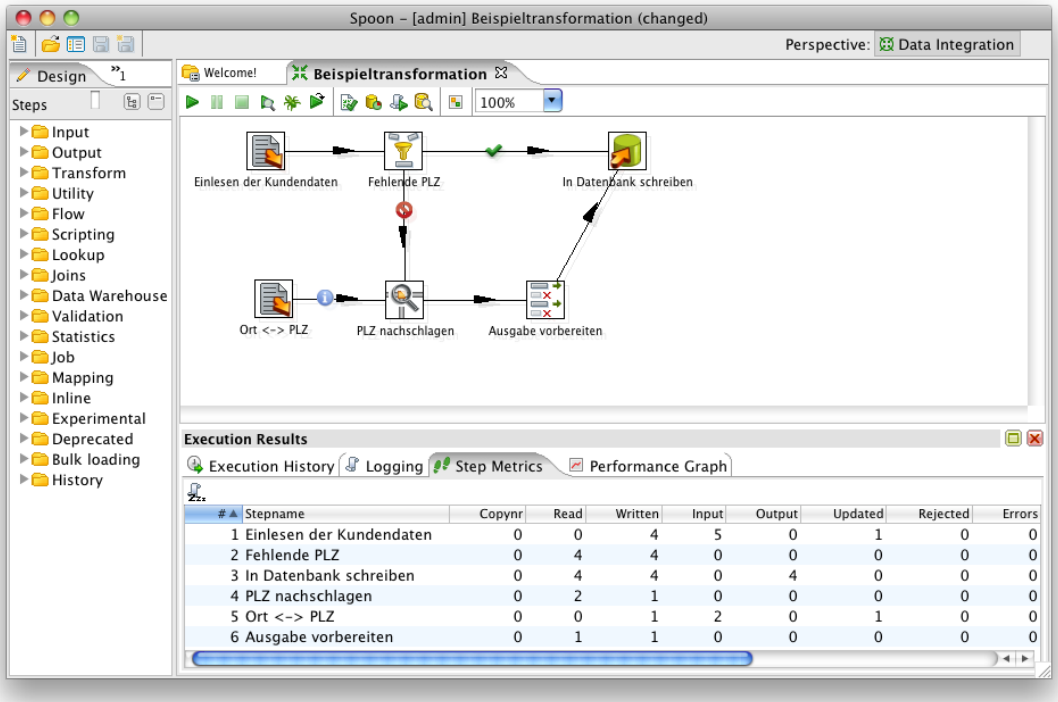

Abbildung 4.12: Vollständige Transformation und Ergebnisfeld nach Ausführung

Zum Testen wird in diesem Fall die lokale Ausführung gewählt. Unterhalb der Arbeitsfläche erscheint ein Feld mit Informationen über die Ausführung der einzelnen Schritte.

Das gerade gezeigte Beispiel stellt nur einen sehr kleinen Teil der Funktionalität von PDI dar. In der Praxis sind Aufgaben häufig nicht mit einer einzelnen Transformation zu lösen, sondern werden in Jobs zusammengefasst, die das Zusammenspiel von mehreren hintereinander durchgeführten Transformationen koordinieren, vorbereitende Aktionen durchführen wie das Prüfen auf Verfügbarkeit von Datenquellen, das Laden von sehr umfangreichen Datenbeständen aus mehreren Datenquellen und das Bearbeiten von Dateioperationen wie das Herunterladen von Daten von einem entfernten Server. Auch das automatische Versenden von Erfolgs- oder Fehlermeldungen per E-Mail ist durch Jobs zu bewerkstelligen.

4.3 Datenanalyse

Das Logging Tool ermöglicht es, Datensätze über Kundenkontakte zu speichern. Diese sollen nun analysiert werden. Da keine Eigenschaften der Datenspeicherung vorgegeben sind

499	2010-12-20	12:04:42	192.168.200.62	61
500	2010-12-20	12:04:46	192.168.200.62	15
501	2010-12-20	12:10:12	192.168.200.62	14
502	2010-12-20	12:13:35	192.168.200.62	11
503	2010-12-20	12:18:30	192.168.200.62	24
504	2010-12-20	12:32:54	192.168.200.62	61
505	2010-12-20	12:33:18	192.168.200.62	21
506	2010-12-20	12:43:25	192.168.200.62	21
507	2010-12-20	12:56:32	192.168.200.62	10
508	2010-12-20	14:15:00	192.168.200.26	7
509	2010-12-20	14:26:58	192.168.200.62	17

Abbildung 4.13: Ausschnitt der Tabelle »reasonCodes«

und diese selbst bestimmt werden können, ist es nicht notwendig, die Daten — wie zuvor vorgestellt mit Hilfe von ETL-Transformationen — weiter aufzubereiten oder zu validieren.

Gewünscht ist es, Reports aus den Daten zu generieren. Ein Report ist ein Bericht, der auf Datenanalyse basiert. Die Darstellung ist statisch, es ist also nicht möglich, auf einen erzeugten Report Einfluss zu nehmen. Es ist jedoch möglich, während der Erzeugung mit Hilfe von Dialogfeldern Attribute festzulegen, die die Handhabung der Daten während der Generierung beeinflussen.
Anhand dieser Anforderung ist es nicht erforderlich, das in Abschnitt 2.1.2.3 vorgestellte OLAP einzusetzen, da keine Echtzeitanalyse am Bildschirm gewünscht ist, sondern die Erstellung ausdruckbarer Berichte.

