Diplomarbeit

Deskriptives Data-Mining
für Entscheidungsträger:
Eine Mehrfachfallstudie

Benedikt Kämpgen

Lehrstuhl für Informatik VI

Abgabe: 31. Dezember 2009

Betreuer:
Prof. Dr. Frank Puppe
Dr. Martin Atzmüller
Dipl.-Inform. Florian Lemmerich

Verfasser der Diplomarbeit:
Benedikt Kämpgen
Am Altenberg 40
97078 Würzburg

Julius-Maximilians-Universität
Würzburg
Sanderring 2
97070 Würzburg

Telefon: (09 31) 31-0
Fax: (09 31) 31-2600
www.uni-wuerzburg.de

Lehrstuhl für Informatik VI der
Universität Würzburg
Am Hubland
97074 Würzburg

Telefon: (09 31) 888-6731
Fax: (09 31) 888-6732
www.is.informatik.uni-wuerzburg.de

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Diplomarbeit selbständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe.

Würzburg, den 30. Dezember 2009

.....
Benedikt Kämpgen

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Quellcodeverzeichnis

1 Einleitung

Im Data-Mining wird Wissen aus Daten extrahiert. Kundeninteressen [43], Lebensversicherungen [69], Spieleentwicklung [14], kardiologische Diagnostik [40], Schmerzerkennung [50] oder Standortplanung [49] – schon diese wenigen Beispiele für Anwendungsfelder des Data-Mining zeigen, dass nicht Daten das zentrale Thema darstellen, sondern vielmehr allgemeine Probleme; erst für ein konkretes Problem kann entschieden werden, ob es durch die Wissensentdeckung in Daten gelöst werden kann. Grundsätzlich kann Data-Mining immer eine Lösung darstellen und besitzt damit besonders für Wissenschaft und Industrie ein großes Potenzial [48, S. 88]. Große Firmen wie IBM[1] investieren bereits in Data-Mining-Systeme und -Expertise, um dieses Potenzial auszuschöpfen – oder werden es noch tun[2].

Als kritische Entwicklung wird allerdings gesehen, dass bis zu 60% der Data-Mining-Projekte scheitern (vgl. [48]; [35]; [51]; [34]).

Probleme, die zum Scheitern eines Projekts führen können, gibt es zahlreiche, z.B. „Mythen“ [37], „Worst-Practices“ [54] oder „Fehler“ [50, S. 734-753]. Diese sollen im Folgenden als Einführung in die Arbeit dienen.

Zwei wesentliche Gruppen von Problemen können unterschieden werden. Die erste Gruppe, darunter die „Mythen“, beschreibt falsche Vorstellungen zum Data-Mining, die bereits vor Beginn oder zu Anfang eines Projekts auftreten können:

- Data-Mining ist zu komplex und nur von Technologieexperten in einem Labor durchführbar: In dieser Arbeit werden zwei Data-Mining-Projekte detailliert beschrieben, um einen Eindruck von der Komplexität und dem Aufwand zu vermitteln.
- Data-Mining benötigt eine große Datenbank, oder sogar ein Data-Warehouse: Die Arbeit belegt, dass derartige Voraussetzungen im Grunde nicht bestehen, vielmehr fast beliebige Daten analysiert werden können, indem ein auf das Projekt zugeschnittenes eigenes Data-Warehouse erstellt wird.
- Erwartungen an das Data-Mining sind zu hoch, jegliche Probleme lassen sich lösen, zudem automatisch: Auch wenn es eines der Ziele des Data-Mining ist, die Entdeckung von

[1] Jack Noonan (CEO SPSS Inc.), http://www.spss.com/ibm-announce/, Dezember 2009

[2] IBM's SPSS Deal May Spark BI Market Consolidation, http://www.kdnuggets.com/news/2009/n16/41i.html, Dezember 2009

Wissen weitgehend zu automatisieren (vgl. [21]), kann das bereits am Beschreiben und Erklären von Daten scheitern, einer vermeintlich leichteren Aufgabe. Deskriptives Data-Mining ist in vielen Anwendungsbereichen nützlich, dennoch muss abgewogen werden, ob ein anderer Lösungsansatz sinnvoller ist. Die in der Arbeit betrachteten Projekte zeigen, was mittels Deskriptivem Data-Mining erreicht werden kann, außerdem, welche der nötigen Teilaufgaben sich automatisieren lassen.

- Das Ergebnis des Data-Mining hängt nur vom Umfang der Daten und der Leistungsfähigkeit eines Algorithmus ab: Die beschriebenen Projekte zeigen vielmehr, dass die Herausforderung für erfolgreiches Data-Mining darin besteht, eine genau definierte Teilmenge der Gesamtdaten auszuwählen und so vorzubereiten, dass ein Algorithmus darauf ausgeführt werden kann.
- Data-Mining ist eine „Kunst“, die schwer erlernbar ist und Talent erfordert (vgl. [53]; [30]): Die nachvollziehbare Beschreibung des Data-Mining in dieser Arbeit widerspricht dem. Wenn auch bestimmte Tätigkeiten im Data-Mining von kreativem Geschick profitieren, werden stets Techniken verwendet, die erlernbar und größtenteils von Übung und Erfahrung geprägt sind.
- Endnutzer können die Arbeit der Data-Miner nicht nachvollziehen, da, ähnlich wie in der Software-Entwicklung, das „we are different syndrome“ [59] auftaucht: Die Arbeit identifiziert Teile des Data-Mining, die von einem Endnutzer für ein erfolgreiches Vorhaben verstanden werden müssen und stellt eine Vorgehensweise für Endnutzer-verständliches Data-Mining vor.

Die zweite Gruppe an Problemen, darunter die „Fehler“ und „Worst-Practices“, besteht aus falschem Vorgehen während eines Data-Mining-Projekts:

- Data-Mining wird ohne Vorgehensplan und sehr „ad-hoc“-mäßig [70] durchgeführt. In dieser Arbeit werden nicht nur die Aufgaben des Data-Mining – das Was – beschrieben, sondern auch die Techniken, die nötig sind, um diese Aufgaben zu erledigen – das Wie.
- Data-Mining wird ohne ausreichendes Verständnis der Daten und des Anwendungsbereichs durchgeführt. Die Data-Mining-Projekte aus dieser Arbeit werden jeweils mit einem Daten- und Domänen-Experten durchgeführt, die dieses Hintergrundwissen bieten. Dennoch werden ihre Aussagen nicht einfach übernommen, sondern anhand der Daten überprüft.
- Die Data-Miners verlassen sich zu stark auf ihr Gedächtnis. Diese Arbeit wird eine Möglichkeit vorstellen, um Data-Mining-Projekte zu dokumentieren und Wissensmanagement zu betreiben.
- Der Interaktion zwischen Endnutzer und Data-Miners wird eine zu geringe Bedeutung zuteil; in Folge dessen werden die falschen Ziele erfüllt. Diese Arbeit schlägt vor, Ziele in

unmissverständliche Anforderungen an das Projekt zu übersetzen, mit denen die Endnutzer einverstanden sind und die von den Data-Miners umgesetzt werden können.

- Die Behandlung der Daten geschieht zu oberflächlich, es werden zu viele Daten verwendet oder zu wenige, oder die Daten werden nicht genügend vorverarbeitet. Daten sind nie perfekt. Teile können stets irrelevant oder fehlerhaft sein und müssen gefiltert werden. Auch liegen sie häufig nicht in einer Form vor, die sich für eine Analyse eignet, müssen daher vorverarbeitet werden. Gleichzeitig können grundsätzlich jegliche Daten von Nutzen sein und sollten nicht leichtsinnig verworfen oder erst gar nicht in Erwägung gezogen werden. Die beschriebenen Data-Mining-Projekte demonstrieren, dass diese Aspekte entscheidend sind und einen Großteil des Aufwands ausmachen.
- Data-Mining-Werkzeuge oder Datei-Formate sind nicht kompatibel. Die Arbeit nennt Empfehlungen, womit Data-Mining betrieben werden kann. Es werden verschiedene Kategorien an Werkzeugen identifiziert und Vorschläge für konkrete Werkzeuge gemacht.
- Schwächen der Ergebnisse werden nicht erkannt. Ergebnisse im Data-Mining können nicht vollständig korrekt sein (vgl. [50, S. 752]). Die Data-Mining-Projekte dieser Arbeit zeigen, dass ein wichtiger Teil der Ergebnisse darin besteht, Schwächen aufzuzeigen und mögliche Erweiterungen vorzuschlagen.
- Der Fachbereich Data-Mining ist noch nicht so „erwachsen“ [48] wie die Software-Entwicklung, die z.B. Techniken aus den Ingenieurswissenschaften übernommen hat. Damit verbundene Begriffe wie Konzeption, Tests, Schnittstellen oder Refaktorisierungen werden auch in dieser Arbeit angesprochen.

Die Probleme betreffen sowohl das Entwicklerteam, das die Data-Mining-Ergebnisse realisieren soll als auch den Endnutzer, der von den Ergebnissen profitieren soll. Insbesondere wird die Unzugänglichkeit von Data-Mining für den Endnutzer deutlich (vgl. [21]). Diese Arbeit leistet einen Beitrag dazu, Data-Mining zugänglicher für den Endnutzer zu machen. Sie sucht für Endnutzer und Team nach einem gemeinsamen Kontext, wie mittels Data-Mining Probleme angegangen werden können, damit in Projekten weniger Missverständnisse und Fehler auftreten. Motivation ist eine erfolgreichere und breitere Anwendung von Data-Mining-Techniken.

Ich konzentriere mich in dieser Arbeit auf das Deskriptive Data-Mining, den Entscheidungsträger als Endnutzer und das Team der Entwickler. Um die wesentliche Frage zu beantworten, wird als Methode die Fallstudie genutzt: Wie kann Deskriptives Data-Mining allgemein so durchgeführt werden, dass der Entscheidungsträger es versteht? Fallstudien ermöglichen das Studium realer Data-Mining-Projekte in einem realistischen Szenario (vgl. [71, S.18]).

Dazu ist die Arbeit folgendermaßen aufgebaut: Das zweite Kapitel beschreibt das Design einer Mehrfachfallstudie: Von besonderem Interesse sind demnach die Aufgaben, die normalerweise in Data-Mining-Projekten durchgeführt werden. Die Mehrfachfallstudie kann dabei auf Informa-

tionen aus mehreren Projekten zurückgreifen und hat dadurch ein größeres Potenzial für gültige Verallgemeinerungen (vgl. [71]). Ihr erstes Ergebnis wird im dritten Kapitel behandelt: Ein Lösungsansatz über eine Entscheidungsträger-verständliche Vorgehensweise im Data-Mining. Um die Vorgehensweise praxisnah demonstrieren zu können, wurde sie in zwei Data-Mining-Projekten angewendet. Daher wird im vierten Kapitel das Design zweier Einzelfallstudien zu diesen Projekten beschrieben. Das fünfte Kapitel beschreibt dann das Projekt „Bachelor“: Darin werden die Bachelorstudiengänge der Universität Würzburg mittels Prüfungsdaten bewertet. Das sechste Kapitel beschreibt das Projekt „CaseTrain“: Darin wird der Nutzen eines fallbasierten Lehrsystems anhand von Benutzungsdaten beurteilt. Die Einzelfallstudien werden im siebten Kapitel für das zweite Ergebnis der Mehrfachfallstudie verwendet: Die Beschreibung der Erfahrungen mit der ausgearbeiteten Vorgehensweise. Das achte Kapitel diskutiert die Ergebnisse der Mehrfachfallstudie und gibt einen Ausblick. Abschließend werden im neunten Kapitel die Inhalte der Arbeit zusammengefasst.

2 Grundlagen: Design der Mehrfachfallstudie

In diesem Kapitel werden die Grundlagen zur Durchführung der Mehrfachfallstudie beschrieben, darunter fallen Grundbegriffe, alternative Lösungsansätze in der Literatur, die Forschungsfrage, die relevanten Beobachtungspunkte sowie das Vorgehen.

2.1 Grundbegriffe

Wichtige Grundbegriffe sind insbesondere Data-Mining, Business-Intelligence und Statistik, die im Folgenden kurz behandelt werden sollen.

2.1.1 Data-Mining

Data-Mining ist ein Begriff, der in vielen Zusammenhängen genannt wird und daher nicht so einfach eindeutig zu definieren ist. In dieser Arbeit wird Data-Mining einerseits als essentieller Schritt im „Knowledge Discovery and Data Mining“ (KDDM) [41, S. 2] betrachtet, andererseits mit ihm gleichgesetzt, wenn es im Kontext deutlich wird, was gemeint ist.

Für Data-Mining als Synonym des KDDM halte ich die Definition von Fayyad et al. angemessen, da sie den Fokus auf das Ziel des KDDM rückt: „Der nicht-triviale Prozess zum Identifizieren von validen, unbekannten, potentiell-nützlichen und letztendlich verständlichen Mustern in Daten“ (vgl. [27, S. 6]).

Für Data-Mining als einzelner Schritt im KDDM halte ich folgende Definition für angemessen, da sie den Schwerpunkt auf die Art und Weise rückt, wie das Ziel im letzten Schritt erreicht werden soll, nämlich durch die Anwendung von Techniken – semi- oder vollautomatisch: „Die vom Menschen kontrollierte Anwendung von eigenständigen Techniken“ (vgl. [38, S. 10]) zur Extraktion von Mustern aus Daten. Auch laut Anand und Buchner [3, S. 67f] soll dieser Schritt weitgehend automatisiert werden.

Beide Definitionen verwenden den Begriff Muster. Auch dieser ist nicht eindeutig definiert. Um herauszufinden, was damit gemeint ist, habe ich mir die Aufgaben (bzw. Ziele, Problemtypen oder Funktionen) des Data-Mining in der Literatur angeschaut. Beschreibung, Zusammenfassung und Unterscheidung; Segmentierung oder Clustering; Klassifikation; Prädiktion; Assoziation von

Daten, sind einige der am häufigsten genannten Aufgaben (vgl. [32, S. 21-27]; [17, S. 53-58]; [3, S. 77ff]). Auffallend ist, dass keine Aufzählung der Ziele vollständig und überschneidungsfrei ist. Ein Beispiel für erstere Behauptung: Die Analyse von Geo-, Bild-, Musik- oder Videodaten wird häufig nicht genauer behandelt, obwohl sie spezielle Muster enthalten. Ein Beispiel für Letzteres: Beim Segmentieren handelt es sich um eine Form der Unterscheidung von Daten, angewendet auf den gesamten Datensatz.

Eine vollständige Behandlung des Themas in Form eines Standard-Werks zum Data-Mining ist vermutlich nicht zu realisieren; zu facettenreich und allgemein anwendbar sind die Aufgaben des Data-Mining. Ich beschränke mich daher in dieser Arbeit auf das Beschreiben und Erklären von Daten. Mehrere Werke trennen dieses *Deskriptive Data-Mining* vom *Prädiktiven Data-Mining*, das die Vorhersage mittels Daten bezweckt ([32, S. 21]; [53, S. 98]). Desweiteren setze ich in dieser Arbeit voraus, dass sich jegliche Daten in eine strukturierte Form, als Tabelle mit Texten, Zahlen oder Zeitwerten, bringen lassen.

Ein Muster kann unter dieser Voraussetzung gut als potenziell nützliche Information über die Daten bezeichnet werden. Das Wissen über solche Informationen kann je nach Datenqualität und -interpretation auf die Realität übertragen und dort genutzt werden. Deshalb ist das Muster so wichtig: Die „Erschaffung von Wissen geschieht durch Information, und Information ist nichts anderes als erfasste, abgefragte, aufbereitete und analysierte Daten“ (vgl. [30, S. 171]); sie stellt damit den Nutzen des Deskriptiven Data-Mining dar, die Schaffung von Wissen zur Entscheidungsunterstützung.

Allerdings ist der Schritt von der Beschreibung von Daten zur Erklärung der Daten groß (vgl. [39]). Ein KDDM-Projekt startet meist mit Fragestellungen, die Behauptungen enthalten. Die Daten können diese Behauptungen bestärken bzw. neue Behauptungen ergeben. Nur wenn sorgfältig überprüft und evaluiert, können diese Behauptungen Erklärungen liefern.

2.1.2 Business-Intelligence, Statistik und andere Fachbereiche

Wie Data-Mining an sich, sind auch seine Grenzen zu anderen Fachbereichen nicht eindeutig definiert. Data-Mining wird häufig mit *Business-Intelligence*, *Statistik*, *Maschinen-Lernen* und *Künstliche Intelligenz* in Verbindung gebracht. Inwiefern diese Bereiche fachlich verwandt sind, sich gegenseitig überschneiden oder enthalten, soll nicht Thema dieser Arbeit sein. Fakt ist, dass Data-Mining Techniken aus angrenzenden Fachbereichen nutzen kann, um sein Ziel der Wissensentdeckung zu erreichen.

Anwendungen der Business-Intelligence bieten „Geschäftspersonen“ (vgl. [63, S. 4]) leichten Zugang zu entscheidungsunterstützenden Daten. In diesem Zusammenhang werden häufig ETL – das Extrahieren, Transformieren und Laden von Daten; Data-Warehousing – das abrufbereite Speichern der Daten in relationalen oder multidimensionalen Datenbanken; OLAP –

die interaktive Analyse multidimensionaler Daten; und Reporting – die Endnutzer-aufbereitete Präsentation von Daten genannt (vgl. [28]). Auch im Data-Mining sind diese Techniken nützlich, was in dieser Arbeit demonstriert werden soll.

Laut Hirji [34, S. 92] werden in der Statistik Hypothesen bestätigt oder verworfen, wohingegegen im Data-Mining Informationen entdeckt werden, deren Form im Voraus nicht feststeht. Wenn auch diese Aussage möglicherweise zu pauschal ist, immerhin wird im Data-Mining nach einem bestimmten Muster gesucht, das Schaffen von neuem Wissen aus Daten ist das gemeinsame Ziel von Data-Mining und Statistik; daher wundert es nicht, dass beide Disziplinen Techniken des anderen nutzen. Die Korrelationskoeffizienten aus der Statistik werden beispielsweise im Data-Mining verwendet, um eine Abhängigkeit zwischen Daten zu belegen. Aber auch zum Beschreiben und Zusammenfassen von Daten bietet die Statistik bewährte Ansätze, z.B. Tabellen oder Diagramme. Im Gegenzug kann die Statistik Techniken aus dem Data-Mining nutzen, um z.B. textuelle Daten zu analysieren oder informelles Hintergrundwissen in Analysen einfließen zu lassen (vgl. [3, S. 38f]).

Viele Data-Mining-Algorithmen finden ihren Ursprung im Maschinellen Lernen (vgl. [3, S. 41]), darunter Neuronale Netze, Genetische Algorithmen oder Fallbasiertes Schließen. Entwicklungen aus dem gesamten Feld der Künstliche Intelligenz können Data-Mining zum Vorteil sein, etwa Suchalgorithmen oder Wissensrepräsentationen.