4.3.1 Erstellen von Reports

Innerhalb der Pentaho Business Intelligence-Suite werden Reports mit dem Pentaho Report Designer entworfen. Verwendung findet hier die Version 3.6.1. Diese Software ermöglicht die Erstellung verschiedener Arten von Reports. So ist es möglich, sich einen unter geringem Zeiteinsatz mittels eines Assistenten, der hier »Report Wizzard« genannt wird, zusammenzuklicken. Jedoch kann die Erstellung einen erheblichen Einsatz von Zeit und Mühe erfordern, wenn dieser sehr umfangreich ist.
Für den Einstieg empfiehlt es sich, den bereits angesprochenen Report Wizzard in Anspruch zu nehmen und sich auf Basis der mitgelieferten Beispieldaten einen Bericht generieren zu lassen, um einen ersten Eindruck vom Aufbau eines Reports zu erlangen.
Nach seinem Aufruf führt er den Benutzer durch den Erstellungsprozess, lässt das Layout

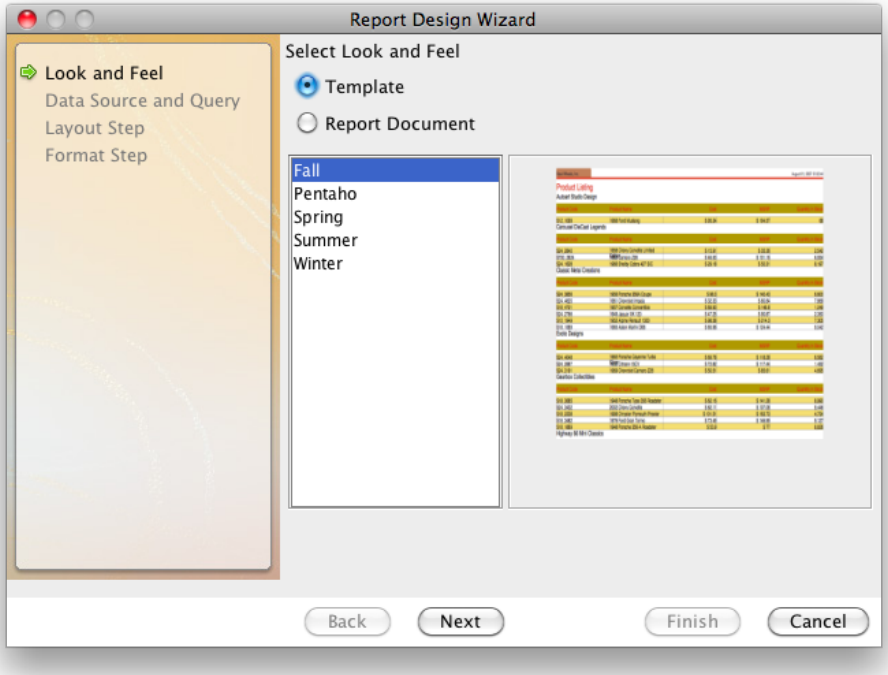

Abbildung 4.14: Report Designer Wizzard

anhand vorgegebener Templates bestimmen, Datenquellen auswählen und Felder bestimmen, die in dem erstellten Report verwendet werden.

Anhand des erzeugten Reports lassen sich viele Eigenschaften erkennen, die einem während der Reporterstellung immer wieder begegnen. Am auffälligsten ist die Einteilung der Fläche in verschiedene Bereiche wie Kopf- und Fuß-Zeilen und die Überschrift. Außerdem fällt auf, dass auch dann, wenn unter »File« → »Page Setup« das Papierformat auf »DIN-A4« oder ein anderes gewünschtes Format festgelegt wurde, der Report in der Layout-Vorschau nicht an die Höhenmaße der Auswahl heranreicht. Ursache dessen ist, dass der Report on demand gerendert wird und verschiedene Bereiche — hauptsächlich der Bereich »Details« — durch die Abfrage der Datenquelle eine nicht vorhersehbare Länge haben werden. Deshalb wird ein kürzeres Layout angezeigt, das nur nach Sektionen getrennt ist. Übersteigt die Datenausgabe die auf dem Ausgabeblatt zur Verfügung stehende Fläche, wird eine weitere Seite generiert. In diesem Fall erscheinen die Kopf- und Fuß-Zeilen auch auf den folgenden Seiten. Die »Header« und »Footer« bieten sich an, das Firmenlogo, das aktuelle Datum, die Seitenanzahl u.ä. aufzunehmen, wogegen die »Details«-Abschnitte die aufbereiteten Daten aufnehmen.