2.2 Lösungsansätze in der Literatur

Die Ergebnisse der Mehrfachfallstudie müssen sich an „rivalisierenden Erklärungen“ (vgl. [71, S. 34]) messen. Drei Informationsquellen könnten Data-Mining für *Endnutzer* und *Data-Miner* thematisieren: Data-Mining-Prozessmodelle, -Methodologien sowie -Fallstudien. Im Folgenden beschreibe ich, inwiefern sie den Endnutzer und Data-Miner berücksichtigen und wie sie zum Ergebnis meiner Arbeit beitragen.

2.2.1 Prozessmodelle

Eine direkte Anwendung von Data-Mining-Algorithmen kann zu sinnlosen Mustern führen (vgl. [41, S. 2f]. Der Zweck eines Prozessmodells besteht darin, dies zu vermeiden.

Einen Überblick über bekannte Data-Mining-Prozessmodelle geben Kurgan und Musilek [41]. Das am häufigsten verwendete ist der Industriestandard „CRISP-DM“, bestehend aus den übergeordneten Phasen „Business Understanding“, „Data Understanding“, „Data Preparation“, „Modelling“, „Evaluation“ und „Deployment“. Diese beschreiben die Aufgaben eines Data-Mining-Vorhabens sowie deren Eingaben und Ausgaben. Für Marban et al. [48] weist dabei ein

gutes Prozessmodell folgende Eigenschaften auf: Effektivität, Wartbarkeit, Vorhersagbarkeit, Wiederholbarkeit, Qualität, Verbesserbarkeit, Verfolgbarkeit.

Prozessmodelle bieten damit erstens einen Nutzen für den Endnutzer: Sie beschreiben das Vorgehen zum Data-Mining abstrakt, mit Betonung auf die Eingaben und Ausgaben, unter Verzicht auf Implementierungsdetails; für den Endnutzer die Möglichkeit, um auch ohne spezielles Hintergrundwissen zu verstehen, was in einem Data-Mining-Vorhaben gemacht wird. Aufgaben werden hierarchisch in Phasen eingeteilt. Das Ende einer jeden Phase bietet einen Meilenstein, anhand eines jeden der Endnutzer den aktuellen Status des Vorhabens verfolgen kann. Wenn ein Prozessmodell über mehrere Vorhaben hinweg genutzt wird, kann der Endnutzer die einzelnen Vorhaben miteinander vergleichen und Erfahrungen aus einem Vorhaben auf ein anderes übertragen.

Zweitens bieten sie einen Nutzen für diejenigen, die Data-Mining betreiben: In Prozessmodelle sind häufig Erfahrungen aus vielen Data-Mining-Vorhaben eingeflossen. Es ermöglicht dem Data-Miner, die richtigen Ergebnisse möglichst effektiv zu erreichen, zudem mit einer hohen Qualtität.

Diese Vorteile gelten allerdings ausschließlich unter einer Voraussetzung: Der Data-Miner kennt nicht nur die Aufgaben aus dem Prozessmodell, sondern weiß auch, wie diese konkret durchgeführt werden können; denn dies beschreibt ein Prozessmodell nicht.

Und auch dem Endnutzer fehlen in einem Prozessmodell wichtige Informationen. Er erfährt nicht, wie Eingaben identifiziert und gerechtfertigt sowie Ausgaben zunächst beschrieben und nach Umsetzung bewertet werden können – Kenntnisse, die nötig sind, um ein Data-Mining-Vorhaben zu initiieren.

Als Rahmen der Data-Mining-Vorhaben, die ich während der Mehrfachfallstudie betrachte, werde ich ein Prozessmodell auswählen. Dieses werde ich mittels Erfahrungen aus Data-Mining-Vorhaben konkretisieren und einen eigenen Lösungsansatz entwickeln; so nutze ich in meiner Arbeit die Vorteile eines Prozessmodells, vermeide jedoch die Nachteile.

2.2.2 Methodologien

Ein Prozessmodell beschreibt allgemeine Aufgaben. Eine Methodologie kann laut Marban et al. [48] als konkrete Anwendung eines Prozessmodells verstanden werden, die neben Aufgaben, deren Eingaben und Ausgaben auch beschreibt, wie die Aufgaben durchgeführt werden können.

Eine klassische Methodologie bietet „Catalyst“ [53]. Sie beschreibt in zwei Teilen eine *Schritt-für-Schritt*-Anleitung zum Vorgehen im Data-Mining: Im „Modeling“ [53, S. 91-272] werden die Randbedingungen, z.B. das behandelte Problem und die beteiligten Personen, identifiziert sowie eine Repräsentation der Wirklichkeit in einem Modell abgebildet. Im „Data Mining“ [53, S. 275-529] gilt es, die Güte dieses Modells mit harten Fakten, den Daten, zu belegen. Pyles Methodologie enthält Empfehlungen, die auch zum jetzigen Zeitpunkt noch gelten; sie habe ich in

meiner Arbeit berücksichtigt. Meinens Erachtens ist die Methodologie jedoch teilweise veraltet. Techniken aus der Business-Intelligence, wie ETL, OLAP, Data-Warehouse oder Reporting werden nicht behandelt, obwohl sie den Data-Mining-Prozess für Endnutzer und Data-Miner deutlich erleichtern können. Keines der genannten Werkzeuge gehört zur Sparte der *Open-Source*-Werkzeuge, die in den letzten 5-10 Jahren am Markt gereift ist. Marban et al. [48] nennen weitere Data-Mining-Methodologien. Diese sind entweder auch veraltet oder beschränken sich auf einzelne Bereiche oder Werkzeuge des Data-Mining.

Ich werde meinen Lösungsansatz in Form einer Methodologie entwickeln und zeigen, wie gut sie sich auf beliebige Data-Mining-Vorhaben anwenden lässt.

2.2.3 Data-Mining-Fallstudien

Anders als Prozessmodelle und Methodologien, die allgemein gültig sein sollen, besitzen Fallstudien die Möglichkeit, Data-Mining-Projekte bis ins Detail zu beschreiben und das Verständnis von Data-Miner und Endnutzer zu berücksichtigen. Data-Mining-Fallstudien lassen sich in drei Gruppen aufteilen:

- Fallstudien zur Demonstration einer Technik oder eines Werkzeugs zum Data-Mining.
- Fallstudien zum Vorstellen eines Ergebnis, das mittels Data-Mining erreicht wurde.
- Fallstudien zum Demonstrieren eines Prozessmodells oder einer Methodologie.

Erstgenannte Fallstudien dienen meist der Evaluation von speziell entwickelten Techniken oder Werkzeugen, die einen Teil des Data-Mining betreffen. Solche Evaluationen gehören zum guten Ton eines Forschungsergebnisses. Beispiele finden sich in der Literatur daher häufig, z.B. das Vorstellen der „Conjoint Analysis“ [20], der Abweichungsentdeckung als vielversprechende Technik (vgl. [16]) oder einem Werkzeug zum verteilten Data-Mining (vgl. [65]). Auch als „Step-by-Step“-Anleitungen von Werkzeugen (vgl. [50, S. 373]) werden sie verwendet. Außerdem gehören hierzu jegliche Fallstudien von Werkzeug-Herstellern wie JasperSoft[1] oder Talend[2]. Daher sind in diesen Fallstudien das behandelte Problem, die Anwendung der Neuentwicklung und das erreichte Ergebnis des Data-Mining von Bedeutung. Diese Aspekte werden in solchen Fallstudien ausführlich behandelt. Sachverhalte, die nicht dem Bestätigen von Anforderungen sowie dem Nennen von Vorteilen der Technik oder des Werkzeugs dienen, spielen eine untergeordnete Rolle. Zur Vollständigkeit wird zwar meist das gesamte Data-Mining-Vorhaben überrissen, diesbezüglich aber nur oberflächlich. Die Data-Mining-Vorhaben werden häufig nur als pseudo-reale Umgebung für die Neuentwicklung verwendet, weshalb Informationen zum Anfang des Vorhabens, z.B. wie es initialisiert wurde, zur konkreten Durchführung des Vorhabens, z.B.

[1] Case Studies, http://www.jaspersoft.com/case-studies, Dezember 2009
[2] Talend Customers, http://www.talend.com/open-source-provider/reference.php, Dezember 2009

wie es dokumentiert wurde, und zum Abschluss des Vorhabens, z.B. wie es dem Auftraggeber präsentiert wurde, fehlen. Von Herstellern werden gerne ausschließlich Vorteile ihrer Entwicklungen bzw. der Zusammenarbeit mit Kunden beschrieben, Schwierigkeiten bei der Umsetzung bzw. Hinweise für spätere Vorhaben fehlen. Daher sind diese Fallstudien nicht dazu geeignet, das Verständnis von Endnutzer und Data-Miner zu behandeln.

Die zweite Gruppe an Fallstudien befasst sich mit dem Ergebnis eines Data-Mining-Vorhabens. Hier spielen das Problem und das Ergebnis die größte Rolle. Deshalb werden auch der Hintergrund und die Motivation des Data-Mining-Vorhabens beschrieben. Außerdem, wie das Ergebnis genutzt, z.B. dem Endnutzer vorgestellt wird. Diese Fallstudien zeigen, welches Potenzial Data-Mining besitzt, um z.B. Steuerhinterzieher (vgl. [22]) oder Triebtäter (vgl. [1]) zu entlarven, den Zustand herzkranker Patienten zu überwachen (vgl. [58]) oder Verkaufszahlen zu maximieren (vgl. [43]). Von weniger großer Bedeutung erscheint jedoch alles, das zwischen der Beschreibung des Problems und der Präsentation der Lösung passiert ist. Hier wird nur oberflächlich die verwendete Technik oder das Werkzeug genannt, das konkrete Vorgehen wird vernachlässigt. Wenn beschrieben, lässt sich das Vorgehen nur auf das behandelte Problem anwenden und nicht auf weitere Anwendungsbereiche übertragen. Abgesehen von der Problem- und Lösungsbeschreibung wird somit das Verständnis von Endnuter und Data-Miner nicht berücksichtigt.

Die dritte Gruppe an Fallstudien behandelt das allgemeine Vorgehen in einem Data-Mining-Vorhaben ohne Konzentration auf eine spezielle Technik, Werkzeug oder Problem. Dabei verwenden sie häufig explizit ein bestimmtes Prozessmodell, z.B. „CRISP-DM“ (vgl. [49]), das von Anand und Buchner (vgl. [15]) oder von Cios (vgl. [40]). Der Schwerpunkt dieser Fallstudien liegt darin, ein allgemeines Vorgehen zu demonstrieren, das auf weitere Probleme angewendet werden kann, z.B. ein “predictive model“ [14] oder „Business Intelligence“ [69]. In diesem Zusammenhang erwähnen sie häufig Erfahrungen, z.B. als Lessons-Learned ([39]; [42]; [36]), die hilfreiche Hinweise in weiteren Vorhaben liefern. Deshalb werde ich diese Fallstudien als hauptsächliche Informationsquelle verwenden, um einen Lösungsansatz zu entwickeln. Dennoch behandeln sie den Data-Miner und Endnutzer nicht ausreichend, sondern sind entweder zu technisch ausgelegt, um als Endnutzer-verständlich zu gelten oder zu oberflächlich, um einen fundierten Einblick in den Prozess des Data-Mining zu bieten.

Drei Punkte werden in keiner der Fallstudiengruppen ausreichend behandelt: Einerseits die Daten. Häufig wird von perfekt aufbereiteten Rohdaten ausgegangen, von denen aus das Vorhaben beschrieben wird; wie die Daten erfasst und dem Data-Miner übergeben wurden, bleibt – wohl auch aus Vorgaben von Verschwiegenheitserklärungen unbekannt. Zweitens die Werkzeuge. Die genaue Anleitung eines Werkzeugs nützt wenig, wenn im idealisierten Vorhaben unverständlich bleibt, mit welchen Kriterien das Werkzeug ausgewählt worden ist. Und außerdem der Endnutzer. Wenn er behandelt wird, dann nur zum Ende eines Vorhabens, wenn es gilt, die Ergebnisse zu präsentieren. Die Kommunikation mit dem Data-Miner bleibt verborgen.

Insgesamt wird das Potenzial von Fallstudien zum Behandeln des Verständnis von Endnutzer und Data-Miner nicht ausgenutzt. Ich werde die Fallstudienmethodik in dreierlei Hinsicht verwenden. Zunächst als Informationsgrundlage für die Hypothese der Mehrfachfallstudie. Außerdem zum Demonstrieren der Methodologie. Und schließlich zum Bewerten der Methodologie.

2.3 Forschungsfrage und behandelte Rollen

Im Folgenden wird zunächst die Forschungsfrage gestellt und der angesprochene Leser der Mehrfachfallstudie genannt. Anschließend wird die Sicht beschrieben, aus der die Forschungsfrage gestellt wird.

Die Forschungsfrage lautet: „Wie kann Deskriptives Data-Mining allgemein so durchgeführt werden, dass der Entscheidungsträger es versteht?“ Das Ergebnis soll eine möglichst allgemeine, Entscheidungsträger-verständliche Methodologie sein.

Die Fallstudienmethodik ist damit als Forschungsmethode besser geeignet als z.B. ein Experiment oder eine Umfrage, weil die Forschungsfrage untersucht, wie in bestimmten Situationen vorzugehen ist (vgl. [71, S. 8]).

Die Methodologie behandelt den Endnutzer, soll jedoch an Personen gerichtet sein, die Data-Mining selbst betreiben wollen und einen technischen Hintergrund besitzen. Sie erhalten durch die Mehrfachfallstudie Hinweise, um erfolgreiches und für den Endnutzer verständliches Data-Mining allgemein zu betreiben. Erst in einem späteren Teil der Arbeit werden Einzelfallstudien beschrieben, die auch Personen ansprechen sollen, die als Endnutzer von Data-Mining profitieren wollen.

Damit der Leser weiß, aus welchem Blickwinkel die Forschungsfrage gestellt und Data-Mining in den folgenden Kapiteln beschrieben wird, soll nun die Ausgangssituation festgelegt werden. Sie erklärt, welche Rollen die Data-Miners und der Endnutzer einnehmen und welche Aspekte eine Bedeutung besitzen, um die Forschungsfrage zu beantworten.

Da Deskriptives Data-Mining hier möglichst allgemein angewendet werden soll, wird es im Folgenden aus Sicht einer Organisation beschrieben, in der Data-Mining-Techniken von internen oder externen Mitarbeitern sowohl kurz- als auch langfristig angewendet werden sollen. In einem einzelnen Data-Mining-Vorhaben werden in einer „einzigartigen Unternehmung eine Menge an Ergebnissen innerhalb festgelegter Zeit, Budget und Qualität geliefert“ (vgl. [66, S. 2]); es handelt sich also stets um ein Projekt. Der grobe Rahmen eines solchen Projekts, einer Deskriptiven „Data-Mining-Anwendung“ [11], sieht folgendermaßen aus: Ein Entwicklerteam aus Data-Mining-, Daten- und Domänen-Experten wendet Techniken und Werkzeuge zur Datenanalyse an, um relevante Informationen bezüglich eines Problems zu erstellen; diese werden für den Endnutzer so aufbereitet, dass er sie in realen Situationen als Entscheidungshilfe zur Lösung des

Problems nutzen kann[3].

2.3.1 Entscheidungsträger

Der Nutzen des Deskriptiven Data-Mining ist die Entscheidungsunterstützung, der Endnutzer demnach der Entscheidungsträger. In der Literatur treten eine Reihe von Data-Mining-Beteiligten auf, die dem Entscheidungsträger ähnlich sind, so z.B. die sog. „Stakeholder“ [53, S. 165-169] eines Projekts. Darunter werden Personen verstanden, die am Ergebnis des Projekts interessiert sind bzw. sein können. Gründe dafür gibt es viele, Pyle nennt fünf: das Problem – Personen, die es am eigenen Leib erfahren; das Budget – Personen, die es zur Verfügung stellen; der Auftrag – Personen, die ihn zum Start des Projekts stellen; die Vorteile – Personen, denen das Projekt diese bescheren kann; und die Gunst – Personen, deren Gunst des Vorgesetzten vom Verlauf des Projekts abhängt. Ich bin der Meinung, dass diese Stakeholder nicht speziell auf Data-Mining-Projekte zutreffen, sondern in allen Projekten auftreten. Daher fasse ich sie zur Vereinfachung in der Person des Entscheidungsträgers zusammen, der nun wie folgt charakterisiert wird:

Für ein erfolgreiches Data-Mining-Projekt ist Engagement der Beteiligten nötig. Der Entscheidungsträger ist der direkte Profiteur des Deskriptiven Data-Mining, tritt daher als Auftraggeber des Projekts auf und trägt als Führungskraft innerhalb der Organisation die letztendliche Verantwortung für dessen Ergebnis. Er verwaltet eigene Ressourcen, muss daher über benötigtes Personal, Budget, Software, Hardware o.ä. im Projekt unterrichtet sein. Genauso auch über die benötigte Zeit für Ergebnisse.

Der Entscheidungsträger stammt aus einem beliebigen Geschäftsbereich, besitzt daher selten einen technischen Hintergrund und ist oft weder mit Data-Mining noch mit fachlich angrenzenden Bereichen wie z.B. Statistik, Business-Intelligence, Künstliche Intelligenz, Maschinelles Lernen oder Datenbanksystemen vertraut. Er muss über die Komplexität oder nötige Erfahrung bei der Anwendung von Techniken oder Werkzeugen Bescheid wissen, da sie Risiken für den erfolgreichen Abschluss des Projekts beinhalten, z.B. eines zeitraubenden Software-Fehlers in einem projektentscheidenden Werkzeug.

Auch die Art und Weise, wie die Ergebnisse präsentiert werden können, hängt vom Entscheidungsträger ab ([53, S. 218]). Jede Form der Wissensrepräsentation ist unterschiedlich stark verständlich, was Anand und Buchner [3, S. 51] mit „perspicuity“ ausdrücken. Auch nach *Ockham's Razor* (vgl. [50, S. 246]) sollte immer die einfachste Form bevorzugt werden. Eine einzelne Regel ist demnach verständlicher als ein ganzer Entscheidungsbaum. Auch möchte ein Entschei-

[3] Wenn auch in dieser Arbeit aus Gründen der Bequemlichkeit durchgängig von Endnutzern bzw. Entscheidungsträgern und Experten gesprochen wird, so kann es sich bei ihnen selbstverständlich auch um weibliche Personen handeln.

dungsträger nicht komplizierte statistische Konzepte verstehen müssen (vgl. [39]). Dies ist insbesondere im Deskriptiven Data-Mining von Bedeutung, in dem Beschreibungen und Erklärungen von Daten geliefert werden sollen.