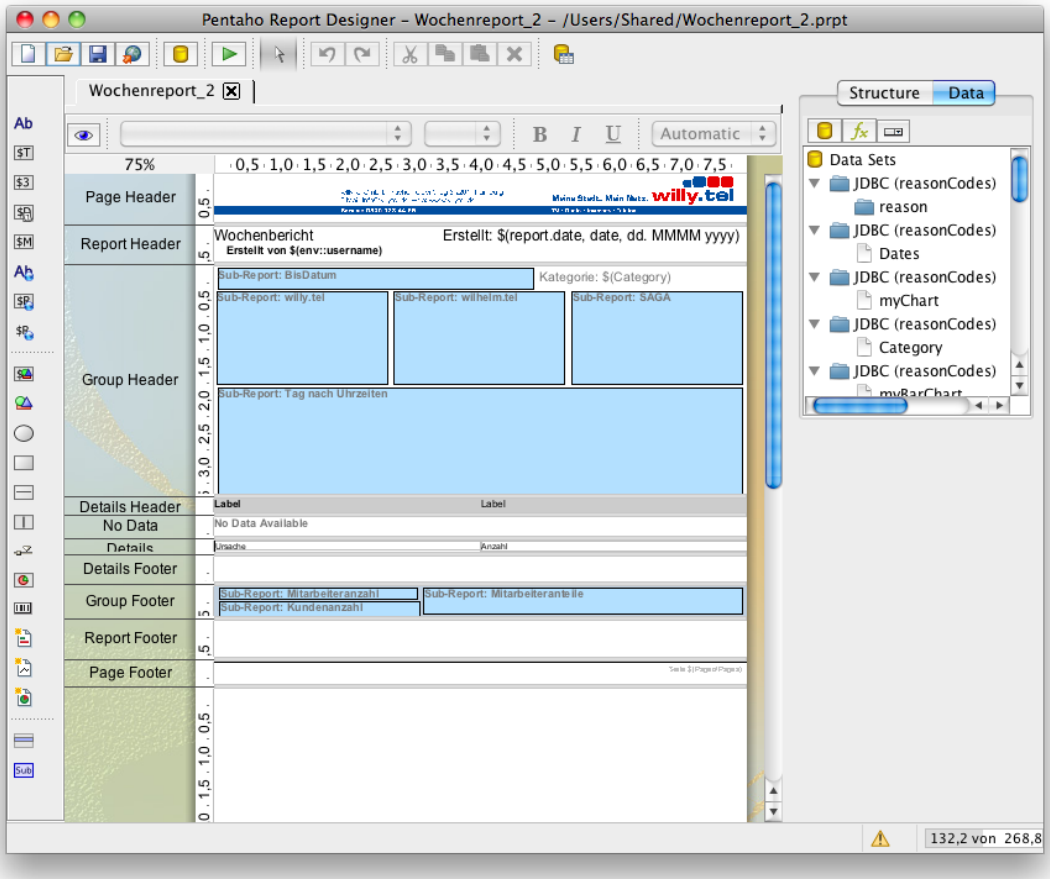

Abbildung 4.15: Report mit mehreren Sub-Reports

Innerhalb eines Reports kann eine Datenquelle und ein Datenbank-Query definiert werden. Dies ist hinderlich, wenn man nicht nur Daten beispielsweise tageweise sortiert aus einer Datenbank ausgeben will, sondern zusätzlich Berechnungen zu ihnen durchführen möchte oder es eine Grafik zur Visualisierung einzufügen gilt.
Weitere Datenquellen zu verwenden, ist durch das Einbetten von »Sub-Reports« in die Hauptdatei möglich. Ein Sub-Report verhält sich wie eine eigenständige Datei, d.h. sie kann eine eigene Datenquelle und einen eigenen Query zugewiesen bekommen. Genau wie ein eigenständiger Report kann er Attribute übergeben bekommen, mit denen gearbeitet werden kann.
Solche Attribute wirken sich auf die Verarbeitung der Daten vor der Erstellung des Reports aus, können also vom Benutzer gewählt werden. Dieser kann z.B. Zeiträume oder Kategorien vorwählen, um nur die Daten verarbeiten zu lassen, die gerade relevant sind.

Im vorliegenden Fall werden drei unterschiedliche Reports erstellt (R6):

1. Tagesreport: Dieser bietet die Möglichkeit, das Start- und das End-Datum (R7), sowie die Kategorie (Anrufer/Besucher) (R8) vorzuwählen. Es wird für jeden Tag eine eigene Seite erstellt.
2. Wochenreport: Hier wird die Möglichkeit gegeben, eine zuvor auszuwählende, ganze Woche auf einmal auszuwerten (R7). Auch hier findet sich wieder die Unterscheidung der Kategorie (R8).
3. Monatsreport: In diesem Report sind Start- und End-Datum frei wählbar (R7). Der angegebene Zeitraum wird auf einer Seite abgebildet, so dass so z.B. Monats-, Quartals- oder Jahres-Berichte erstellt werden können. Auch hier ist die Auswahl der Kategorie gegeben (R8).

In ihrer Darstellung unterscheiden sich die verschieden Reports leicht. Da es beim Tagesbericht wahrscheinlich ist, nur auf eine kleinere Teilmenge der verfügbaren unterschiedlichen Kategorien zu treffen, wird hier zur Darstellung ein Balkendiagramm verwendet. Bei Wochen- und Monats-Reports wird eine größere Vielfalt erwartet, weshalb die Datensätze hier zweispaltig in Tabellenform dargestellt werden. Links die Beschreibung der Kategorie, rechts die Anzahl der Vorkommen im gewählten Zeitraum.

Die erstellten Reports können direkt aus dem Report Designer in das Repository der Business Intelligence Suite gespeichert werden. Damit dies möglich ist, muss in der Pentaho-Konfiguration ein »Publishing Password« gesetzt werden[11], das zu diesem Zweck einzugeben ist.

In den meisten Fällen wird es verschiedene Reports unteschiedlicher Abteilungen geben. Aus Gründen des Datenschutzes ist es sinnvoll, dass nicht jeder Mitarbeiter, der Zugang zu dem System hat, auf alle Reports oder andere Inhalte zugreifen kann. Um dies sicherzustellen, bringt der Server eine Benutzer- und Gruppen-Verwaltung mit. In der in Abb. 4.16 gezeigten Administrations-Oberfläche[12] ist es möglich, Benutzer anzulegen und diese wiederum verschiedenen Gruppen zuzuordnen. In den Eigenschaften des Reports können dann für die angelegten Benutzer Berechtigungen festgelegt werden. Ebenso lassen sich zur Userverwaltung in dieser Konsole Wartungsaufgaben ausführen, wie z.B. das Löschen von Caches. Außerdem lassen sich Datenquellen definieren und bearbeiten, die vom Anwender verwendet werden können, um direkt aus der Webapplikation Inhalte zu erstellen oder zu modifizieren.

Ist der Report dem Server zur Verfügung gestellt, kann dieser über die Weboberfläche des Business Intelligence-Servers[13] nach erfolgreichem Login mit den eigenen Zugangsdaten (R5) aufgerufen und ausgeführt werden (s. Abb. 4.17).

[11] In der Datei /pentaho/pentaho-solutions/system/publisher_config.xml
[12] Erreichbar unter http://computername:8099
[13] Erreichbar unter http://computername:8080/pentaho/

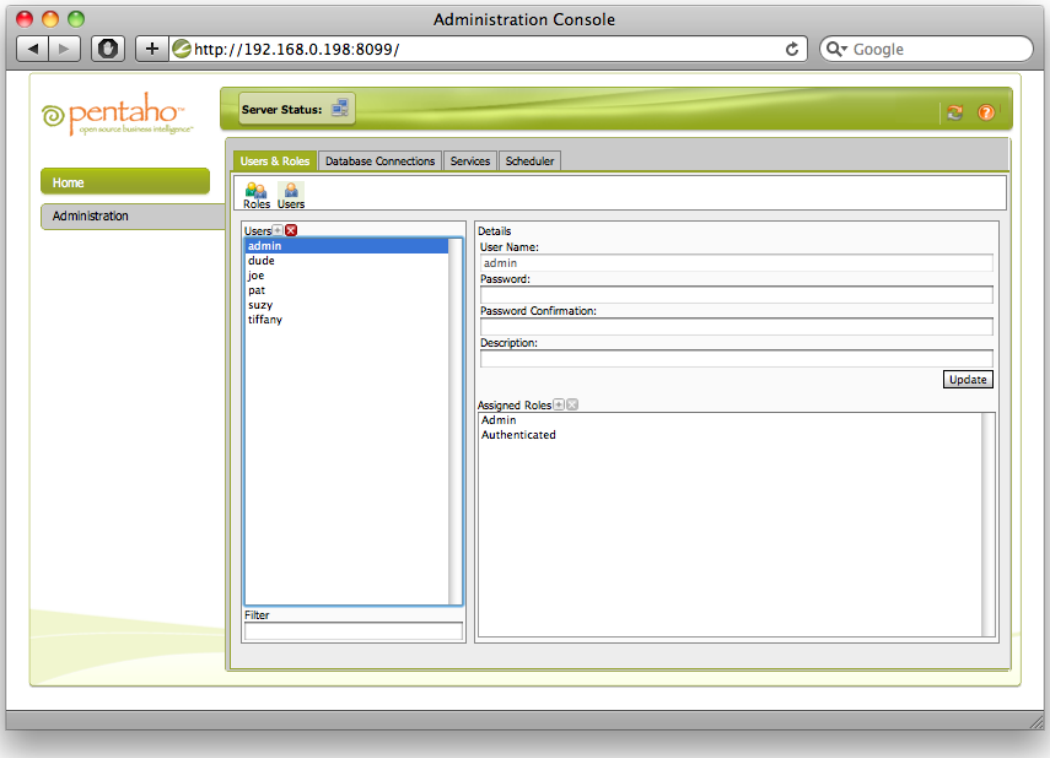

Abbildung 4.16: Administrations-Konsole

Der Bericht wird von der Software nun auf Basis der getroffenen Entscheidungen generiert und kann im gewünschten Format (PDF oder HTML) (R9) gespeichert und auf Wunsch ausgedruckt werden (R10). Während der Erzeugung wird ausschließlich lesend, also nichtdestruktiv, auf die Daten zugegriffen. Diese verbleiben also im System und können bei Bedarf zur Erstellung weiterer Berichte herangezogen werden (R11).
Das Verwenden definierter Reports ist so sehr einfach möglich, ohne dass es für den benutzenden Mitarbeiter erforderlich ist, sich Hintergrundwissen über die Funktionsweise anzueignen.