Und letztendlich ist wichtig: Ein Entscheidungsträger hat wenig Zeit. Er möchte nur soviel lesen und tun, wie unbedingt nötig ist. So interessiert ihn nicht, wie und womit etwas getan wird, wenn er darauf keinen Einfluss nehmen kann. Ihn interessieren einerseits die nötigen Eingaben, in Form von Ressourcen und Daten. Diese stellt er zur Verfügung. Außerdem, inwiefern das Problem durch Ergebnisse behandelt wird. Diese Ergebnisse nimmt er als Kunde des Teams zum Projektende ab.

2.3.2 Entwicklerteam

Laut Anand und Buchner (vgl. [3, S. 108]) sind die Expertise zum Data-Mining, zu den Daten und zum Problembereich eine Grundvoraussetzung für das Team. Ich personifiziere sie der Einfachheit halber zu drei Personengruppen, die wie folgt charakterisiert werden:

2.3.2.1 Data-Mining-Experten

Die Aufgabe eines Data-Mining-Experten lässt sich als „wissensintensive Interaktion zwischen einem Menschen und einer Datenbank, unterstützt durch eine Menge an heterogenen Werkzeugen“ (vgl. [11]) beschreiben.

Der Data-Mining-Experte besitzt demnach Hintergrundwissen zum gesamten Data-Mining-Prozess und verwendet eine eigene Methodik oder eine übernommene Methodologie, um die einzelnen Aufgaben durchzuführen. Da er häufig mehrere verschiedene Werkzeuge einzusetzen hat, muss er technisch versiert sein.

2.3.2.2 Domänen-Experten

Das „Application Domain Understanding“ [41] ist ein wichtiger Schritt im Data-Mining-Prozess. Es betont das Wissen zum Fachbereich, in dem das zu behandelnde Problem auftritt. Ein Domänen-Experte hat solches Hintergrundwissen meist auf Grund einer langen Tätigkeit in diesem Bereich erworben. Daher kann nicht vorausgesetzt werden, dass er gleichzeitig einen technischen Hintergrund besitzt.

2.3.2.3 Daten-Experten

Das „Data Understanding“ [41] ist ein wichtiger Schritt im Data-Mining-Prozess. Gäbe es für die verwendeten Rohdaten eine allgemeingültige und verständliche Dokumentation, z.B. in einem „Data Dictionary“ (vgl. [11]), wäre ein Daten-Experte nicht nötig. Der Daten-Experte besitzt umfassendes Verständnis zu den Rohdaten, auch um heterogene Daten in einen gemeinsamen Kontext zu setzen. Er weiß, z.B. als Datenbank-Administrator, wie die Daten erfasst, gespeichert und abgerufen werden. Die Aufgabe des Daten-Experten ist es, diese Informationen im Team verfügbar zu machen.

Jeder dieser Beteiligten kann von innerhalb oder außerhalb der Organisation kommen, abhängig vom behandelten Problem und den vorhandenen Ressourcen. So kann man sich auch einen externen Experten vorstellen, wenn es gilt externe Daten zu analysieren. In der Literatur finden sich viele Synonyme dieser Beteiligten eines Data-Mining-Projekts, z.B. „Business Analyst”, „Data Analyst“, „Data Engineer“, „Knowledge Engineer“ oder „Strategic Planner“ (vgl. [35]). Dabei handelt es sich hauptsächlich um Ergänzungen, um in größeren Projekten einzelne Aufgaben an Personen mit speziellen Erfahrungen zu delegieren.

Während eines Data-Mining-Projekts fließt das Wissen der Data-Mining-, Domänen- und Daten-Experten über den gesamten Projektverlauf in den Prozess ein. Ein Teammitglied tritt gegenüber dem Kunden, dem Entscheidungsträger, als Projektleiter auf. Thema der Arbeit ist nicht die Kommunikation im Team, wie z.B. in der Arbeit von Hofmann und Tierney [35], sondern ausschließlich die Kommunikation zwischen dem gesamten Team und dem Entscheidungsträger.

2.4 Beobachtungspunkte: Data-Mining-Projekte und ihre Prozesse

Im Folgenden wird beschrieben, was während der Mehrfachfallstudie als Fall gilt und welche Aspekte eines Falls besonders interessant sind.

Als Fall gilt in dieser Mehrfachfallstudie ein konkretes Data-Mining-Projekt. In den einzelnen Fällen sind Beobachtungspunkte, sog. „Analyseeinheiten“ eingebettet (vgl. [71, S. 46]). Sie erklären, welche Aspekte eines Falls von besonderem Interesse für die Mehrfachfallstudie sind: Es sind die einzelnen Aufgaben eines Prozessmodells, welches für diese Arbeit aus verschiedenen Prozessmodellen – insbesondere dem Industriestandard „CRISP-DM“ [17] und eines Modells von Marban et al. [48], das ihn erweitert – zusammengestellt wurde. Abbildung 2.1 beschreibt das Design der Fallstudie.

Im Folgenden wird das Prozessmodell erläutert. Darin werden nur Aufgaben beschrieben, die für Deskriptives Data-Mining von Bedeutung sind.

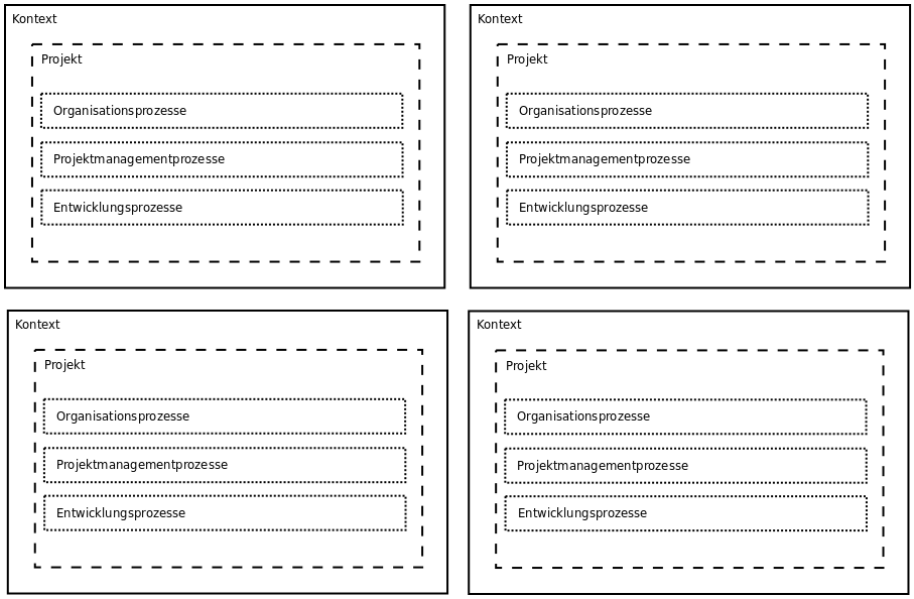


Abb. 2.1: Analyseeinheiten der Mehrfachfallstudie (vgl. [71, S. 46])

2.4.1 Organisationsprozesse

Anforderungen an das Data-Mining haben Dimensionen angenommen, die die gesamte Organisation betreffen (vgl. [48]), innerhalb der Data-Mining betrieben werden soll. Aufgaben, die sich daraus implizieren, sind Thema dieses Abschnitts.

2.4.1.1 Wissensmanagement

Zunächst muss sichergestellt werden, dass erworbene Kenntnisse in Data-Mining Projekten innerhalb der Organisation verbreitet werden können (vgl. [48]) und für andere Projekte als Informationsquelle zur Verfügung stehen. Dieser Schritt ist wichtig, da die Erfahrung des Data-Mining-Teams entscheidend für den Projekterfolg ist (vgl. [23]). Da jedoch nicht jeder die Erfahrung aus vielen Projekten besitzen kann, spielt das Wissensmanagement eine große Rolle. Erfahrungen oder Lessons-Learned, z.B. zu Techniken und Werkzeugen, besitzen im Data-Mining laut Marban et al. ein hohes Potenzial, wiederverwendet zu werden. Denn Aufgaben wiederholen sich hier immer wieder, nicht nur innerhalb eines Vorhabens bei dem eine Aufgabe nach Identifizieren eines Fehlers wiederholt werden muss, sondern auch zwischen mehreren Vorhaben, die Gemeinsamkeiten aufweisen. Ein solches Wissensmanagement ermöglicht auch das effiziente Anlernen von Mitarbeitern – ein weiteres wichtiges Ziel, um langfristig Data-Mining-Projekte durchzuführen (vgl. [48]).

2.4.1.2 Infrastruktur

Desweiteren soll in der Organisation eine Infrastruktur erreicht werden, die das Data-Mining bestmöglich fördert. Was Marban et al. [48] nur nennen, wird von Anand und Buchner [3, S. 110] näher beschrieben: Besonders wichtig ist der geregelte Zugriff auf die nötigen Daten, idealerweise über einen Server mit Datenbank. Die sichere Speicherung, insbesondere von personenbezogenen oder ähnlichen sensiblen Daten muss sichergestellt sein.

2.4.2 Projektmanagementprozesse

Für ein erfolgreiches Data-Mining-Projekt müssen Techniken aus dem Projektmanagement ihre Anwendung finden, im Modell von Marban et al. als „Projektmanagementprozesse“ und „integrale Prozesse“ [48] beschrieben.

2.4.2.1 Planung

Es sind verschiedene Lebenszyklen möglich, um Data-Mining-Ergebnisse zu entwickeln (vgl. [48]) – einer sollte ausgewählt werden. Ein Data-Mining-Lebenszyklus beschreibt nicht nur Aufgaben, sondern auch eine Reihenfolge, in der diese durchgeführt werden sollen. Kurgan und Musilek [41] fassen bekannte Prozessmodelle in einem „generischen Modell“ aus mehreren Phasen zusammen. Es impliziert die *Wasserfallmethode mit Backtracking*, bei der mehrere Aufgaben hintereinander auszuführen sind, man jedoch jederzeit zu einer vorhergehenden Aufgabe zurückkehren kann. Auch nachdem die letzte Aufgabe durchgeführt worden ist, können neue Erkenntnisse (z.B. *Lessons-Learned*) weitere Ergebnisse erforderlich machen und den Lebenszyklus von Vorne starten lassen.

Der Lebenszyklus wird mit konkreten Arbeitspaketen instantiiert und in einem Projektplan festgehalten. Dieser ist vorläufig, nach jeder Phase muss er aktualisiert werden (vgl. [17, S. 14]).

In dem Zusammenhang werden auch, unter Berücksichtigung der vorhandenen, die nötigen Ressourcen abgeschätzt und der voraussichtliche Bedarf des gesamten Projekts genannt. Metriken sind zu bestimmen, um den aktuellen Status sowie letztendlich den Erfolg des Projekts messbar zu machen. Desweiteren sind Erfolgskriterien im Bezug auf die Organisationsziele und die Data-Mining-Ziele zu bestimmen.

Aber nicht nur der Lebenszyklus, sondern der gesamte Data-Mining-Prozess ist zu planen, inklusive der Organisations- und Projektmanagementprozesse. Im Plan werden auch Annahmen und Risiken berücksichtigt.

2.4.2.2 Dokumentation

Unter dem Begriff der Dokumentation fasse ich die Dokumentation, das Dokumentenmanagement und das Änderungsmanagement zusammen; laut Marban et al. [48] sind diese in Data-Mining-Projekten besonders nötig, weil in einem solchen Projekt jeweils eine große Menge an Informationen erzeugt werden.

Von Bedeutung für das Data-Mining ist beispielsweise die Benennung und Speicherung von Dateien oder das Veranlassen von Projektänderungen, z.B. Anforderungen oder Ressourcen. Das Data-Mining-Team kommuniziert regelmäßig und wird über Änderungen in Kenntnis gesetzt, der Entscheidungsträger nur bei relevanten Änderungen einbezogen. Außerdem ist das Erstellen und Nutzen der Dokumentation zu einem Projekt wichtig. Auch wenn eine Lösung, ähnlich wie in der Software-Entwicklung, später, ggf. auch von anderen Personen, gewartet und erweitert werden soll, muss sie ausreichend dokumentiert sein.

2.4.2.3 Evaluation

Ein wichtiger Teil der Projektmanagement-Prozesse ist die Evaluation; häufig genannte Aufgaben sind:

- Fehler in der Entwicklung erkennen.
- Evaluieren, ob die Ziele des Projekts erreicht werden.
- Bewerten des Projektverlaufs, z.B. durch Metriken, Meilensteine.

Die Evaluation geschieht idealerweise durch Personen, die nicht an der Entwicklung der Ergebnisse beteiligt sind (vgl. [23]).

Wenn Data-Mining-Expertise eingekauft oder verkauft werden soll, ist auch das in den Projektmanagementprozessen zu organisieren. Weitere Aspekte sind in kommerziellen Projekten von Bedeutung, z.B. Verträge oder Verschwiegenheitserklärungen. Für Informationen zu diesen Themen wird speziell für Business-Intelligence-Projekte auf Atre und Moss [63] bzw. auf allgemeine Literatur zum Projektmanagement verwiesen (vgl. [62]; [66]).

2.4.3 Entwicklungsprozesse

Hier sind die Aufgaben enthalten, die die Ziele des Data-Mining-Projekts erreichen sollen.

2.4.3.1 Vorentwicklungsprozesse vor Start des Data-Mining

Es müssen die beteiligten Personen identifiziert werden, also der Entscheidungsträger sowie die Data-Mining-, Daten- und Domänen-Experten (vgl. [3, S. 108]). Ein Kommunikationskanal in natürlicher Sprache wird zwischen diesen aufgebaut (vgl. [13]). Dazu gehört es laut Chapman et al. [17] auch, ein gemeinsames Verständnis von wichtigen Begriffen zu erarbeiten.

Es müssen die Idee, der Zweck und die Ziele des Projekts ausgearbeitet werden (vgl. [48]). Laut Brachman und Anand [11] ist diese „Task Discovery“ sogar dann nötig, wenn der Entscheidungsträger das Problem genau beschreiben kann. Desweiteren wird in der „Data Discovery“ [11] die Qualität der Daten festgestellt, die dem Team zur Verfügung stehen – die sog. Rohdaten.

Es muss die aktuelle Situation im Bezug auf Daten, Ressourcen, Einschränkungen, Annahmen und Risiken bewertet werden.

Ein Data-Mining-Ansatz zur Lösung des identifizierten Problems wird herausgearbeitet. Dazu werden einerseits die Ziele aus Sicht der Organisation, andererseits konkrete Data-Mining-Ziele festgelegt. Desweiteren werden Kosten und Vorteile genannt.

Schließlich ist vor dem Start des Projekts eine Bewertung von Werkzeugen und Techniken nötig, da sie das gesamte Projekt beeinflussen kann.

2.4.3.2 Entwicklungsprozesse während des Data-Mining

Alle Entwicklungsprozesse zwischen dem Kickoff des Projekts und der Abnahme der Ergebnisse durch den Entscheidungsträger, also nachdem das Projekt offiziell vom Entscheidungsträger beauftragt wurde.

Zur Einteilung dieses Prozesses in einzelne Aufgabenbereiche wird der CRISP-DM-Standard [17] verwendet. In welcher Reihenfolge diese Aufgaben durchgeführt werden, gibt der Lebenszyklus vor und ist abhängig vom konkreten Projekt. Keiner der Aufgabenbereiche kann mit Sicherheit vor dem Ende eines Projekts abgeschlossen werden.

Business-Understanding bezieht sich darauf, ein Verständnis zum Problembereich aufzubauen. Im Rahmen der Zielbeschreibung, auch in Machbarkeitsstudien, kann das Team unmöglich alles Wissen erwerben. So können auch während der Entwicklung Änderungen an den Zielen entstehen.

Data-Understanding bedeutet das Verstehen der zur Verfügung gestellten Daten. Damit ist die Sammlung, das Kopieren und das zugreifbare Speichern von Rohdaten gemeint. Außerdem,

Daten zu explorieren und zu beschreiben. Aufgabenziel ist es, die nötige Datenqualität im Bezug auf Projektziele sicherzustellen.

Data-Preparation bezweckt die Vorbereitung der Daten. Folgende Aufgaben werden allgemein darunter gefasst:

- *Selektion* beschreibt die begründete Auswahl von Daten, die zur Bearbeitung des Problems nötig sind.
- *Säuberung* beschreibt das Filtern von Daten, die zur Lösung nicht beitragen können, z.B. fehlende oder falsche Werte.
- *Konstruktion* beschreibt das Erstellen von abgeleiteten Daten aus Rohdaten.
- *Transformation* beschreibt das Abändern der Daten für eine bessere Analyse.
- *Integration* beschreibt das Verbinden von heterogenen Daten.

Data-Mining befasst sich schließlich mit der tatsächlichen Wissensentdeckung. Darunter fallen insbesondere die geeignete Auswahl und Parametrisierung von Algorithmen, die auf den vorbereiteten Daten ausgeführt werden. Im Deskriptiven Data-Mining gilt es, Muster zu entdecken, die die Daten beschreiben und erklären. Die Resultate der Algorithmen sind darauf zu überprüfen, ob nützliches Wissen geschaffen wurde. Die Frage ist, ob die Muster die gesetzten Ziele behandeln und sich auf die Wirklichkeit übertragen lassen.

2.4.3.3 Nachentwicklungsprozesse nach Abschluss des Data-Mining

Die Ergebnisse des Data-Mining-Projekts müssen dem Entscheidungsträger übergeben werden, und zwar in der Form, dass er sie nutzen kann, z.B. Formalisierung in einem Expertensystem. Diese Ergebnisse im Bezug auf das behandelte Problem sollen durch zusätzliche Informationen komplettiert werden, z.B. zum Vorgehen, wie diese erreicht wurden. Zum „Deployment“ [17] gehört es auch, Empfehlungen und Unterstützung für die langfristige Nutzung anzubieten. Zum „Support“ gehört es auch, sicherzustellen, dass der Entscheidungsträger das Wissen richtig interpretiert und anwendet. Außerdem Möglichkeiten, um das entdeckte Wissen zu aktualisieren oder als veraltet zu kennzeichnen.

Abbildung 2.2 zeigt eine Zusammenfassung des Prozessmodells.

2.5 Vorgehen in der Mehrfachfallstudie

Im Folgenden wird beschrieben, wie vorgegangen wird, um die Forschungsfrage zu behandeln.

Abb. 2.2: Verwendetes Prozessmodell

2.5.1 Hypothese

In der Mehrfachfallstudie werden verschiedene Data-Mining-Projekte betrachtet. Einerseits Projekte, an denen der Autor dieser Arbeit selbst beteiligt war und Erfahrungen gesammelt hat, darunter eines zur Untersuchung von Erfolgsfaktoren bei Mittelohroperationen und eines zur Ursachenuntersuchung von Produktionsausfällen. Die Hauptinformationsquelle zu diesen Fällen sind die Dokumentation und die erstellten Dokumente aus den Projekten. Des Weiteren wurden Projekte einbezogen, über die sich Fallstudien in der Literatur finden. Dazu wurden insbesondere solche Fallstudien berücksichtigt, die den Prozess des Data-Mining näher beschreiben, siehe Abschnitt 2.2.3. Die Inhalte der Fallstudien stellen gleichzeitig die Hauptinformation zu diesen Fällen dar.