4.3.2 Test

Zur Einführung eines neuen Systems gehört es, sich davon zu überzeugen, dass dieses ordnungsgemäß arbeitet. Die Daten werden vom Logging-Tool in die Datenbank geschrieben und von dort aus vom Business Intelligence-System ausgelesen und zu Berichten aufberei-

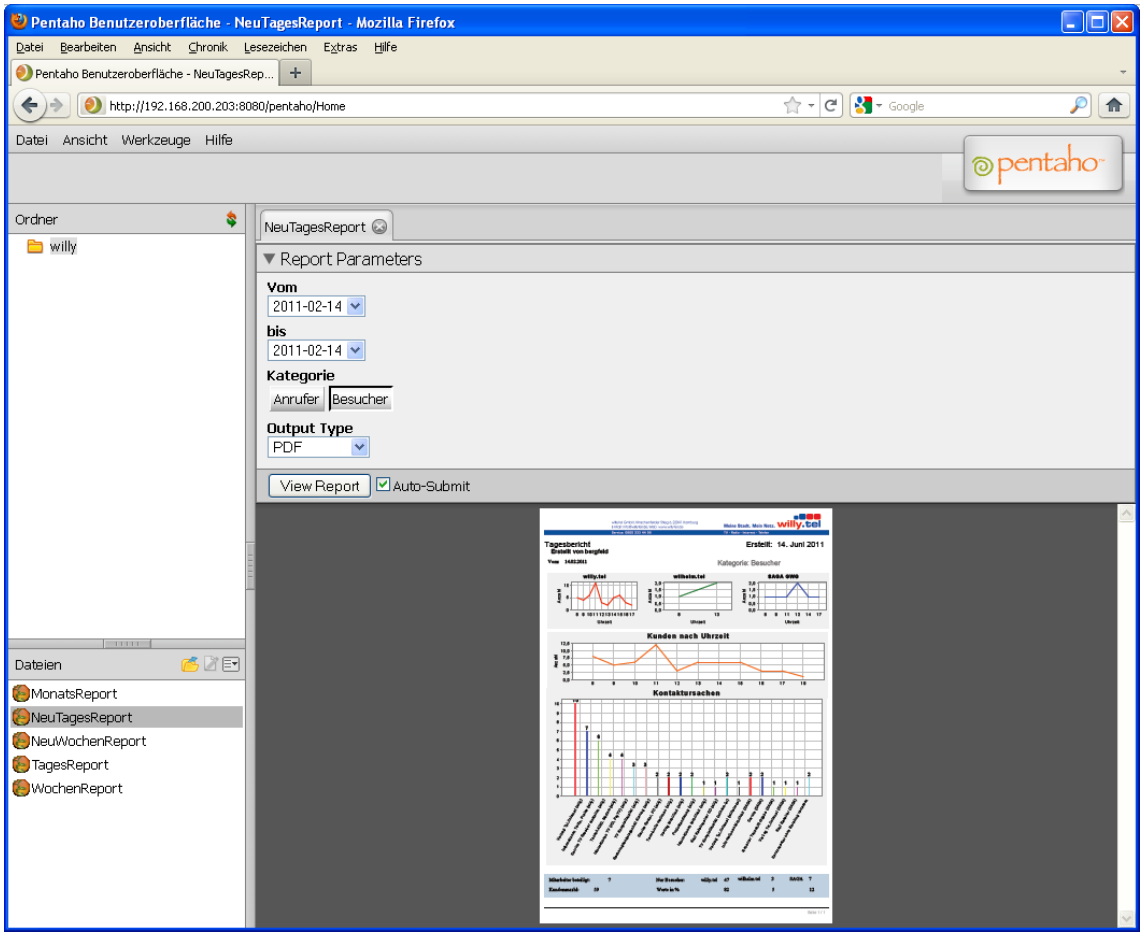

Abbildung 4.17: Tagesbericht im Server

tet. Aus dieser Abfolge ergeben sich verschiedene Schwachstellen, die es mittels Testfällen zu untersuchen gilt.

Das Logging-Tool betreffend:

- Wird beim Speichern eines Datensatzes durch das Logging-Tool die richtige Anzahl — nämlich nur ein Datensatz — in die Datenbank geschrieben?
- Wird der richtige Fremdschlüssel in die Spalte »reasonID« der Tabelle »reasonCodes« gespeichert?
- Arbeitet die Datenbank noch fehlerfrei, wenn mehrere Anfragen (nahezu) parallel gestellt werden?

Die Berichterstellung betreffend:

- Werden die zur Berichtserstellung benötigten Abfragen fehlerfrei ausgeführt, so dass alle betreffenden Datensätze gelesen und verarbeitet werden?
- Werden die Datensätze bei der Generierung von Charts korrekt interpretiert?
- Werden die zu verarbeitenden Datensätze richtig gezählt?
- Ergeben arithmetische Operationen, die während der Berichtserstellung auf die Daten angewendet werden, das richtige Ergebnis?

Nur wenn alle Testfälle fehlerfrei durchlaufen, ist es möglich, das System produktiv einzusetzen.