Die Betrachtung der Fälle bezüglich der Beobachtungspunkte aus Abschnitt 2.4 haben ein erstes Ergebnis dieser Mehrfachfallstudie ergeben: Eine Methodologie zum Entscheidungsträgerverständlichen Data-Mining. Sie soll im nächsten Kapitel als Hypothese für die Mehrfachfallstu-

die beschrieben und als Entscheidungsträger-verständlich begründet werden. Die Behauptung besteht darin, dass diese Methodologie verallgemeinert und auf beliebige Data-Mining-Projekte erfolgreich angewendet werden kann.

2.5.2 Einzelfallstudien

Um diese Behauptung zu bestärken, wurden zwei Data-Mining-Projekte mittels der Methodologie durchgeführt, eines über Bachelorstudiengänge und eines über ein fallbasiertes Lehrsystem. Auch diese Projekte gelten als Fälle der Mehrfachfallstudie. Die Hauptinformationsquelle zu ihnen besteht aus zwei Einzelfallstudien. Diese sind Teil der Arbeit und sollen dem Leser die Methodologie anhand der beiden realen Data-Mining-Projekte demonstrieren.

Um die Forschungsfrage zu beantworten, besteht das Endergebnis der Mehrfachfallstudie aus Aussagen darüber, wie gut die Methodologie in diesen Fällen geeignet war und welche Erfahrungen mit ihr gemacht worden sind. Da sich beide Fälle thematisch stark voneinander unterscheiden, kann in einer „analytischen Generalisierung“ [71, S. 38] geschlossen werden, inwiefern sich die Entscheidungsträger-verständliche Methodologie auf weitere Projekte anwenden lässt.

3 Hypothese: eine Entscheidungsträger-verständliche Methodologie

Im Folgenden werden die Ansätze dieser Arbeit zur Implementierung des Prozessmodells beschrieben, das in Abschnitt 2.4 als Fokus für die Mehrfachfallstudie vorgestellt wurde. Diese Hypothese besteht aus einer Entscheidungsträger-verständlichen Methodologie.

Kurz zusammengefasst implementiert die Methodologie das Prozessmodell folgendermaßen: In einem Business-Case[1] legen die Beteiligten die Randbedingungen sowie Ziele des Data-Mining-Projekts fest. Die Ziele bestehen aus Berichten und Mustern. Ein Data-Assay wird vom Team erstellt, für einen Zugriff auf die Rohdaten sowie die Nutzung der Domänen- und Daten-Expertise. Daraufhin erstellt das Team ein Data-Warehouse, das den raschen Zugriff auf ausschließlich relevante Daten bietet. Die Abfrage des Data-Warehouse zur Erstellung von Berichten geschieht im Reporting. Im Data-Mining verwendet das Team spezielle Techniken, um nützliche Muster in den Berichten zu finden. In einer Business-Story fasst das Team schließlich seine Ergebnisse zusammen und präsentiert sie dem Entscheidungsträger. Dabei werden die genannten Teile nur im Idealfall sequentiell durchgeführt. Für den Entscheidungsträger sind der Business-Case und die Business-Story interessant; sie werden in diesem Kapitel zuerst behandelt. In den anschließenden Abschnitten werden dann die Umsetzungsteile beschrieben, die hauptsächlich das Team betreffen.

3.1 Business-Case

In dieser Arbeit wird angenommen, dass ein sogenannter Business-Case (vgl. [53, S. 205]) geeignet ist, um die Projektmanagement- und Vorentwicklungsprozesse durchzuführen, siehe Abschnitte 2.4.2 und 2.4.3.1.

Der Hauptzweck des Business-Case besteht darin, ähnlich einem Lasten- oder Pflichtenheft in der

[1] Für die in dieser Arbeit verwendeten englischen Begriffe, z.B. Business-Case, Data-Assay oder Management-Summary, wird auf eine deutsche Übersetzung verzichtet, da es sich bei ihnen um feststehende Ausdrücke aus der Literatur handelt.

Software-Entwicklung, Anforderungen zu beschreiben, die an ein erfolgreiches Projekt gestellt werden (vgl. [66, S. 17-24]).

Der Business-Case wird vom Team verfasst. Seine Informationen stammen aus Gesprächen zwischen dem Team und dem Entscheidungsträger sowie aus Machbarkeitsstudien (vgl. [48]), die das Team für ein Projekt durchführt. Eine Machbarkeitsstudie beschränkt sich meist auf das relativ oberflächliche Beschreiben der Daten, dem „sifting through the raw data“ (vgl. [11]). Dabei können ansatzweise Techniken aus der gesamten Methodologie nötig sein, insbesondere solche zur Datenvorverarbeitung.

Nun beschreibe ich die typischen Inhalte eines Business-Case und nenne weitere Gründe für seine Erstellung.

Wenn ein Business-Case auf mehrere Seiten angewachsen ist, lohnt es sich laut Pyle [53, S. 214] ein „Management-Summary“ voranzustellen, das für jedes Kapitel des Business-Case einen eigenen Abschnitt enthält und den Inhalt in weniger als zwei Seiten zusammenfasst. So kann sichergestellt werden, dass der Entscheidungsträger einen vollständigen Überblick über den Inhalt erhält, ohne den gesamten Text lesen zu müssen.

Die erste Version des Business-Case wird vor Projektbeginn erstellt. Mit ihm empfiehlt das Team dem Entscheidungsträger eine Lösung für eines seiner Probleme und möchte ihn davon überzeugen, diese Lösung in einem Projekt realisieren zu können. Der Business-Case ist daher ausschließlich „in management terms“ [53, S. 205] beschrieben, d.h., er muss ohne technische Hintergründe lesbar sein.

3.1.1 Hintergrund und Motivation

Der Business-Case startet mit einer Beschreibung der Hintergründe, die zu einem offenen Problem führen. Pyle nennt zwei häufige Einstiegszenarien für ein Data-Mining-Projekt: Entweder das Projekt startet mit dem Bereitstellen von Daten und der Anfrage, diese auf interessante Muster zu untersuchen. In diesem Fall gilt z.B. folgende allgemeine Herausforderung als Problem: „das richtige Produkt, am richtigen Ort, zum richtigen Preis, zur richtigen Zeit in der richtigen Menge zu besitzen“ [53, S. 216]. Oder das Projekt startet mit einem Problem, das untersucht werden soll. Pyle hält das zweite Szenario für sinnvoller, denn Data-Mining ist ein „Werkzeug“; kein Unternehmen kauft erst die Werkzeuge und entscheidet dann, was mit ihnen gemacht werden soll. Ich vermute, dass meist eine Kombination beider Szenarien eintritt: Daten werden verfügbar und motivieren, ein Problem zu behandeln; die Anforderungen, um das Problem zu behandeln, müssen allerdings erst ausgearbeitet werden. Zum Beispiel werden häufig erst seit der Umstellung auf das Bachelor/Master-Studiensystem die Prüfungsleistungen systematisch elektronisch erfasst und das Analysieren der Zusammenhänge zwischen Noten, Engagement und Studienabbrüchen möglich. Oder erst nachdem jahrzehntelang Fragebögen zu

Ohroperationen elektronisch erfasst worden sind, waren genügend Daten für eine Gesamtanalyse der Ohrdatenbank vorhanden.

3.1.2 Problemstellungen und Möglichkeiten

Hier werden die Problemstellungen beschrieben, die es gilt während des Projekts zu behandeln. Es sind die Ziele aus Sicht der Organisation, die in Abschnitt 2.4.3.1 genannt werden.

„[T]he most difficult part of finding a solution is accurately finding and stating the problem“ [53, S. 35]. Dieser Teil ist allerdings umso wichtiger, denn ein Projekt, welches das Problem des Entscheidungsträgers nicht berücksichtigt, kann nicht als erfolgreich bezeichnet werden (vgl. [30, S. 172]).

Für ein konkretes Problem werden Fragestellungen herausgearbeitet, die mittels Deskriptivem Data-Mining gelöst werden können. Diese können vom Entscheidungsträger frei formuliert werden. Dennoch ist laut Brachman und Anand [11] für diese „Aufgabenentdeckung“ ein reger Austausch zwischen Entscheidungsträger und Data-Mining-Team notwendig. Auch laut González-Aranda [30, S. 172] stellt das eine Herausforderung dar: Das Data-Mining-Team muss die Problemstellung aus Sicht des Entscheidungsträgers verstehen, um eine Lösung entwickeln zu können.

Aber nicht nur das Problem ist wichtig: In der Medizin wird eine Untersuchung normalerweise nur dann angefordert, wenn ihre Ergebnisse die Therapie beeinflussen können. Genauso hält es sich beim Data-Mining. Ein Data-Mining-Projekt rechtfertigt sich erst, wenn nicht nur Problem und Daten vorhanden sind, sondern wenn seine möglichen Ergebnisse in Maßnahmen operationalisierbar sind, um das Problem zu lösen (vgl. [34, S. 92]). Daher muss bereits zu diesem Zeitpunkt überlegt werden, inwiefern die KDDM-Ergebnisse genutzt werden sollen.

3.1.3 Aktuelle Situation und Datenlage

Auch die aktuelle Situation muss Überlegungen zu einem Data-Mining-Projekt begründen können, indem z.B. die Lösung der Problemstellungen spezielle Voraussetzungen erfordert. Der dabei wichtigste Aspekt entspricht den Daten. So muss geklärt werden, welche Datenquellen oder Rohdaten möglich sind, um mit ihnen die Fragestellungen zu behandeln. Auch dies ist ein Teil des Prozessmodells, siehe Abschnitt 2.4.3.1.

3.1.4 Empfohlene und alternative Lösungen

Hier stellt das Team seine Lösungsansätze zum Problem vor. Dabei sollte auch auf zusätzliche Datenquellen sowie alternative Lösungen eingegangen werden. Es sind die Ziele aus Sicht des

Teams, die es im Entwicklungsprozess zu erreichen gilt und in Abschnitt 2.4.3.1 des Prozessmodells genannt werden.

Mit dem Thema genauer auseinander gesetzt haben sich Britos et al. [13]. Das Ausarbeiten von Anforderungen in Data-Mining-Projekten ist nicht leicht, da „Nutzer und Team verschiedene Sprachen sprechen“. Die Autoren behaupten, dass bestehende Methodologien nicht ausreichen, z.B. „keine entsprechende Dokumentation unterstützen“.

Dieser Teil des Business-Case soll soweit wie möglich formalisiert und nachvollziehbar gestaltet werden. Denn wenn auch Data-Mining das Suchen nach verborgenen Informationen darstellt, so ist es in einem Projekt entscheidend, dass Entscheidungsträger und Team sich auf Kriterien einigen, mit denen der Erfolg des Projekts gemessen werden kann. Dieser Ansatz soll nun vorgestellt werden.

Eine KDDM-Lösung besteht aus einzelnen Anforderungen, die sich jeweils auf eine oder mehrere konkrete Fragestellungen beziehen. Jede Anforderung besteht aus vier Teilen, strukturiert in einer Tabelle, siehe 3.1.

Name:	**Der Name gibt die Bezeichnung vor, mittels der man sich in der Dokumentation oder in Gesprächen mit den Projektbeteiligten auf diese Anforderung beziehen kann.**
Eingabe:	Die Eingabe nennt die Datenquellen oder Rohdaten, die für diese Anforderung nötig sind.
Bericht:	Tabellen: Jeder Bericht besteht jeweils aus einer Menge an Tabellen, die jeweils eine bestimmte Gruppe an Objekten beschreiben.
	Zeilen: Jeder Bericht besteht jeweils aus einer Menge an Zeilen, die ein Objekt beschreiben.
	Spalten: Jede Zeile enthält eine bestimmte Menge an Spalten, die mit aufbereiteten Inhalten der Eingabe gefüllt sind.
Muster:	Ein Muster beschreibt eine Information, die aus dem Bericht herausgefunden werden soll.

Tab. 3.1: Struktur einer Anforderung

Das Vorgehen wird so in zwei Teile geteilt: Dem Erstellen von Berichten und der Analyse dieser Berichte. Diese Aufteilung wurde ausgewählt, da solche Fragen von einem Entscheidungsträger verstanden werden können und einen geringen Paradigmawechsel zwischen Organisationszielen und Data-Mining-Zielen darstellen. Laut Kohavi werden diese „reporting type questions“ [39] häufig sogar explizit vom Entscheidungsträger zur Beschreibung seiner Problemstellung verwendet.

Gleichzeitig müssen dem Entscheidungsträger „deeper analytic questions“ verständlich gemacht

werden, so dass er sie gegebenenfalls sogar selbständig stellen kann. Auch dies wird durch einen Bericht erreicht: Anstatt der Frage „Wie ist die Verteilung von Männern und Frauen bei Personen, die mehr als 500 Dollar ausgeben?“, kann die Frage „Was sind die Charakteristiken von Personen, die mehr als 500 Dollar ausgeben?“ sinnvoll sein (vgl. [39]). Erstere Frage bezieht sich direkt auf den Inhalt der Rohdaten und ist daher einfach zu stellen; zweitere Frage kann meiner Meinung nach erst gestellt werden, nachdem mögliche Charakteristiken festgelegt worden sind, ist allerdings weitaus ergiebiger. Ein Bericht kann erstere Frage direkt beantworten: Für Personen, die mehr als 500 Dollar ausgeben wird darin für weibliche und männliche Personen die Anzahl angegeben. Er kann allerdings auch weitere Attribute für solche Personen sowie deren Anzahl angeben und zu der *tieferen* Frage führen, welche Attributwerte bei den meisten Personen auftreten. Auch ein technisch unversierter Entscheidungsträger kann eine solche Frage stellen und nachvollziehen. Denn sie lässt sich auch manuell beantworten; in diesem Fall müsste im Bericht für jede Kombination an Attributwerten die Anzahl an Personen abgelesen werden. Da die Anzahl an Kombinationen jedoch exponentiell zur Anzahl der Attribute ist, ist der Aufwand jedoch zu hoch. Deshalb wird das Muster – in dem Fall eine häufig auftretende Attributwertkombination – im Data-Mining-Schritt weitgehend automatisch gesucht, z.B. mit einem Algorithmus. In Abschnitt 3.6 werde ich auf weitere Muster eingehen, die sich auf solche Entscheidungsträger-verständlichen Fragen abstrahieren lassen (vgl. [30, S. 172]) und Data-Mining-Techniken nennen, um sie zu entdecken.

Genauso wie die anderen Teile des Business-Case, sollen auch die Anforderungen ausschließlich in einer Entscheidungsträger-verständlichen Sprache formuliert werden. Im Reporting und Data-Mining, die in späteren Abschnitten behandelt werden, können beliebige Berichte und Muster, die ein Entscheidungsträger verlangt, umgesetzt werden. In Tabelle 3.2 wird eine Anforderung aus dem Data-Mining-Projekt einer Hals-Nasen-Ohren-Klinik als Beispiel beschrieben.

Name:	**Hörfähigkeitsentwicklung**
Eingabe:	Ohrdaten
Bericht:	Zeilen: Patient
	Spalten: für jedes „Audiogramm“, Durchschnitt der „LL-KL-Differenz“ bei Frequenzen des „sozialen Gehörs”
Muster:	Ein Diagramm, auf dem diese Kennzahl der Hörfähigkeit für alle Patienten eingezeichnet und für jeden Patienten mit Linien verbunden ist und sich ein möglicher Trend ablesen lässt.

Tab. 3.2: Anforderung Hörfähigkeitsentwicklung

Jede Anforderung wird durch Gründe bestärkt, die zu ihrer Auswahl geführt haben. Daneben werden Annahmen und mögliche Schwierigkeiten bei der Erstellung der Anforderung beschrieben. Data-Mining ist hochinteraktiv und -iterativ, weshalb stets Änderungen an den Anforderun-

gen auftreten können, beispielsweise auf Wunsch des Entscheidungsträgers und aus erworbenen Kenntnissen während des Projekts. Der Lebenszyklus des Projekts, wie in Abschnitt 2.4.2.1 behandelt, ist insbesondere durch die Änderungen der Anforderungen geprägt. Auch aus diesem Grund ist es wichtig, jegliche Änderungen des Business-Case mit dem Entscheidungsträger abzusprechen und ihm stets eine aktuelle Version zur Verfügung zu stellen.

Neben dem empfohlenen Lösungsansatz dürfen alternative Ansätze nicht fehlen; es gilt zu begründen, weshalb sie nicht empfohlen werden.

3.1.5 Projektplanung

Nachdem im vorherigen Abschnitt dem Entscheidungsträger eine Lösung empfohlen wurde, wird nun beschrieben, wie diese erreicht werden soll. Außerdem müssen die nötigen Mittel für diese Lösung beschrieben und gerechtfertigt werden. Bereits zu Projektstart sollten die Beteiligten, also Auftraggeber bzw. Entscheidungsträger und Team identifiziert worden sein, siehe Abschnitt 2.4.3.1. Wichtigster Punkt sind die Ressourcen, da für sie ein Budget nötig ist, beispielsweise Hardware und Software. Sofern für den Entscheidungsträger relevant, können sie auch näher spezifiziert werden. In einem Zeitplan werden die Termine für Meilensteine genannt. Diese bestehen in Folge aus dem geplanten Abschließen eines *Data-Assay* – einer umfassenden Beschreibung der Daten –, eines *Data-Warehouse* – einer Umgebung zur flexiblen Abfrage –, des Reporting – der Erstellung der genannten Berichte – und dem Data-Mining – der Erstellung der Muster aus den Anforderungen. Diese werden in späteren Abschnitten behandelt und implementieren die Entwicklungsprozesse aus Abschnitt 2.4.3.2. Außerdem wird der Ablieferungstermin der Ergebnisse festgelegt.

Letztendlich wird zum Thema Projektplanung beschrieben, wie die Ergebnisse dem Entscheidungsträger abgeliefert und langfristig genutzt werden sollen, siehe 2.4.3.3. In der Methodologie werden die Berichte und Muster übergeben, in einer Business-Story in einen Zusammenhang mit den Fragestellungen und dem Problem gebracht sowie Hinweise zur Nutzung der Ergebnisse, z.B. des Data-Warehouse, gegeben.

3.1.6 Glossar

Zur Unterstützung der Kommunikation zwischen Team und Entscheidungsträger, aber auch zwischen den einzelnen Teammitgliedern, erklärt ein Glossar die entscheidenden Begriffe während des Projekts (vgl. [39]), wie in Abschnitt 2.4.3.1 gefordert. Dabei werden insbesondere solche Begriffe genannt, die in mehreren Anforderungen von Bedeutung sind und nicht jeweils erneut erklärt werden sollen.

Der Business-Case ist kein statisches Dokument, vielmehr wird es während der Durchführung

des Projekts nötig sein, ihn in Folge neuer Erkenntnisse zu aktualisieren, z.B. die Anforderungen oder Meilensteintermine. Über eine Änderung des Business-Case ist der Entscheidungsträger in Kenntnis zu setzen (vgl. [66, S. 160]). Wie bereits erwähnt, handelt es sich bei einem KDDM-Projekt um einen interaktiven und iterativen Prozess, was somit berücksichtigt wird.

3.2 Business-Story

Das Deployment im Deskriptiven Data-Mining besteht laut Pyle [53, S. 509] aus dem Zusammenstellen der Ergebnisse und ihrer Präsentation in einer „Geschichte“. Sie beschreibt die Ergebnisse und gibt Vorschläge für die kurz- und langfristige Nutzung.

Dabei ist Data-Mining nur eine Eingabe im Entscheidungsprozess (vgl. [23]). Daher geht es in diesem Teil um die Ausstattung des Entscheidungsträgers mit den Ergebnissen, aber nicht um das tatsächliche Treffen der Entscheidungen. Welche Motivation die einzelnen Anforderungen haben, steht im Business-Case; die Ergebnisanalyse nach dem Data-Mining und die fortdauernde Evaluation stellen sicher, dass die Ergebnisse mit den Zielen und damit auch mit den Motivationen zusammenhängen.

Die Business-Story soll weder in der Sprache von Daten noch Werkzeugen, sondern ausschließlich in der Sprache des Entscheidungsträgers erfolgen (vgl. [53, S. 265]). Sie ist neben dem Business-Case das einzige, das den Entscheidungsträger mit Sicherheit interessiert und soll nicht nur fachlich überzeugen, sondern „einnehmend“, „interessant“ und „unterhaltend“ geschrieben sein. Ähnlich wie der Business-Case, kann daher auch die Business-Story mit einem Management-Summary starten.

3.2.1 Ziele

Die Business-Story wird durch eine kurze Zusammenfassung des Business-Case eingeleitet. Insbesondere die Ziele sollen hier nochmal genannt werden, um dem Entscheidungsträger die Vorteile des Projekts deutlich zu machen.

3.2.2 Entdeckung und Verifikation

Das zentrale Thema der Geschichte ist das Problem des Entscheidungsträgers. Laut Pyle [53, S. 264] wird im Hauptteil jede Fragestellung mit den Abschnitten „Entdeckung“ und „Verfikation“ behandelt.

Die Entdeckung besteht aus beschreibenden und erklärenden Informationen des Teams zum Beantworten einer Fragestellung. Pyle [53, S. 264ff] versteht darunter das Nennen von interessanten

Mustern, z.B. „Kennzahlen zu den Rohdaten“, „Beziehungen zwischen Attributen“ oder weiteren „plausiblen Erklärungen“. Mehrere Muster können auch in einer „logischen Erklärungskette“ verbunden werden.

In der Verifikation soll der Entscheidungsträger von der Richtigkeit der Beschreibungen und Erklärungen überzeugt werden. Dazu wird ihm erläutert, wie diese durch die Berichter und Muster abgeleitet worden sind – ggf. unterstützt durch Diagramme, Tabellen, Formeln oder beliebige andere Mittel, die dem Verständnis des Entscheidungsträgers dienen (vgl. [53, S. 265]). Sein Vertrauen in die Ergebnisse ist von äußerster Bedeutung, weshalb Pyle auch empfiehlt, Informationen in die Business-Story einzubauen, die mit bekanntem Hintergrundwissen des Entscheidungsträgers übereinstimmen (vgl. [53, S. 265]). Allerdings gehört es zur Verifikation auch, mögliche Schwächen der Erklärungen, z.B. „negative Beweise“ zuzugeben und kritische Themen wie „Repräsentativität“, „Annahmen“ oder „Ausreißer“ anzusprechen.

3.2.3 Ausblick

Ein weiteres Thema der Business-Story ist die langfristige Nutzung der Ergebnisse. Ein Data-Warehouse besitzt den Vorteil, auch neue Daten aufnehmen zu können. Genauso können auch die Techniken des Data-Assay, des Reporting und des Data-Mining mit neuen Daten – unter der Voraussetzung, sie besitzen die selbe Struktur – vereinfacht wiederholt werden. Ein Ziel von Data-Mining sollte es sein, die nötige Erfahrung und Zeit bei der Analyse zu verringern. Idealerweise kann der Entscheidungsträger eigene Analysen erstellen (vgl. [39]). Deshalb sollte meiner Meinung nach die Geschichte mit einem Teil „Ausblick“ abgerundet werden, der auf dieses Thema eingeht.

Die Business-Story ist stark Kontext-abhängig, das konkrete Beschreiben und Abliefern der Ergebnisse hängt von individuellen Bedürfnissen des Entscheidungsträgers ab, muss daher sehr allgemein gehalten sein und lässt sich nicht weiter formalisieren.

3.3 Data-Assay

Das Team besitzt allgemeines Hintergrundwissen zu den Daten, dem Fachbereich sowie zu Data-Mining-Techniken. Um dieses Hintergrundwissen zu nutzen, ist es wichtig, ein „Gefühl für die Daten“ (vgl. [39]) zu bekommen, und diese in den Kontext des Hintergrundwissens zu bringen. Dazu werden die Daten im Data-Assay umfassend beschrieben. Im Rahmen dessen werden das Business-Understanding, Data-Understanding sowie ein Teil der Data-Preparation aus Abschnitt 2.4.3.2 implementiert.

Das allgemeine Ziel des Data-Assay besteht darin, sicher zu stellen, dass in darauffolgenden

Entwicklungsprozessen die Anforderungen realisiert werden können. Dazu gehören:

- Informationen zum Inhalt und Umfang der Daten. Die Qualität der Daten ist zur Realisierung der Anforderungen von Bedeutung.
- Möglichkeiten zur Abfrage und Weiterverarbeitung der Daten. Es muss sichergestellt sein, dass ein Zugriff auf die Daten möglich ist, über den beliebige Fragen an die Daten beantwortet werden können, wenn auch ggf. mit großem Aufwand.
- Informationen über die Daten zur Entwicklung eines Data-Warehouse, das in Abschnitt 3.4 beschrieben wird. Diese sind nötig, um die Daten in vordefinierte Strukturen bringen zu können.

3.3.1 Datenquellen

Den Anfang des Data-Assay stellen Datenquellen dar: Sie bieten Daten in elektronischer Form. In Anlehnung an Pyle [53] gibt es drei verschiedene Datenquellen:

Entwickelte Datenquellen; Datenquellen, die speziell für ein Data-Mining-Projekt gebildet wurden (z.B. Umfragen, Ergebnisse von Daten-Crawlern). Diese können sehr ergiebig sein, da die Struktur der Inhalte – z.B. Antworten auf spezielle Fragen – vorher geplant werden kann. Diese Daten sind besonders aufwändig zu erhalten, im Gegenzug ist die Analyse einfacher, da ihre Aufbereitung für die Analyse praktisch wegfällt.

Gekaufte Datenquellen; externe Datenquellen außerhalb der Organisation (z.B. Webseiten, Wetterdaten, Geodaten, Satelitendaten), deren Inhalte meist bezahlt werden müssen.

Operationale Datenquellen; interne Datenquellen, die direkt verfügbar sind (z.B. aus Datenbanken, Dokumenten). Ihre Daten wurden nicht speziell zum Data-Mining gesammelt, sondern entstehen aus den Tätigkeiten der Organisation heraus. Operationale Datenquellen sind die kostengünstigste und daher häufigste Datenquelle. Sie werden in dieser Arbeit schwerpunktmäßig behandelt.

Die Daten müssen ursprünglich nicht in elektronischer Form vorliegen: Für eines unserer Data-Mining-Projekte wurden papierne Strichlisten elektronisch erfasst und standen anschließend als Datenquelle zur Verfügung. Die elektronischen Daten können in verschiedenen Formaten vorliegen – von völlig unstrukturiert als Freitext in *Microsoft Word*-Dokumenten, bis vollständig strukturiert aus einer relationalen Datenbank. Unabhängig davon, wie die tatsächliche Datenquelle beschaffen ist, zur Analyse mit der vorgestellten Methodologie müssen ihre Daten zu einer *Tabellarischen Form* strukturiert werden.

Wie der Name bereits andeutet, bestehen solche Daten aus Tabellen mit einer festen Anzahl an Spalten und Zeilen, mit Inhalten gefüllt: Jede Tabelle beschreibt ein Objekt der Datenquelle;

jede Spalte repräsentiert ein Attribut des Objekts; jede Zeile beschreibt eine Instanz des Objekts und besitzt für jedes Attribut einen Attributwert (ggf. einen leeren Attributwert).

Ich habe Tabellarische Daten als Voraussetzung für die Analyse ausgewählt, da sie eine bewährte Datenstruktur darstellen. Auch Pyle [53], Han und Kamber [32] sowie Nisbet et al. [50] haben sich in ihren Werken auf die Analyse Tabellarischer Daten konzentriert, wenn sie auch teilweise andere Begriffe für Attribute oder Instanzen verwenden, respektive *Felder* und *Vektoren* zum Beispiel. Außerdem können Tabellarische Daten relativ problemlos in relationalen Datenbanken gespeichert werden, eine Voraussetzung für die vorgestellte Analysemethode.

Das konkrete Ziel des Data-Assay besteht darin, die Daten der Datenquellen in Tabellarische Form zu bringen, in einer Datenbank zu speichern und zu beschreiben. Letztendlich sollen für jedes Objekt einer Datenquelle folgende Informationen herausgefunden werden:

Name Der Name gibt gleichzeitig den Namen der Tabelle vor, in der das Objekt gespeichert werden soll.

Beschreibung Der Inhalt der Daten kann in einer Beschreibung näher erklärt werden.

Anzahl Instanzen Diese Zahl gibt einen Eindruck über den Umfang der Daten.

Anzahl Attribute Auch diese Zahl gibt einen Eindruck über den Umfang der Daten.

Bemerkungen Hier sind z.B. Informationen zur Form der Datenquelle, z.B. die ASCII-Kodierung einer Datei enthalten. Solche Informationen können vielfältig sein – ihr Hauptzweck besteht darin, mögliche Schwierigkeiten bei der weiteren Verarbeitung der Daten herauszufinden – und können daher nicht näher formalisiert werden.

Für jedes Attribut eines Objekts sollen folgende Informationen herausgefunden werden:

Name Jedes Attribut benötigt einen aussagekräftigen Namen. Dieser ist entweder von den Rohdaten vorgegeben und kann übernommen werden oder er ist nicht vorhanden bzw. schwer verständlich (z.B. in einer anderen Sprache oder numerisch kodiert). In dem Fall wird ein aussagekräftiger Name gewählt.

Beschreibung Die Funktion des Attributs kann in einer Beschreibung näher erklärt werden.

Datentyp Diese Arbeit beschränkt sich auf die Analyse von Numerischen, Zeitgebenden und Kategorischen Attributen, die im weiteren Verlauf dieses Abschnitts definiert werden.

Datenquelle Die Datenquelle eines Attributs entspricht seinem Objekt. Diese Information ist von Bedeutung, da es beim Aufbau des Data-Warehouse nötig sein kann, Attribute verschiedener Objekte in einem Objekt zu integrieren.

Bemerkungen In den folgenden Abschnitten werden mögliche Eigenschaften von Attributen genannt, die es gilt, während des Data-Assay festzustellen. Auch diese Eigenschaften können

nicht weiter eingegrenzt werden, denn jegliche mögliche Schwierigkeiten bei der Weiterverarbeitung der Attribute sollten genannt werden.

Ein Objekt einer Datenquelle kann dann grafisch in einem Diagramm repräsentiert werden, das seinen Namen sowie die Namen und Datentypen seiner Attribute nennt, Abbildung 3.1 zeigt als Beispiel ein Objekt, in dem Patienten mit einem Schlüsselwert, einer Kennzeichnung, ihrem Geburtsdatum, sowie ihrem Geschlecht, kodiert als Zahl, repräsentiert werden.

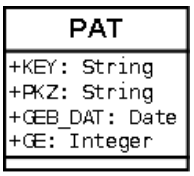


Abb. 3.1: Beispiel einer Datenquelle in Tabellarischer Form

Eine Datenquelle, in der diese Patientenstammdaten vorliegen können ist die *CSV-Datei*. Sie liegt bereits in Tabellarischer Form vor und besteht aus Textzeilen, in denen jeweils eine feste Menge an – mit einem Zeichen, z.B. Komma getrennten – Werten beschrieben sind. Diese Werte können ggf. von Anführungsstrichen umschlossen werden, für den Fall, dass sie Kommas enthalten. Die erste Zeile beschreibt meist als Kopfzeile die Namen der Attribute. Wenn Werte mit Leerzeichen umschlossen sind, können diese durch sog. *Trimmen* entfernt werden.

3.3.2 Datentypen

Im Folgenden werden Numerische, Kategorische und Zeitgebende Attribute definiert.

3.3.2.1 Numerische Attribute

Numerische Attribute teilen einer Instanz eine konkrete Zahl zu. Dies können natürliche (diskrete) oder reelle (kontinuierliche) Zahlen sein. Eine reelle Zahl besitzt ein festgelegtes Dezimalzeichen.

Ein Numerisches Attribut kann in einer Datenbank entweder als natürliche Zahl, *Integer*, oder als reelle Zahl, *Real*, gespeichert werden. Es dient der Verminderung des Speicherbedarfs sowie der Performanz der Abfragen, wenn der maximale Wert sowie die maximale Anzahl an Nachkommastellen festgelegt wird.

Von besonderer Bedeutung sind *Missing-Values*, die eingegeben werden, wenn zur Zeit der Datenerfassung kein Wert bekannt war. In eine relationalen Datenbank können Missing-Values über den Leerwert *null* eingegeben werden. Dieser muss allerdings vom Leerwert, der eingegeben wird, wenn explizit kein Wert vorhanden ist, unterschieden werden. In Numerischen Werten wird in

diesem Fall häufig ein „surrogate null“-Wert (vgl. [53, S. 280]), z.B. „-1“ verwendet, den es zu identifizieren gilt.

3.3.2.2 Kategorische Attribute

Kategorische Attribute teilen einer Instanz eine konkrete Kategorie, beschrieben durch eine alphanumerische Zeichenfolge, zu. Darunter fallen auch Attribute mit beliebig langen Zeichenketten als mögliche Werte. Eine spezielle Form eines Kategorischen Attributs ist das binäre Attribut, dass ausschließlich die Werte „0“ und „1“ enthält.

Bei Kategorischen Attributen muss der Missing-Wert, z.B. „?“, „unknown“ oder *null* von dem explizit leeren Wert „“ unterschieden werden. Ein Kategorisches Attribut wird in einer Datenbank als *String* gespeichert. Auch hier dient es dem Speicherbedarf sowie der Performanz von Datenbankabfragen, wenn die maximale Länge der Zeichenfolge vorgegeben wird.

3.3.2.3 Zeitgebende Attribute

Zeitgebende Attribute teilen einer Instanz eine konkrete Zeit in Form eines Datums (ggf. mit Uhrzeit) zu. Zeitgebende Attribute kommen in verschiedenen Formaten vor, z.B. „YYYY-MM-dd“, die die Reihenfolge und die Trennzeichen der Inhalte wie Jahr, Monat und Tag angeben. Sie können als Mischung aus Numerischen und Kategorischen Attributen angesehen werden. Sie besitzen eine Reihenfolge, ein Maximum mit der jüngsten Zeit und ein Minimum mit der ältesten Zeit.

Auch bei Zeitgebenden Attributen ist ein „surrogate null“-Wert üblich. Zur Unterscheidung von Missing-Values und expliziten Leerwerten von Zeitgebenden Attributen bietet es sich an, in der Datenbank die Werte *null* und „00.00.0000“ zu verwenden. Zeitgebende Attribute können in der Datenbank entweder ohne (Date) oder mit Uhrzeit (Datetime) gespeichert werden.

3.3.3 Datenbank

Daten in Tabellarischer Form besitzen die Eigenschaft, dass sie in relationalen Datenbanken gespeichert werden können. Vorher können Aufgaben der Säuberung, Selektion oder Transformation der Daten, wie sie in Abschnitt 2.4.3.2 genannt werden, nötig sein:

- Wenn Daten noch nicht in Tabellarischer Form vorliegen, müssen sie in diese Form gebracht werden.
- Manche Daten müssen zunächst in die „erste Normalform“ [6, S. 308] transformiert werden. Demnach enthält jedes Attribut ausschließlich Werte, die einen „elementaren Typ”(z.B. Text-, Zahl- oder Zeitwert) besitzen und sich nicht in weitere, separat interpretierbare

Werte auftrennen lassen. Dies ist eine Voraussetzung für die Analyse. Weiterführende Normalformen vermindern z.B. Redundanz und stellen langfristig die Konsistenz der Daten sicher; dies ist für die Analyse nicht zwingend notwendig.

- Einzelne Werte müssen abgeändert werden, damit ein Attribut mit einem bestimmten Datentyp in der Datenbank gespeichert werden kann.
- Filtern von Daten mit geringer Qualität, z.B. Attribute, bei denen nur ein konstanter Wert oder nur Missing-Values vorkommen, können entfernt werden (vgl. [53, S. 286]). Attribute mit sehr vielen Missing-Values gelten als „sparsely populated“; bei mehr als 80% ein Hinweis auf schlechte Qualität der Daten (vgl. [53, S. 286]) und ein möglicher Grund, die Datenquelle nicht in die Analyse einzubeziehen.

3.3.4 Beschreibung

Es wurde bereits darauf eingegangen, dass ein Zweck des Data-Assay darin besteht, die Daten in den Kontext des Hintergrundwissens zu setzen und Informationen zur späteren Eingabe in ein Data-Warehouse zu erhalten. Im Folgenden sollen einige Möglichkeiten genannt werden, mit denen sich die Objekte und Attribute beschreiben lassen.

Zum Beispiel kann es niemals schaden, einen direkten Blick in die Rohdaten zu werfen, z.B. mit einem Texteditor. Dabei fallen einem Sonderzeichen, Dezimalzeichen, Leerzeichen o.ä. auf. Auch die Qualität der Daten kann direkt eingeschätzt werden. Zum Beispiel lassen sich leicht extreme Werte (z.B. Ausreißer), falsche Werte (z.B. „-1“ für Note) oder neben „surrogate null“ weitere Ersatzwerte mit besonderer Bedeutung (z.B. „900“ für „Prüfung noch nicht korrigiert“) erkennen.

Des Weiteren können Diagramme wie *Box-Plot*, *Verteilung* oder *Histogramm* nützliche Informationen bieten. In der vorgestellten Methodologie werden sie als Data-Mining-Techniken verstanden, da sie Informationen deutlich machen, die in den Rohdaten nicht direkt sichtbar sind, z.B. Extremwerte oder Mittelwerte. Sie werden im Abschnitt 3.6 beschrieben. Aber nicht nur Diagramme, sondern auch weitere Data-Mining-Techniken können sinnvoll sein, um Eigenschaften der Rohdaten festzustellen.