Die Testfälle des Logging-Tools lassen sich sehr einfach umsetzen. Wird ein kurzes PHP-Script entwickelt, das von mehreren Rechnern — im vorliegenden Fall drei — möglichst gleichzeitig aufgerufen wird und das über eine Schleife jeweils 50 Einträge in der Datenbank mit jeweils einer anderen Kategorieauswahl schreibt, lassen sich danach über kurze SQL-Abfragen die Fragen klären, ob die richtige Anzahl Datensätze in der Datenbank gespeichert ist — im Beispiel also 150 — und ob die richtige Anzahl unterschiedlicher »reasonIDs« — also jeweils dreimal 50 gleiche — vorhanden ist.
Dieses Vorgehen garantiert nicht, dass mehrere Anfragen exakt gleichzeitig gestellt werden. Überschneiden sich die Schleifendurchläufe der Testclients aber, ist die Wahrscheinlichkeit groß, dass sich die einzelnen Anfragen zeitlich ziemlich nahe kommen werden. Die Anzahl der Schleifendurchläufe und auch die der Testclients kann natürlich beliebig erhöht werden.

Die mit Hilfe des Logging-Tools erzeugten Testdaten lassen sich zum Testen der generierten Berichte weiterverwenden. Es empfiehlt sich, die Tests mehrfach mit unerschiedlichen »reasonIDs« aus bewusst gewählten unterschiedlichen Kategorien durchlaufen zu lassen und einige Daten nachträglich im Datum zu ändern, um anhand der Berichte prüfen zu können, dass auch die Unterscheidung nach Kategorien und entsprechend nach Datum korrekt arbeitet.
Nach erfolgreichem Abschluss aller Tests ist davon auszugehen, dass unter den gegebenen Bedingungen ein zuverlässiger Betrieb möglich ist und damit die nichtfunktionale Anforderung an Zuverlässigkeit (RN1) erfüllt ist.

4.3.3 Auswertung und methodische Abstraktion

Wie anhand der Tabelle 4.2 ersichtlich ist, sind die in 3.3.1 formulierten Anforderungen an das System umgesetzt. Im Entstehungsprozess zeigte sich, dass diese jedoch im Vorfeld

nicht vollständig definiert waren. So wurden während der Entwicklung Änderungen notwendig, die sowohl das Datenmodell betrafen als auch die Kategorisierung. Der Tabelle »reasonCodes« wurde die Spalte »text« hinzugefügt, die dazu dient, bei Bedarf eine kurze Präzisierung zu dem Datensatz speichern zu können. Dass diese Anforderung bestehen würde, war zu Beginn der Entwicklung nicht definiert. Auch zeigte sich, dass die im Vorfeld erarbeitete erweiterte Kategorisierung der Kundenkontakte und die Präsentation im Logging-Tool nicht vollständig war und während der Einführung mehrfach geändert und erweitert werden musste. Diese Erkenntnisse belegen, dass Entwicklung, Einfürung und Betrieb eines solchen Systems wohl nur in absoluten Ausnahmefällen linear nach dem Wasserfallmodell[14] erfolgen kann, sondern es sich dabei um ein rein iteratives Vorgehen handelt.

Anforderung	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11
erfüllt	•	•	•	•	•	•	•	•	•	•	•

Tabelle 4.2: Umgesetzte Anforderungen

Die entwickelten Methoden zur Datenerfassung und -Analyse sind keineswegs unternehmens- oder branchenspezifisch. Betrachtet man die gestellten Anforderungen etwas allgemeiner, ist zu erkennen, dass diese durchaus in völlig andere Umgebungen übertragbar und auch in denen interessant sind. Sie sind übertragbar auf beliebige Geschäftsbereiche, in denen Kundenkontakt besteht.

4.4 Fazit

Insgesamt kann festgestellt werden, dass auch im Bereich Business Intelligence Open Source Software ein ernsthafter Konkurrent für kommerzielle Produkte ist. Für alle benötigten Aufgaben stehen sehr gute, frei verfügbare und kostenlos einsetzbare Produkte bereit, mit denen die Anforderungen an das gewünschte System abgebildet werden können.
Im Laufe der Einführung hat sich gezeigt, dass die Software nicht nur äußerst stabil läuft, sondern sich auch während des Betriebs ohne große Probleme auf neue Bedürfnisse anpassen lässt.
Die Systemanforderungen an ein solches System sind nicht sehr hoch, so dass in vielen Fällen sogar ein eigentlich ausgedienter Rechner ausreichend ist, um die verschiedensten Auswertungen erstellen zu können.
Ein solches System aufzusetzen und in Betrieb zu nehmen, ist allerdings ein relativ aufwändiges Projekt und sollte umsichtig vorbereitet werden. Unerlässlich ist es, vor dem Beginn

[14] http://de.wikipedia.org/wiki/Wasserfallmodell

der Entwicklung einen Anforderungskatalog zu erstellen, der genau beschreibt, welche Arten von Daten auf welche Weise verarbeitet werden sollen.

Letztendlich hat sich gezeigt, dass es auch für mittelständische Unternehmen nicht nur erstrebenswert ist, Business Intelligence-Strukturen zu schaffen, um die Geschäftsdatenanalyse zu verbessern und maßgeblich zu erleichtern, sondern dass es Softwarelösungen gibt, die es kombiniert als Ganzes ermöglichen, Business Intelligence-Systeme, die in der Lage sind, komplexe Analyseanforderungen abzubilden, mit einem geringen finanziellen Einsatz einzuführen und zu betreiben. Von diesem Einsatz stellt der Löwenanteil einmalige Kosten für die Entwicklungszeit dar, Wartungs- und Betriebskosten sind gering und Lizenzkosten nicht vorhanden.