Die Abfragesprache der Datenbank bietet weitere Beschreibungsmöglichkeiten; die Standard-Abfragesprache für relationale Daten ist *SQL*, die in Abschnitt 3.5.1 behandelt wird. Im Folgenden nenne ich ein häufiges Beispiel, die Beschreibung von sog. *Schlüsselattributen*:

Attribute können „fachliche“ [6, S. 314] Schlüsselattribute darstellen. Sie sind eine Voraussetzung für den Aufbau des Data-Warehouse, das im nächsten Kapitel beschrieben wird.

Der sog. *Primärschlüssel* ist das Attribut, für das jede Instanz eines Objekts einen eindeutigen Wert unter allen anderen Instanzen dieses Objekts besitzt. Ein Primärschlüssel kann auch

aus mehreren Attributen bestehen. Er ist insofern wichtig, dass sich anhand seines Wertes eine Instanz eindeutig identifizieren lässt. Ein Primärschlüssel muss immer einen Wert besitzen („Entitäts-Integrität“ [6, S. 307]). Um Primärschlüssel zu identifizieren, muss die Anzahl an unterschiedlichen Attributwerten mit der Anzahl an Instanzen verglichen werden, was mittels „Count“-Operator von SQL möglich ist.

Unter einem sog. *Fremdschlüssel* können ein oder mehrere Attribute eines Objekts verstanden werden, für die es in anderen Objekten Pendants gibt. Wenn die Werte von zugehörigen Fremdschlüsseln in Instanzen zweier Objekte einander entsprechen (meist identisch sind, es gibt aber auch Fälle, in denen andere Entsprechungen sinnvoll sind), stehen die Instanzen in einem Zusammenhang. Fremdschlüssel sind insofern sehr wichtig, da sie Beziehungen (bzw. „Assoziationen“ [6, S. 307]) zwischen Objekten beschreiben. Ein Fremdschlüssel, der zu einem Primärschlüssel korrespondiert, besitzt entweder einen leeren Wert oder einen Wert, für den ein Primärschlüsselwert vorhanden ist (vgl. „Referentielle Integrität“ [6, S. 307]). Um ein Attribut eines Objekts als Fremdschlüssel zu einem Attribut eines anderen Objekts zu identifizieren, kann ein *Verbund* zwischen diesen Objekten abgefragt werden. In einem Verbund werden für jede Instanz des einen Objekts die Instanzen des anderen Objekts gesucht, bei denen Fremdschlüssel und Pendant-Attribut sich gegenseitig entsprechen. Für jedes Instanzenpaar enthält der Verbund eine Instanz mit den Attributen beider Objekte. Der Verbund wird ggf. um Instanzen des Objekts mit dem Fremdschlüssel ergänzt, für das keine entsprechende Instanz des anderen Objekts gefunden werden kann; dessen Attribute erhalten im Verbund keinen Wert. Der Verbund zweier Objekte lässt sich mit dem „Join“-Operator von SQL abfragen.

Bei der Umsetzung der Anforderungen aus dem Business-Case können beliebige Fragen an die Datenquellen auftreten, die zudem kontextabhängig sind; deshalb lässt sich ihre Beschreibung hier nicht weiter formalisieren.

Die Ergebnisse aus dem Data-Assay sollen dokumentiert und jedem Teammitglied zugänglich gemacht werden, in Abschnitt 7.6 werde ich auf meinen Ansatz zur Dokumentation eingehen. Wenn verlangt, können die Inhalte aus dem Data-Assay Entscheidungsträger-freundlich aufbereitet und dem Entscheidungsträger zur Verfügung gestellt werden – als erstes nützliches Ergebnis des Data-Mining-Vorhabens (vgl. [37]).

Ich bezeichne die Datenquellen sowohl in ihrer Ursprungsform, als auch in Tabellarischer Form in der Datenbank als *Rohdaten*. Denn an ihnen wurden so wenig Änderungen wie möglich vorgenommen, ihre Vorverarbeitung kann vollständig nachvollzogen werden. Daher bieten sich die Informationen aus dem Data-Assay zur Evaluation (siehe Abschnitt 2.4.2.3) an, z.B. zur Prüfung von Annahmen der Daten- und Domänen-Experten. Des Weiteren können die Kennzahlen aus dem Data-Assay in späteren Phasen des Projekts zum Vergleich herangezogen werden, um die dazwischenliegenden Vorverarbeitungsschritte zu evaluieren.

3.4 Data-Warehouse

Die Rohdaten sind durch das Data-Assay *händelbar* geworden, eignen sich jedoch weder zum Erstellen von Berichten, noch zum Suchen von Mustern. Um in verständlichen Berichten und nützlichen Analysen zu münden, sind „integrierte, konsistente und saubere“ (vgl. [31]) Daten notwendig. Außerdem kann es nötig sein, interaktiv durch die Daten durch zu gehen, sie auf unterschiedlichen Detailstufen in einem „360-degree view“ [39] zu betrachten. Daher lautet eine weitere Hypothese, dass für das Data-Preparation während der Entwicklungsprozesse aus Abschnitt 2.4.3.2 ein Data-Warehouse aufgebaut werden kann und diese Anforderungen erfüllt. Ein Data-Warehouse besteht im Grunde aus einer Datenbank, in der die Daten in vordefinierten Strukturen, Datenbankmodellen, gespeichert sind und abgefragt werden können. Nur in wenigen Situationen ist bereits ein Data-Warehouse vorhanden. Vielmehr bringt häufig ein Data-Mining-Projekt den Ausschlag, ein Data-Warehouse aufzusetzen.

„Ein guter Ingenieur hebt sich vom Handwerker ab, wenn er Probleme nicht an der Maschine sondern auf der Zeichnung löst“ [24]. Die Analyse und der Entwurf eines Data-Warehouse sollen durch Konzeptmodelle abstrahiert werden können, die die Umsetzung möglichst komplett vorgeben und sie dadurch vereinfachen.

3.4.1 Entity-Relationship-Modell

Ein Entity-Relationship-Modell stellt eine Gesamtheit an Objekten, Attributen sowie Beziehungen zwischen Objekten dar. Der Zweck eines solchen ER-Modells besteht darin, Elemente der Realität in einem Modell und durch die vorhandenen Daten abzubilden. Dabei sollen diejenigen Elemente in das Modell aufgenommen werden, die in den Anforderungen genannt werden. Letztendlich können nur Instanzen der enthaltenen Objekte sowie Werte ihrer Attribute später in Berichten auftauchen und mittels Data-Mining-Techniken auf Muster untersucht werden.

Ich habe das ER-Modell als Teil des Data-Warehouse ausgewählt, weil es folgende Vorteile besitzt:

- Ein ER-Modell kann in einem Diagramm übersichtlich dargestellt und konzeptioniert werden (vgl. [6, S. 22]).
- Es bietet vergleichbar einem „data source view“ [32, S. 128] einen Überblick über die vorhandenen Daten.
- Es gibt die Umsetzung relativ fest vor, vermindert damit Fehler.
- ER-Modelle können mittels *SQL*, das in einem späteren Abschnitt behandelt wird, abgefragt werden und bieten die Möglichkeit, Berichte zu erstellen.

Das Vorgehen bei der Konzeptionierung des ER-Modells besteht darin, mittels Informationen aus dem Data-Assay ein Gesamtmodell an Objekten mit Attributen sowie Beziehungen zueinander zu entwerfen. Die Objekte können dabei aus integrierten Objekten der Datenquellen bestehen. Und auch die Attribute können aus Attributen der Rohdaten abgeleitet worden sein. Damit erfüllt das ER-Modell insbesondere die Aufgaben der Integration und Konstruktion der Daten, die in Abschnitt 2.4.3.2 zum Prozessmodell gezählt werden.

Ein ER-Modell kann als „Klassendiagramm“ [6, S. 22] dargestellt werden. Darin werden die Objekte mit Namen und einer Menge an Attributen beschrieben. Jedes Attribut wird wiederum mit seinem Namen, Datentyp sowie Datenquelle beschrieben, ggf. durch „/“ als abgeleitetes Attribut gekennzeichnet. Beziehungen werden durch Verbindungen zwischen Objekten dargestellt, erhalten zudem eine Information über die Art der Beziehung. Zum vereinfachten Verständnis können auch die Beziehungen eine Bezeichnung erhalten. Die Umsetzung der Beziehungen erfolgt über Schlüsselattribute, diese werden im Diagramm nicht dargestellt, da sie indirekt vorgegeben sind.

An relevanten Beziehungen eines Objekts zu einem anderen Objekt können unterschieden werden:

1:n wenn jeder Instanz des Objekts ein oder mehrere Instanzen eines anderen Objekts zugeordnet werden. Der Wert „n“ kann dabei auch durch eine konkrete Zahl oder ein Intervall ersetzt werden.

1:0,n wenn jeder Instanz des Objekts kein, ein oder mehrere Instanzen eines anderen Objekts zugeordnet werden.

Andere Beziehungen ergeben sich aus diesen Grundbeziehungen: Eine n:1- und 0,n:1-Beziehung entsteht aus der 1:n- bzw. 1:0,n-Beziehung, indem auch das Pendant-Attribut als Fremdschlüssel verwendet wird. Die m:n-Beziehung wird über ein Zwischenobjekt, zu dem die beiden Objekte mit 1:n-Beziehungen stehen, realisiert. Eine umfassendere Beschreibung bietet Balzert (vgl. [6, S. 43ff]).

Abbildung 3.2 zeigt ein Beispiel für das Diagramm eines ER-Modells. Diesem Modell nach unterzieht sich jeder Patient ein oder mehreren Operationen. Jede Operation besitzt höchstens eine Voruntersuchung, höchstens eine OP-Untersuchung und beliebig viele Nachuntersuchungen.

Während der Konzeption des ER-Modells werden Objekte mit Attributen entworfen. Objekte können dabei mehrere Datenquellen integrieren. Es werden neue, aus vorhandenen Attributen abgeleitete, Attribute erzeugt. Pyle [53, S. 301] nennt dies „Feature Extraction“, Nisbet et al. [50, S. 47] nennen es „Data Derivation“, in unserem Prozessmodell ist es in der Aufgabe „Konstruktion“ aus Abschnitt 2.4.3.2 enthalten. Dies bedarf Kreativität und ist eine Form der künstlerischen Modellierung; Nisbet et al. [50, S. 47] vergleichen insbesondere diesen Schritt des KDDM mit der künstlerischen Tätigkeit eines Bildhauers. Wichtig ist dennoch der Zusammenhang mit dem Problem des Entscheidungsträgers. Falls Anforderungen sich nicht direkt umsetzen lassen, kann

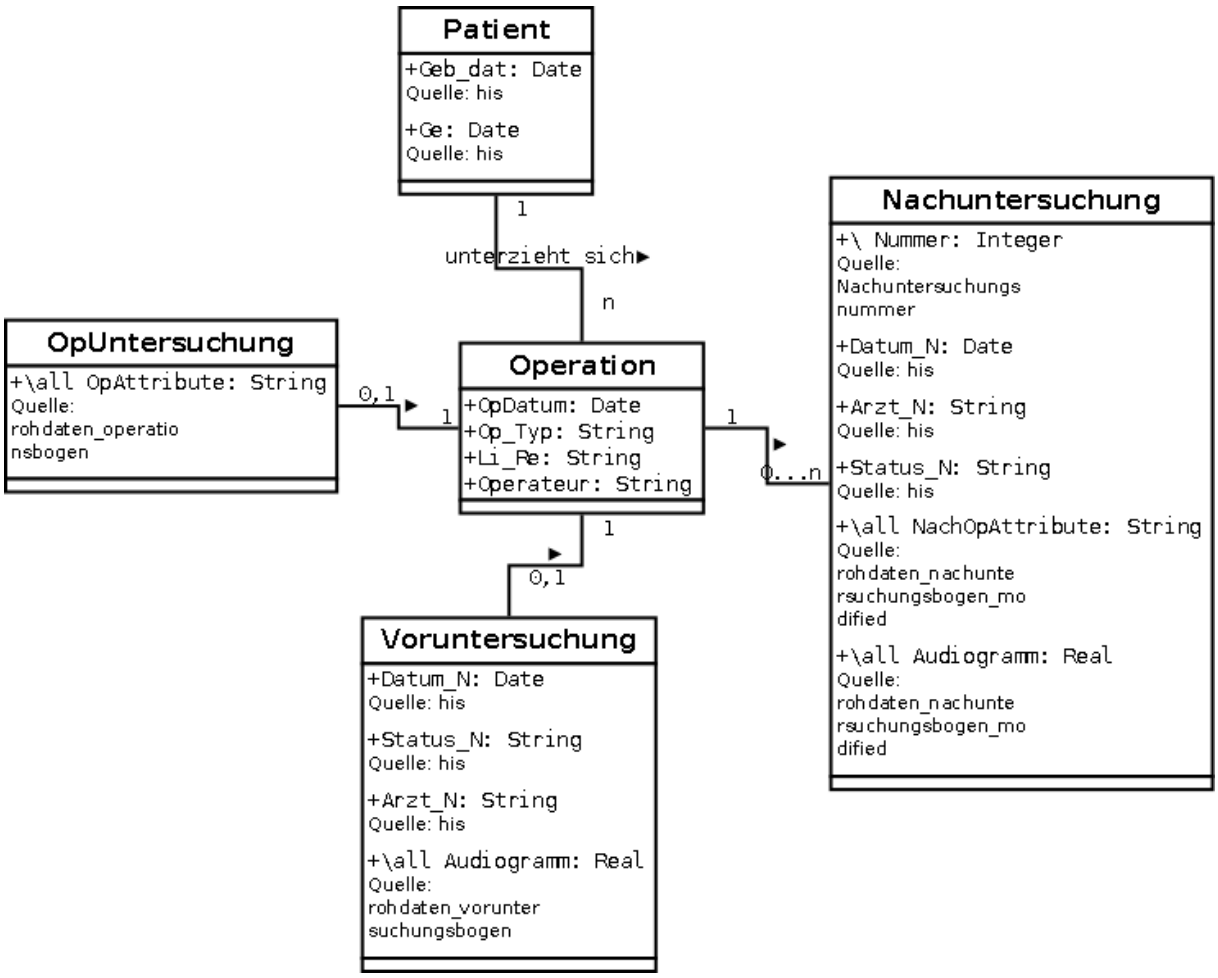


Abb. 3.2: Beispiel des Diagramms eines ER-Modells

es nötig sein, sie anzupassen. Dabei helfen Hintergrundwissen zur Domäne und den Daten. Außerdem spielt Erfahrung ein große Rolle, denn je nach Anforderung sind verschiedene abgeleitete Attribute geeignet. Folgende Aufzählung gibt einige Beispiele:

- Bei Zeitgebenden Attributen kann es z.B. nötig sein, anstatt der absoluten Zeit eine relative Zeit – z.B. Anzahl Tage – zu einer anderen Zeit anzugeben. Zeitgebende Attribute können auch in mehrere separate Numerische Attribute, z.B. „Jahr“, „Monat“, „Tag“ aufgeteilt werden, wodurch z.B. Zyklen wie „jeder dritte Tag im Monat“ deutlich werden, die ansonsten verborgen bleiben.
- Numerische Attribute können in aussagekräftigen Kategorischen Attributen diskretisiert werden.
- Ein Kategorisches Attribut mit einer Vielzahl von Werten kann durch ein neues Attribut, welches verschiedene Attributwerte mit einem Wert zusammenfasst, ausgedrückt werden.

Genauso können mehrere Attribute in einem neuen Attribut zusammengefasst werden, Han und Kamber [32, S. 94] nennen dies „Concept Hierarchy Generation“. Ein Kategorisches Attribut kann auch in ein Numerisches umkodiert werden, um eine feste Reihenfolge zu erhalten.

Aber nicht nur Attribute können hinzugefügt und entfernt werden. Unter Umständen kann es sinnvoll sein, einzelne Instanzen zu filtern, zum Beispiel wenn es sich bei ihnen um Fehler oder Ausreißer handelt, die durch das Modell nicht speziell beschrieben werden müssen (vgl. [50, S. 65]). Dies kann zur Performanz und Interpretierbarkeit beitragen, allerdings nur, wenn die Filterung explizit dokumentiert und später – zumindest, wenn der Verständlichkeit oder Glaubhaftigkeit dienlich – dem Entscheidungsträger mitgeteilt wird.

Die Umsetzung eines ER-Modell sieht folgendermaßen aus: Für jedes Objekt wird eine Tabelle, für jedes Attribut des Objekts wird eine Spalte in der Tabelle und für jede Instanz des Objekts wird eine Zeile in der Tabelle erstellt. Auch die Schlüsselattribute werden zu eigenen Spalten in der jeweiligen Tabelle. Primärschlüssel werden aus Performanzgründen als solche in der Datenbank gekennzeichnet. Jeder Fremdschlüssel erhält in der Datenbank einen sog. Index, damit Abfragen auf dem ER-Modell, die den Verbund benötigen, schneller bearbeitet werden. Außerdem dient es der Geschwindigkeit von Abfragen, Fremdschlüssel, die aus mehreren Attributen bestehen, zu einem einzigen Numerischen Attribut umzuwandeln.

3.4.2 Multidimensionales Modell

Um bestimmte Anforderungen aus dem Business-Case zu erfüllen, kann es nötig sein, aufbauend auf den Informationen aus dem ER-Modell, ein *Multidimensionales Modell* zu entwickeln. Dieses bietet eine andere Sicht auf die Daten: *Data-Cubes*.

Ein Data-Cube kann als beliebig dimensionierter *Würfel* verstanden werden, der ein Objekt beschreibt, ein sog. Fakt. Ein Fakt besitzt erstens Dimensionsattribute. Mengen an Dimensionsattributen werden in ein oder mehrere Hierarchien aus Levels angeordnet und beschreiben eine Dimension des Fakts. Der höchste Level beschreibt stets die Gesamtheit der Werte einer Dimension und wird mit „all“ bezeichnet. Die Kombination aus Attributen des jeweils tiefsten Levels der Dimensionen definiert die kleinste Einheit des Data-Cube, Instanzen des Fakts, einzelne Fakten. Auf höhere Levels der Dimensionen betrachtet, werden Fakten zu Zellen im Data-Cube zusammengefasst. Ein Fakt besitzt zweitens Kennzahlattribute. Diesen wird jeweils ein Aggregationsoperator zugeordnet. Typische Operatoren sind: Maximalwert (Max), Minimalwert (Min), Anzahl an gesetzten Werten (Count), Anzahl an gesetzten, unterschiedlichen Werten (Distinct-Count), Summe (Sum) und Durchschnitt (Avg). Für jede Zelle im Data-Cube kann somit ein aggregierter Wert des Kennzahlattributs berechnet werden. Abbildung 3.3 zeigt ein Beispiel. Darin sind die Dimensionen „Time“, „Source“ und „Route“ mit jeweils einer Hierarchie

aus drei Levels dargestellt. Das Fakt besitzt die Kennzahlen „Packages“, das die Anzahl an Paketen zählt und „Last“, das den Zeitpunkt des letzten Pakets nennt. Für jede Zelle, beschrieben durch Levels der Dimensionen, lassen sich diese Kennzahlen angeben[2].