Die Einführung rechnet sich für den Betrieb sogar in vielfacher Hinsicht: Durch die automatische Verarbeitung der Daten und die dadurch gewonnene Flexibilität im Bezug darauf, bei Bedarf schnell und unkompliziert weitere Kategorien hinzuzufügen, werden weit präzisere Ergebnisse erreicht, die es der Geschäftsführung erleichtern, die Bedürfnisse und Anliegen der Kunden zu erfassen, als es mit manuellen Methoden möglich wäre. Außerdem wird eine massive Zeitersparnis in der Berichterstellung erreicht, durch die sich die Entwicklungskosten schnell amortisieren. Rechnet man für die Übertragung der auf händisch erstellten Strichlisten enthaltenden Daten in ein Tabellenkalkulationsdokument einen Aufwand von etwa drei Stunden pro Woche und veranschlagt, dass die automatische Erstellung eines Berichts in etwa drei Minuten möglich ist, ergibt dies eine Zeitersparnis von etwa 12 Stunden pro Monat oder ca. 7,5% der Arbeitszeit des Mitarbeiters.
Natürlich verursachen Einführung und Betrieb eines Business Intelligence-Systems ebenfalls Kosten, jedoch amortisieren sich diese ob der Regelmäßigkeit der durchzuführenden Auswertungen schnell.

5 Zusammenfassung und Ausblick

In diesem Kapitel werden die Ergebnisse der vorliegenden Arbeit zusammengefasst und bewertet. Der Ausblick zeigt, wie eine Erweiterung des Business Intelligence-Systems in einem Unternehmen über die Themen, die im Rahmen dieser Bachelorarbeit bearbeitet werden konnten, hinaus möglich ist.

5.1 Zusammenfassung und Bewertung

Im Rahmen dieser Arbeit wurde die Frage untersucht, ob es in einem mittelständischen Unternehmen sinnvoll und praktikabel ist, Systeme zur Erfassung und Analyse von Geschäftsdaten mit Hilfe der Pentaho Business Intelligence-Suite einzuführen. Verschiedene Softwarekomponenten, die in der Lage sind, diese Aufgaben zu übernehmen, wurden vorgestellt und diskutiert (Kapitel 2).

Nach der darauffolgenden Untersuchung von möglichen Szenarien, wurden die Anforderungen an ein solches System formuliert (s. Kapitel 3).

Aufbauend auf diesen wurde ein Zielsystem skizziert und schließlich entwickelt. Dabei wurde neben der Architektur des Zielsystems auch vorgestellt, welche Verfahren zur Erhebung und Verarbeitung der zu analysierenden Daten verfügbar sind und unter welchen Umständen diese eingesetzt werden (Kapitel 4).

Das Ergebnis ist ein System, durch das eine gewisse Analyseinfrastruktur geschaffen wurde. Es übernimmt die Aufgaben, die im Vorfeld festgelegt wurden, steht aber ebenso für umfangreiche Erweiterungen zur Verfügung.

Die Arbeit mit den vorgestellten Applikationen verlief in weiten Teilen erfreulich. Die Programme weisen einen überraschend hohen Reifegrad vor, der sich sowohl in guter Usability äußert, als auch in hoher Stabilität. Einzig der Pentaho Report Designer legte bisweilen ein unkooperatives Verhalten an den Tag, wenn er sich bei sehr komplexen Reports weigerte, den gerade bearbeiteten Report zu speichern und stattdessen eine Exception warf. Die Ursache für dieses Verhalten konnte während der Arbeit nicht in Erfahrung gebracht werden.

Durch die Tatsache, dass es sich bei aller eingesetzter Software um freie Software handelt, ist die Versorgungslage mit Literatur in einigen Bereichen als angespannt zu beschreiben. Dies trifft natürlich nicht auf das Betriebssystem oder die Datenbank zu, jedoch generell auf Komponenten von Business Intelligence-Systemen und damit auch auf die Pentaho Suite. Wenige Informationen sind in gedruckter Form verfügbar, an vielen Stellen ist es erforderlich, sich benötigte Informationen durch Wikis auf Projektseiten oder Foren zu beschaffen. Jedoch ist davon auszugehen, dass sich mit steigender Verbreitung von Business Intelligence-Anforderungen auch durch kleinere Unternehmen die Literaturversorgung erweitern wird.

5.2 Ausblick

Die in dieser Arbeit gezeigte Einführung der Pentaho Business Intelligence-Suite ist ein erster Schritt, diese Technologie einzusetzen. Zu diesem Stand werden Besucher- und Anrufaufkommen in beliebigen Zeitfenstern analysiert, und zwar weitaus effizienter, als dies händisch möglich wäre. Um die Möglichkeiten der Suite jedoch weiter auszunutzen, müssen Überlegungen angestellt werden, welche weiteren Daten es zu analysiert gilt, ob diese schon in digitaler Form vorliegen oder erst erfasst werden müssen und schließlich auf welche Art diese weiter aufbereitet werden sollen.