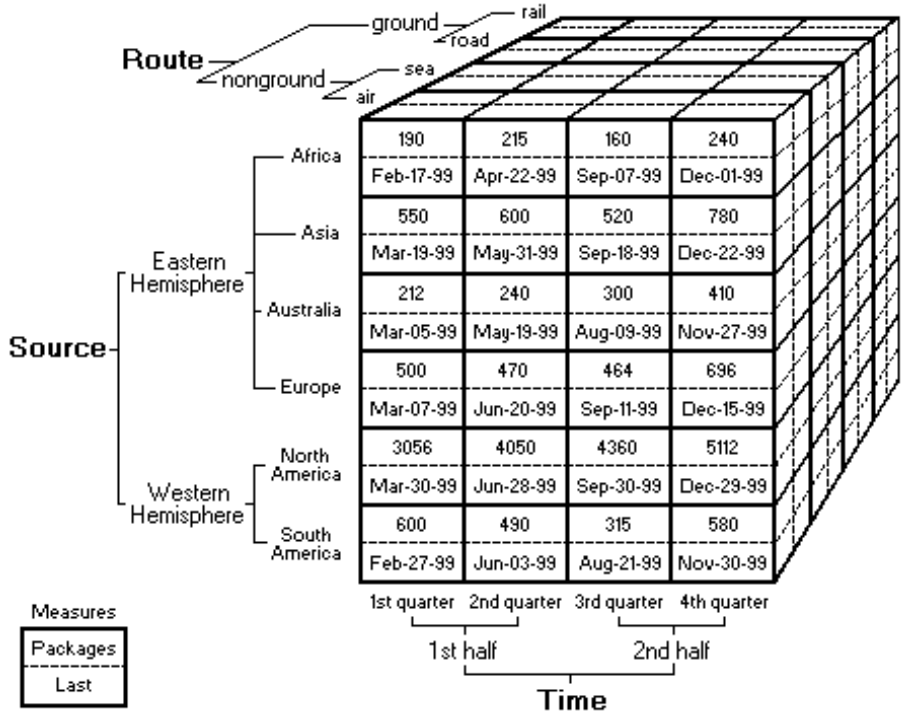


Abb. 3.3: Veranschaulichung Data-Cube[2]

Ein Multidimensionales Modell ist häufig die Basis eines Data-Warehouse (vgl. [47]); auch ich möchte es aus folgenden Gründen verwenden:

- Auch ein MD-Modell kann in einem Diagramm übersichtlich dargestellt und konzeptioniert werden (vgl. [47]).
- Im Gegensatz zum ER-Modell ist das Prinzip von Datenwürfeln im MD-Modell leicht verständlich. Data-Cubes ermöglichen die Abfrage natürlicher und bedeutungsvoller Teilmengen (vgl. [57]).
- Auch das ER-Modell unterstützt die Aggregation, dies jedoch nur eingeschränkt (vgl. [19]); komplexere Aggregationen sind ohne verschachtelte und komplizierte SQL-Abfragen nicht möglich. MD-Modelle vereinfachen das Verständnis von Aggregation, ermöglichen mit MDX auch komplexe Aggregationen in übersichtlichen Abfragen. Ein Data-Cube wird desweiteren so gespeichert, dass Aggregationen über ihre Kennzahlen schnell berechnet werden können.

[2]Quelle: Microsoft, http://msdn.microsoft.com/en-us/library/aa216772%28SQL.80%29.aspx, Dezember 2009

- Ein ER-Modell beschreibt ausschließlich eine Gesamtansicht. Ein MD-Modell beschreibt mit jedem Data-Cube eine bestimmte Sicht auf die Daten, die genauestens untersucht werden kann. Dennoch können auch mehrere Data-Cubes gemeinsam abgefragt und aufeinander bezogen werden.
- MD-Modelle bieten die Möglichkeit, die Daten interaktiv zu betrachten.

Auch ein MD-Modell lässt sich mit einem Klassendiagramm darstellen (vgl. [47]). Darin werden Fakten, Dimensionen, Levels und Kennzahlen als Klassen dargestellt. Hierarchien werden als gerichteter Graph ohne Zyklen dargestellt (vgl. [19]). Abbildung 3.4 zeigt ein Beispiel, in dem Audiogramme als Fakten betrachtet werden. Mengen an Fakten können durch den Operationstyp, das Untersuchungsdatum und das Geschlecht des Patienten ausgewählt werden. Das Untersuchungsdatum kann dabei in drei verschiedenen Granularitätsstufen zur Auswahl verwendet werden: Jahr, Monat und Tag. Für jede Menge an Fakten lässt sich der Durchschnitt einer Kennzahl berechnen, die die Gehörverbesserung seit der letzten Untersuchung angibt.

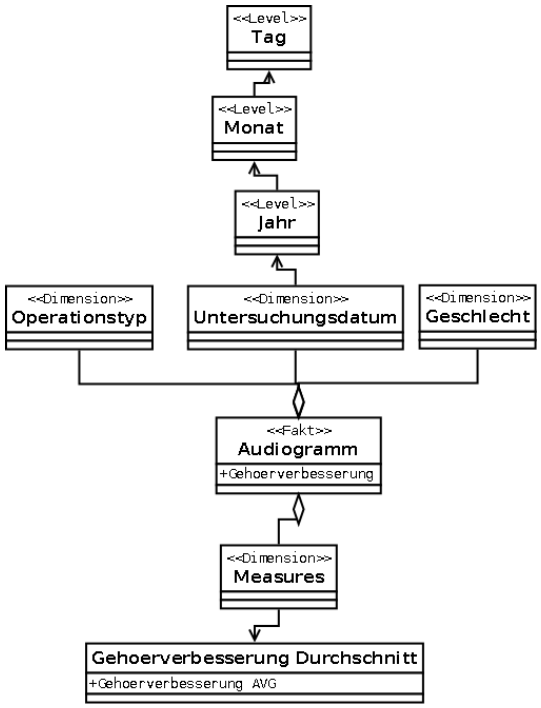


Abb. 3.4: Beispiel des Diagramms eines MD-Modells

Es gibt auch Multidimensionale Normalformen (vgl. [44]). Die erste Normalform schreibt vor, dass jede Eigenschaft des Anwendungskontexts im Data-Cube abgebildet ist; für die Methodologie bedeutet dies, dass die Objekte, Attribute und Beziehungen des ER-Modells korrekt abgebildet sein müssen. Außerdem ist der ersten Normalform nach jede Kennzahl eines Fakts von der Gesamtheit der Dimensionsattribute abhängig, was Fehler bei der Berechnung und Interpretation von Kennzahlen vermindert.

Angelehnt an Han und Kamber [32] und Song et al. [64] ist die Vorgehensweise zum Erstellen des MD-Modells für jede Anforderung:

1. Auswählen eines der Objekte aus dem ER-Modell als zentrales Fakt.
2. Auswählen von Mengen von Attributen der Objekte als Dimensionen.
3. Auswählen von Hierarchien der Attribute, jede Hierarchiestufe bzw. Level beschreibt eine Granularität und besteht aus einem Attribut.
4. Auswählen von Attributen der Objekte als Kennzahlen, in höheren Levels aggregiert mittels eines festgelegten Aggregationsoperators.

Eine verständliche und performante Möglichkeit, um m:n-Beziehungen im ER-Modell über ein Multidimensionales Modell abzubilden, besteht darin, für beide Objekte Fakten zu definieren, die Dimensionen gemeinsam besitzen (vgl. [47]). Über diese *Gemeinsamen Dimensionen* können die Kennzahlen beider Data-Cubes abgefragt werden.

Kennzahlen, wie sie in diesem Stadium definiert werden, lassen sich allein aus Aggregation von Kennzahlenattributen definieren. Später im Reporting können weitere Kennzahlen definiert werden, die von solchen Ursprungskennzahlen abgeleitet werden.

Anhand der identifizierten Fakten, Dimensionen und Kennzahlen leiten sich nötige Data-Cubes ab.

Die einfachste Form der Umsetzung eines Data-Cube in einer relationalen Datenbank erfolgt über ein sog. Sternschema (vgl. [32, S. 214]). Darin werden sowohl das Fakt, als auch die Dimensionen jeweils als eigene Tabellen umgesetzt – Faktentabellen und Dimensionstabellen. Die Levels der Dimensionshierarchien werden als Attribut der jeweiligen Dimensionstabelle umgesetzt. Die Kennzahlen werden als Attribute der Faktentabelle umgesetzt. Die Faktentabelle besitzt für jede Dimension einen Fremdschlüssel, zugehörig zum Primärschlüssel der Dimensionstabelle. So kann das MD-Modell als spezielles ER-Modell umgesetzt werden, in dem für jeden Data-Cube ausschließlich n:1-Beziehungen zwischen Fakt und den Dimensionen auftreten.

Das Vorgehen besteht für jeden Data-Cube aus der Abfrage aller Instanzen und relevanten Attribute (Dimensions- und Kennzahlattribute) des Objekts, das das Fakt darstellen soll. Für jede Menge an Attributen einer Dimension werden die Werte jeder Instanz des Fakts in eine Dimensionstabelle eingegeben und stattdessen ein Fremdschlüssel erstellt. Die Fremdschlüssel und Kennzahlattribute werden anschließend in eine Faktentabelle eingelesen. Auf die Fremdschlüssel in Faktentabelle und Dimensionstabellen werden Indices eingestellt. Genauso werden die Primärschlüssel – jede Dimensionstabelle besitzt normalerweise einen, der gleichzeitig ein Fremdschlüssel zur Faktentabelle ist – als solche in der Datenbank deklariert.

Um die Richtigkeit des ER- und MD-Modells zu überprüfen, können nach der Umsetzung ein-

fache Abfragen erstellt und mit Hintergrundwissen der Experten sowie Informationen aus dem Data-Assay verglichen werden. Andersherum bietet das Data-Warehouse nach seiner Umsetzung die Möglichkeit der Evaluation bisheriger Annahmen. Es lässt sich einfacher und umfassender als die Rohdaten explorieren und so überprüfen, ob es die Anforderungen des Entscheidungsträgers erfüllen kann, siehe 2.4.2.3 aus dem Prozessmodell. Auch gibt es Metriken, um die Qualität eines Data-Warehouse zu bewerten, z.B. über die Anzahl an Fakten, Dimensionen oder Kennzahlen (vgl. [61]).

3.5 Reporting

Wie bereits in der Beschreibung der Anforderungen im Business-Case erläutert wurde, wird in dieser Arbeit angenommen, dass Data-Mining verständlicher für den Entscheidungsträger wird, wenn seine Anforderungen mit Berichten eingeleitet werden. Diese Berichte fassen jeweils einen zu analysierenden Teil der Daten zusammen, können weitergegeben und direkt vom Entscheidungsträger verstanden werden. Die Erstellung der Berichte ist ein Teil der Entwicklungsprozesse aus Abschnitt 2.4.3.2; auf sie werden anschließend Data-Mining-Techniken angewendet.

Jeder Bericht besteht aus einer Menge an Tabellen, jeweils mit einer bestimmten Anzahl an Spalten und Zeilen, deren Zellen bei Ausführung des Berichts mit Inhalt gefüllt werden. Jede Tabelle beschreibt damit Attribute und Instanzen. Ein Bericht wird um Zusatzinformationen wie einem Titel, dem Namen der Anforderung, dem Namen des Autors, dem Erstellungsdatum und dem Projektnamen ergänzt, um die Nachvollziehbarkeit zu verbessern. Die Inhalte des Berichts sollten weiterhin für den Entscheidungsträger verständlich sein, ohne dass technische Details aus dem Data-Assay und dem Data-Warehouse vorausgesetzt werden (vgl. [39]), weshalb ggf. Hinweise zur Interpretation des Berichts genannt werden.

Ein Bericht enthält dabei nicht den Inhalt der Daten. Stattdessen definiert er ausschließlich, welche Daten in ihm wie angezeigt werden sollen. Nachdem er entworfen und umgesetzt wurde, kann er auf die Daten ausgeführt und als Auszug in verschiedenen Dateiformaten (z.B. CSV, PDF) gespeichert werden. Selbst bei geändertem Inhalt des Data-Warehouse, z.B. nach einer Aktualisierung, soll ein Bericht funktionieren, ein neuer Auszug die veränderten Daten berücksichtigen.

Im Business-Case wurden die Berichte ausschließlich in der Sprache des Entscheidungsträgers beschrieben. Im Reporting gilt es, sie zu formalisieren und in Abfragen an die Datenbank zu übersetzen. Sowohl das ER-Modell, als auch das MD-Modell besitzen Abfragesprachen, die dies ermöglichen.

3.5.1 Abfrage des ER-Modells

Abfragen eines relationalen Schemas wie dem ER-Modell lassen sich über die „Relationale Algebra“ (vgl. [19]) formalisieren. Sie soll hier nicht behandelt werden, zu gering erscheint ihr Nutzen zur Konzeption von solchen Abfragen; dennoch nenne ich sie kurz, da ich für die Abfragen des MD-Modells, eines multidimensionalen Schemas, eine Algebra vorstellen werde. Ausdrücke in der Relationalen Algebra lassen sich in die *Structured Query Language* übersetzen; dieses *SQL* ist die Standard-Abfragesprache für relationale Daten, die von verschiedenen Datenbanken implementiert wird.

Quellcode 3.1 zeigt ein Beispiel einer typischen SQL-Abfrage, hier in der Implementierung von MySQL.

```
1 select objekt1.attribut1, if(objekt2.attribut1 is null,0, objekt2.attribut1),
      max(objekt3.attribut1)
2 from objekt1 left join objekt2 on objekt1.attribut2 = objekt2.attribut2
3 join objekt3 on objekt2.attribut3 = objekt3.attribut2
4 group by objekt1.attribut1, objekt2.attribut2
5 having max(objekt3.attribut1) > 100
6 order by objekt1.attribut1 desc
```

Quellcode 3.1: Typische SQL-Abfrage

Hierbei werden 3 Attribute von drei verschiedenen Objekten zur Anzeige ausgewählt (*select*). Das zweite Attribut erhält den Wert „0“, wenn das Attribut keinen Wert besitzt und *null* ist. Ein *Join* der Objekte geschieht über den Vergleich von Attributen. Schließlich werden Instanzen, die die ersten zwei Attributwerte gemeinsam haben, gruppiert. Für das dritte Attribut wird daher nicht der jeweilige Wert der Instanz, sondern aus den Werten einer Gruppe der Maximalwert bestimmt. Angezeigt werden schließlich nur die Gruppen, bei denen der Maximalwert über 100 liegt. Letztendlich werden die ausgegebenen Gruppen absteigend nach dem Wert des ersten Attributs sortiert – bei textuellen Attributen alphabetisch. Man beachte, dass je nach Implementierung des *SQL*-Standards eine Reihe von weiteren Funktionen wie „Concat“ zur *Konkatenation* von Zeichenketten oder „Count“ zum Zählen von Instanzen einer Gruppe zur Verfügung stehen.

Views stellen eine Möglichkeit dar, um SQL-Abfragen vorzudefinieren und ihre Ausgabe vergleichbar mit Tabellen zugreifbar zu machen. Ihr Vorteil besteht darin, häufige Abfragen ohne zusätzlichen Speicherverbrauch zu speichern.

Weitere Informationen zu SQL finden sich bei den Herstellern von Implementierungen der Sprache[3].

[3] z.B. Referenzhandbücher zu MySQL, http://dev.mysql.com/doc/#refman, Dezember 2009

3.5.2 Abfrage des MD-Modells

Chen et al. [19] haben eine Algebra entwickelt, mit der sich Abfragen auf multidimensionale Daten formalisieren lassen. Ein Beispiel einer Abfrage, zunächst in Prädikatenlogik:

$$\forall f \in [quelle : IP, zeit : stunde] \in Netzwerkpakete,$$
$$f.Anzahl = |coverage(c)|$$

Darin werden für alle IP-Adressen der Quellen und alle Stunden, in der sie verschickt wurden, die Anzahl an Netzwerkpaketen gezählt.

Bei der Beschreibung des MD-Modells in Abschnitt 3.4.2 habe ich Kennzahlen, die aus anderen Kennzahlen abgeleitet werden, bereits erwähnt. Solche *Abgeleiteten Kennzahlen* sind sehr mächtig, so können beispielsweise Kennzahlen aus verschiedenen Granularitäten in Formeln kombiniert werden; allerdings sind solche Abfragen aufwändiger – sowohl für die Person, die sie konzeptioniert, als auch die Datenbank, die sie berechnet. Der Hauptzweck der Algebra „AW-RA“ besteht darin, diese „composite subset measures“ leichter und performanter zu definieren.

Im Folgenden möchte ich sie kurz anschneiden, eine ausführliche Erklärung findet sich im Werk von Chen et al. [19].

Das Schema eines multidimensionalen Datensatzes D mit d Dimensionen besitzt Dimensionsvektor $X = (X_1, X_2, \ldots, X_d)$, außerdem beliebig viele Kennzahlen. Jedes Fakt r aus D ist als Tupel von Dimensionswerten sowie Kennzahlwerten $(x_1, x_2, \ldots, x_d, m_1, \ldots)$ beschrieben.

Jede Dimension besitzt eine Grundmenge mit möglichen Attributwerten. Diese werden in Hierarchien generalisiert. $D_i <_D D_j$ bedeutet, dass D_j genereller als D_i ist – eine höhere Granularität besitzt. Typischerweise besitzt jede Dimension ein höchstes Level „all“, das die gesamte Grundmenge in einem Attribut „ALL“ zusammenfasst.

Eine Zelle c im Data-Cube umfasst eine Menge an Fakten und kann als Tupel $c = (v_1, v_2, \ldots, v_d)$ aus Attributwerten jeder Dimension beschrieben werden. Dimensionen im höchsten Level können aus Gründen der Übersichtlichkeit weggelassen werden. Eine Zellmenge $S = [X_1 : D_1, \ldots, X_d : D_d]$ besteht aus der Menge an Zellen in einem Data-Cube mit gemeinsamer Granularität der Dimensionen.

Chen et al. [19] zeigen zunächst, dass sich eine Abfrage auf multidimensionale Daten in „intuitive calculus formulas“ darstellen lässt. Mittels relationaler Algebra abgefragt, sind diese Abfragen auf Grund vieler verschachtelter Operationen (insbesondere dem Verbund) schwer zu verstehen und optimieren. Solche Abfragen in Prädikatenlogik lassen sich in der Algebra „AW-RA“ formulieren.