Sehr interessant ist es, die zukünftigen Entwicklungen in diesem Bereich zu beobachten. Dieses Themengebiet bietet noch sehr viel Potential, das bisher nicht ausgeschöpft wird. So gilt es, zu untersuchen, wie Business Intelligence-Systeme für die Anforderung zu realisieren sind, Daten nicht nur innerhalb der Firma zu verarbeiten, sondern auch komfortabel über mobile Endgeräte. Auch der Verlagerung von Geschäftsdaten oder -Server in dezentrale Systeme, wie beispielsweise Cloud-Dienste, gilt es, Beachtung zu schenken. Dies fordert nicht nur neue Lösungen für die Umsetzung, sondern stellt auch Anforderungen an neue Sicherheitskonzepte.

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[von Cube 2003] CUBE, Alexandra von: Eine Einführung in die Pivottabellen in Excel. (2003), 03. – URL http://www.c-c-center.de/nuetzliches/Pivottabellen_in_Excel.pdf

[Diedrich 2006] DIEDRICH, Oliver: MySQL: Die freie Wahl. (2006), 04. – URL http://www.heise.de/open/artikel/MySQL-Die-freie-Wahl-221933.html. – Zugriffsdatum: 21.04.2011

[Engels 2010] ENGELS, Christoph: *Basiswissen Business Intelligence*. W3L, 10 2010 (978-3-937137-37-7)

[Held und Klose 2007] HELD, Marcus ; KLOSE, Ingo: Business Intelligence mit Pentaho. (2007), 12. – URL http://www.heise.de/open/artikel/Business-Intelligence-mit-Pentaho-222177.html. – Zugriffsdatum: 24.04.2011

[Heuer und Saake 2000] HEUER, Andreas ; SAAKE, Gunter: *Datenbanken: Konzepte und Sprachen*. mitp, 01 2000 (3-8266-0619-1)

[Hyde 2006] HYDE, Julian: Mondrian Documentation. (2006), 08. – URL http://mondrian.pentaho.com/documentation/olap.php

[Kleijn 2006a] KLEIJN, Alexandra: Business Intelligence mit Open Source. (2006), 06. – URL http://www.heise.de/open/artikel/Business-Intelligence-mit-Open-Source-221947.html. – Zugriffsdatum: 24.04.2011

[Kleijn 2006b] KLEIJN, Alexandra: Open-Source-Lizenzen. (2006), 07. – URL http://www.heise.de/open/artikel/Open-Source-Lizenzen-221957.html. – Zugriffsdatum: 15.04.2011

[Köster 2002] KÖSTER, Mathias: Business-Intelligence und Data-Warehouse: Datenfriedhof oder Schatztruhe? (2002), 10. – URL http://www.contentmanager.de/magazin/artikel_235_business_intelligence_data_warehouse.html. – Zugriffsdatum: 31.03.2011

[LWN 1998] LWN: 1998 in review. In: *Linux Weekly News* (1998). – URL http://lwn.net/1999/features/1998timeline/. – Zugriffsdatum: 10.04.2011

[Manageability.org 2009] MANAGEABILITY.ORG: Open Source ETL (Extraction, Transform, Load) Written in Java. (2009), 02. – URL http://www.manageability.org/blog/stuff/open-source-etl. – Zugriffsdatum: 24.04.2011

[Manhart 2008] MANHART, Klaus: BI-Methoden (Teil 1): Ad-hoc Analysen mit OLAP. (2008), 04. – URL http://www.tecchannel.de/server/sql/1751285/bi_methoden_teil_1_ad_hoc_analysen_mit_olap/

[Mertens 2002] MERTENS, Peter: Business Intelligence - ein Überblick. In: *Arbeitspapier 2/2002 der Universität Erlangen-Nürnberg, Bereich Wirtschaftsinformatik I* (2002), 02

[O'Reilly und Associates 1999] O'REILLY ; ASSOCIATES: *Open Source - kurz und gut*. O'Reilly Verlag, 05 1999. – URL http://www.oreilly.de/german/freebooks/os_tb/toc.html

[OSI 1998] OSI: The Open Source Definition. (1998). – URL http://www.opensource.org/docs/osd. – Zugriffsdatum: 15.04.2011

[Pientka 2008] PIENTKA, Frank: Berichtssoftware - warum nicht Open Source? (2008), 09. – URL http://www.computerwoche.de/software/bi-ecm/1871839/. – Zugriffsdatum: 10.07.2011

[Reibold 2010] REIBOLD, Holger: *Pentaho Kompakt*. Brain Media, 2010 (978-3-939316-52-7)

[Wikipedia 2011a] WIKIPEDIA: Data Mining. (2011), 04. – URL http://de.wikipedia.org/wiki/Data_Mining. – Zugriffsdatum: 14.04.2011

[Wikipedia 2011b] WIKIPEDIA: ETL-Prozess. (2011), 03. – URL http://de.wikipedia.org/wiki/ETL-Prozess. – Zugriffsdatum: 13.04.2011

[Wikipedia 2011c] WIKIPEDIA: GNU General Public License. (2011), 04. – URL http://de.wikipedia.org/wiki/GNU_General_Public_License. – Zugriffsdatum: 21.04.2011

[Wikipedia 2011d] WIKIPEDIA: Intelligenz. (2011), 03. – URL http://de.wikipedia.org/wiki/Intelligenz. – Zugriffsdatum: 10.03.2011

Index

Versicherung über die Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §16(5) APSO-TI-BM ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen habe ich unter Angabe der Quellen kenntlich gemacht.

Hamburg, 26. Juli 2011	
Ort, Datum	Unterschrift