Die wichtigsten Operatoren in AW-RA sind wie folgt:

Selektion Die Auswahl von Fakten aus einer Zellmenge, die eine Bedingung (cond) erfüllen, über $\sigma_{cond}(T)$ mit Zellmenge $T \in AW - RA$ und $AW - RA$ als Menge aller Ausdrücke.

Aggregation Die aggregierte Berechnung von Kennzahlen aus einer Menge an Fakten über $g_{G,agg}(T)$ mit $T \in AW - RA$ und die Zellmenge G besitzt eine höhere Aggregationsstufe als die Zellmenge T, kann also zur Aggregation verwendet werden.

Verbund Die Verbindung zweier Zellmengen und Berechnung einer Aggregation über $S \bowtie_{cond,agg} T$ mit $S, T \in AW - RA$ und S befindet sich nicht auf der höchsten Aggregationsstufe, kann also zur Aggregation verwendet werden.

Die Algebra kann leicht auf SQL übertragen werden. Dies ist jedoch nicht notwendig, da, wie bereits erklärt, SQL für die Aggregation nicht besonders gut geeignet ist.

Im Folgenden wird das obige Beispiel fortgeführt und für jede Stunde die durchschnittliche Anzahl an Paketen aller Quellen berechnet. Dazu wird zunächst für jede IP-Adresse und Stunde die „Anzahl“ an Netzwerkpaketen berechnet. Die Zellmenge „Base“ beschreibt die Zellen, für die die Kennzahl „Schnitt“ berechnet werden soll: Einzelne Stunden. Durch einen Verbund der beiden Zellmengen wird schließlich für jede Stunde der Durchschnitt an Paketen aller IP-Adressen berechnet:

$$Anzahl = g_{(quelle:IP,zeit:stunde),count(*)}(Netzwerkpakete)$$

$$Base = g_{(zeit:stunde),0}(Netzwerkpakete)$$

$$Schnitt = Base1 \bowtie_{Base1.zeit=Anzahl.zeit,avg(Anzahl.M)} Anzahl$$

Für komplexe Abfragen ist diese Algebra nicht intuitiv genug, weshalb die Autoren mit „Aggregation Workflows“ [19] eine Möglichkeit vorstellen, um komplexe Abfragen zu konzeptionieren. Sie bestehen aus einem Graph mit drei Arten von Elementen:

Rechtecke stellen Zellmengen dar. Sie können als Klassen mit dem Namen des verwendeten Data-Cube und Attributen zur Kennzeichnung der verwendeten Aggregationsstufe dargestellt werden. Auch hier werden Dimensionen auf der höchsten Aggregationsstufe nicht erwähnt.

Ovale stellen Kennzahlen dar. Darin enthalten sind der Name der Kennzahl, die Aggregationsformel und ggf. eine Selektionsbedingung. Jedes Oval ist über eine Kante mit einem Rechteck verbunden, das beschreibt, für welche Zellmenge die Kennzahl berechnet wird.

Pfeile zwischen Ovalen stellen Abhängigkeiten zwischen Kennzahlen dar und werden über eine Verbundbedingung beschrieben.

Abbildung 3.5 zeigt die Aggregation Workflows des obigen fortgeführten Beispiels. Von Unten nach Oben aufgebaut, wird darin zunächst für jede IP-Adresse und Stunde die Anzahl an Paketen berechnet. Jede Stunde aus einer zweiten Zellmenge „Base“ wird nun mit den Paketzahlen der IP-Adressen der selben Stunde verbunden und der Durchschnitt berechnet.

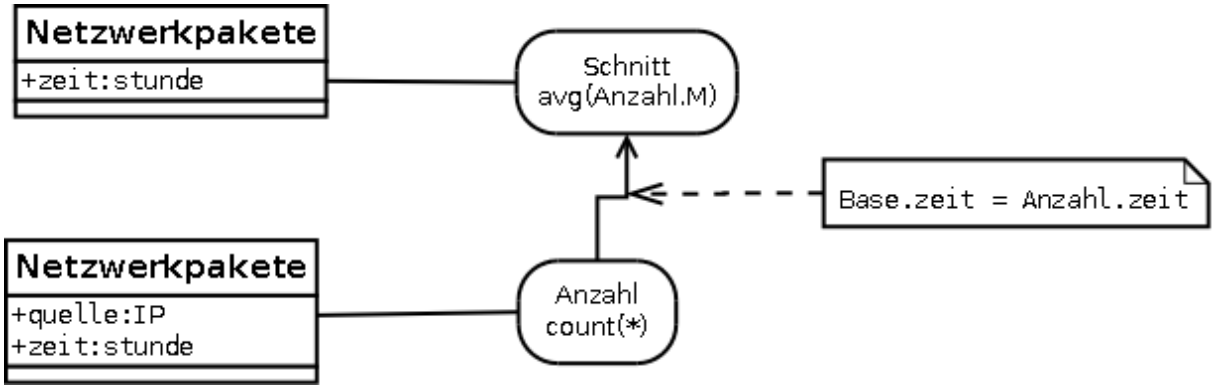


Abb. 3.5: Aggregation Workflows Beispiel

Mittels *Multidimensional Expressions* (*MDX*), einer von Microsoft entwickelten Standard-Abfragesprache für multidimensionale Daten, kann eine Abfrage auf das MD-Modell schließlich umgesetzt und Inhalte für einen Bericht ausgelesen werden. Ein Beispiel für eine typische MDX-Abfrage, wird in Quellcode 3.2 gezeigt.

```
1  with member [Measures].[AbgeleiteteKennzahl]
2  as '[Measures].[Kennzahl_1]/[Measures].[Kennzahl_2]'
3  , FORMAT_STRING = "#,###.00%"
4
5  select NON EMPTY
6  {[Measures].[Kennzahl_1], [Measures].[Kennzahl_2], [Measures].[AbgeleiteteKennzahl]} ON
       COLUMNS,
7  NON EMPTY
8  Order(Crossjoin([Dimension1].[Level1].children, [Dimension2].[Level1].children),
9  [Measures].[AbgeleiteteKennzahl], DESC)
10 ON ROWS from [Datenwuerfel]
11 where ([Dimension3].[Level2].[Wert1])
```

Quellcode 3.2: Typische MDX-Abfrage

Darin wird die abgeleitete Kennzahl „AbgeleiteteKennzahl“ als Anteil der „Kennzahl_1“ von „Kennzahl_2“ berechnet und als Prozentzahl dargestellt. Die Abfrage gibt für jede Kombination

aus den Elementen des jeweils ersten Levels der Dimensionen „Level1“ und „Level2“ diese drei Kennzahlen aus und sortiert die Werte absteigend nach der abgeleiteten Kennzahl. Außerdem werden in dieser Abfrage nur Fakten berücksichtigt, die im zweiten Level von „Dimension3“ den Wert „Wert1“ besitzen. Zu erwähnen ist noch, dass der Operator NON EMPTY Zeilen oder Spalten mit ausschließlich leeren Kennzahlen filtert.

MDX ist anfangs nicht einfach zu erlernen. Am Besten helfen dabei konkrete Beispiele, wie sie in den Einzelfallstudien dieser Arbeit zu finden sind. Es handelt sich jedoch um eine sehr mächtige Sprache. Im Gegensatz zu SQL kann mit MDX beispielsweise aus einer Abfrage, in Abhängigkeit der Daten, eine variable Anzahl an Spalten erhalten werden.

Weitere Informationen zu MDX finden sich in der Spezifikation von Microsoft[4] bzw. in Handbüchern (z.B. [67]).

Nachdem die Abfrage entworfen worden ist, kann ein Bericht erstellt werden. Dieser kann anschließend beliebig oft ausgeführt und Auszüge in verschiedenen Formaten (z.B. CSV, PDF, XLS) ausgegeben werden. Auf die Berichte kann über eine webbasierte Benutzerschnittstelle mit Authentifizierung zugegriffen werden. Das vereinfacht die Erstellung von Auszügen zur Weitergabe und Analyse und verteilt die Berechnungslast auf mehrere Rechner.

Neben ihren Abfragesprachen, besitzen ER- und MD-Modell jeweils die Möglichkeit des Entscheidungsträger-verständlichen Zugangs zu den Daten. Das erscheint insbesondere realistisch, wenn eine begrenzte, dafür jedoch intuitiv bedienbare Zugriffsmöglichkeit geboten wird.

Für das ER-Modell bietet es sich an, die Attribute sowie Sortierungen und Filter in einem „graphical user interface“ (vgl. [32, S. 10]) auswählen und in einer Tabelle anzeigen zu lassen. So können auch Attribute aus verschiedenen Objekten in Abhängigkeit ihrer Beziehung im Modell zusammen angezeigt werden – zumindest ohne Aggregation verständlich für einen Entscheidungsträger.

Das MD-Modell erlaubt darüber hinaus die interaktive Betrachtung der Daten. In diesem sog. *Online Analytical Processing* (OLAP) (vgl. [32, S. 132f]) kann der Data-Cube über seine Dimensionen und ihre Hierarchien durchlaufen werden. Möglich sind unter anderem: Das Betrachten der Daten auf einer höheren Aggregationsstufe, z.B. über Auswahl eines höheren Levels einer Dimension. Das Gegenstück zu einem solchen *Drill-Down* ist der *Drill-Up*. Außerdem kann das Festlegen einer Dimension auf den Wert eines Level, das *Slicing*, die Sicht auf bestimmte Teile der Daten ermöglichen. OLAP ist nach den Regeln von E.F. Codd speziell dafür geeignet, den geregelten Zugriff verschiedener Personen über eine Client-Server-Architektur zu ermöglichen (vgl. [3]).

Die Berichte stellen genauso wie das Data-Assay und das Data-Warehouse einen Meilenstein dar. Einerseits dienen sie dem Team zur Evaluation der bisherigen Ergebnisse, siehe Abschnitt

[4]SQL Server Developer Center, http://mondrian.pentaho.org/documentation/doc.php, Dezember 2009

2.4.2.3. Desweiteren sind sie ein Teil des Deployments aus Abschnitt 2.4.3.3 und werden dem Entscheidungsträger abgeliefert.

3.6 Data-Mining

Ziel des Data-Mining, der letzten Aufgabe der Entwicklungsprozesse aus Abschnitt 2.4.3.2, ist das Entdecken von Mustern in den Berichten, wie sie im Business-Case angefordert werden. Pyle [53, S. 380] fasst das Entdecken solcher Muster mit dem Begriff „Modeling to Understand“ zusammen. Demnach besteht der Zweck des Data-Mining darin, die „Repräsentation der Wirklichkeit“ [53, S. 91], die im Data-Assay und Data-Warehouse aufgebaut worden ist, auf Muster zu untersuchen.

In dieser Arbeit sind Muster als Sachverhalte in Berichten definiert, die von einem Entscheidungsträger verstanden und in Fragen formuliert werden können. Im Deskriptiven Data-Mining auf Tabellarische Daten, wie es Thema dieser Arbeit ist, wurden folgende Grundtypen von Mustern identifiziert:

- Muster charakterisieren Objekte oder Attribute.
- Muster benennen Unterschiede oder Gemeinsamkeiten zwischen Objekten oder Attributen.
- Muster beschreiben Beziehungen zwischen Objekten oder Attributen.
- Muster identifizieren ungewöhnliche Objektinstanzen oder Attributwerte.

Wie ein Bericht, besitzt auch ein Muster zunächst keine Inhalte, sondern beschreibt lediglich, welche Informationen es enthalten kann. In den Daten wird dann nach konkreten Mustern gesucht. Ein interessantes konkretes Muster ist über die Daten hinaus gültig, lässt sich auf die Wirklichkeit übertragen und bietet im Bezug auf die behandelte Fragestellung eine nützliche Information.

Dabei handelt es sich stets um Informationen, die ein Entscheidungsträger auch manuell in einem Bericht suchen oder überprüfen kann, es aus Gründen des Aufwands jedoch nicht tun möchte. Stattdessen wird eine Technik angewendet, um Muster aus einem Bericht zu extrahieren und Entscheidungsträger-verständlich darzustellen. Zur Auswahl solcher Techniken habe ich folgende Kriterien ausgewählt:

Nachvollziehbarkeit Die Technik erfordert vom Entscheidungsträger kein Verständnis darüber, wie das Ergebnis erreicht wird; denn der Entscheidungsträger weiß, wie er manuell nach dem Muster suchen könnte.

Ausführbarkeit Die Technik lässt sich ohne inhaltliche Änderung auf den Bericht ausführen. Wenn dem Bericht Informationen hinzugefügt oder entfernt werden, soll dies über eine neue

Erstellung eines Berichts geschehen, was der Interpretierbarkeit der entdeckten konkreten Muster dient (vgl. [50, S. 59]).

Interpretierbarkeit Die Technik erfordert vom Entscheidungsträger zur Interpretation der konkreten Muster kein technisches Hintergrundwissen, z.B. die Erklärung von Parametern.

Verfügbarkeit Außerdem wird Wert darauf gelegt, dass die Technik fest definierte und bekannte Algorithmen anwendet, die anhand ihres Namens identifiziert und deren technischen Beschreibungen in Nachschlagewerken gefunden werden können. Diese werden von vielen Werkzeugen unterstützt.

Ich habe mittels dieser Kriterien eine Reihe von Techniken ausgewählt, die im Entscheidungsträger-verständlichen Data-Mining eine Rolle spielen können. Im Folgenden möchte ich jede dieser Techniken kurz beschreiben, anschließend in einer Tabelle die Auswahl anhand der Kriterien begründen. Ausführlichere Erklärungen der Techniken finden sich in der Literatur (vgl. [38]; [32]; [50]).

3.6.1 Diagramme, Kennzahlen und Hervorhebungen

Eine einfache Möglichkeit, um Inhalte aus einem Bericht hervorzuheben, stellt die Sortierung über ein oder mehrere Attribute dar. Außerdem die Berechnung von Kennzahlen über einzelne Attribute, z.B. Summe oder Anzahl. Neben Sortierungen bieten sich speziell für generierte Berichte aus dem MD-Modell weitere Hervorhebungen an. So können Attributwerte farblich hervorgehoben werden, wenn sie im Data-Cube eine „Anomalie“ (vgl. [32, S. 189f]) darstellen. Han und Kamber nennen mit „SelfExp“, „InExp“ und „PathExp“ drei Möglichkeiten, um solche Abweichungen auf verschiedenen Granularitätsstufen festzustellen.

Besonders typisch, um Eigenschaften von Objekten und Attributen festzustellen, sind Diagramme. Einerseits können sie verwendet werden, um einzelne Attribute zu beschreiben (vgl. [32, S. 54-61]):

Ein *Box-Plot* stellt für ein Numerisches Attribut Minimum, 1. Quartil (25% der Daten, die kleiner als der Median sind), Median (50% der Instanzen besitzen einen kleineren/größeren Attributwert als der Median), 3. Quartil (25% der Instanzen mit Attributwerten, die größer als der Median sind) und Maximum dar. Außerdem den Durchschnitt, den mittleren Wert. Die Varianz beschreibt, wie stark die Werte vom Durchschnitt abweichen. Und auch die Standard-Abweichung ist eine Kennzahl über die Verteilung der Werte um den Mittelwert.

Eine *Verteilung* gibt die Häufigkeit der Werte eines Numerischen Attributs an; somit den häufigsten Wert, Mode, und den seltensten Wert, Least, sowie den Skew, die *Schiefe* (bei symmetrischen Daten liegen Durchschnitt, Median und Mode übereinander; bei negativ oder positiv schiefen Daten liegt der häufigste Wert vor oder hinter dem Median) und die Kurtosis (ob die

Daten hauptsächlich um den Mittelwert angeordnet sind oder mehr außen liegen).

Ein *Histogramm* gibt die Häufigkeit der einzelnen Werte eines Kategorischen Attributs an, nennt also auch den häufigsten Wert, Mode, und den seltensten Wert, Least. Dabei werden modale (viele Werte in wenigen Kategorien bzw. wenige in vielen), gleichmäßige (jeder Wert annähernd gleich häufig) und monotone (jeder Wert genau einmal) Verteilungen deutlich.

Andererseits können durch Diagramme auch mehrere Attribute gleichzeitig visualisiert werden. Ein Punkt-Diagramm oder *Scatter-Plot* kann die Beziehung von bis zu vier (x-Achse, y-Achse, Größe, Farbe), ein 3-D-Punkt-Diagramm von bis zu fünf Numerischen, Kategorischen oder Zeitgebenden Attributen deutlich machen. *Säulendiagramme* (z.B. für Häufigkeiten), *Balkendiagramme* (z.B. für Rangfolgen), *Kurven- /Liniendiagramme* (z.B. für Zeitreihen), *Keisdiagramme* (z.B. für Anteile) sind weitere Möglichkeiten, um Daten zu inspizieren (vgl. [32, S. 56]). Mit ihnen können mehrere Attribute verglichen werden, indem z.B. in einem Balkendiagramm die Werte verschiedener Attribute aufeinander gestapelt angezeigt werden. Diagramme sind intuitiv verständlich; manche Implementierung ermöglicht zusätzliche Funktionen, wie z.B. einen *Jitter*, bei dem übereinanderliegende Punkte in einem Scatter-Plot leicht verschoben dargestellt werden.

Die Auswahlkriterien werden erfüllt: Es werden Charakteristiken von Objekten oder Attributen, Beziehungen zwischen Attributen sowie ggf. ungewöhnliche Objektinstanzen oder Attributwerte deutlich. Die Sachverhalte im Diagramm können im Bericht nachvollzogen und überprüft werden. Diagramme lassen sich auf Berichte direkt anwenden. Sie bedürfen keines Verständnisses darüber, wie die Visualisierung erstellt wird.

Diagramme können zudem der Darstellung von Ergebnissen anderer Techniken dienen.

3.6.2 Korrelationstabelle

Eine Korrelationstabelle enthält Korrelationskoeffizienten und drückt mit ihnen für je zwei Numerische Attribute deren Abhängigkeit voneinander in einer Zahl zwischen -1 und 1 aus. Je stärker zwei Attribute negativ korreliert sind, desto negativer die Zahl; je größer bzw. kleiner das eine Attribut, desto kleiner bzw. größer das andere Attribut. Liegt die Zahl näher an der 1, sind die zwei Attribute positiv korreliert – je größer bzw. kleiner das eine, desto größer bzw. kleiner das andere. Ein Wert um 0 bedeutet eine geringe Korrelation. Von einer starken Korrelation sprechen Nisbet et al. [50, S. 71], wenn der Wert größer als 0,9 bzw. kleiner als -0,9 ist. Die Interpretation einer Korrelationstabelle lässt sich in einer Visualisierung, die auffallende Korrelationskoeffizienten hervorhebt, vereinfachen. Zur Verdeutlichung einer einzelnen Korrelation zwischen zwei Attributen bietet sich ein Scatter-Plot an (vgl. [32, S. 68]).

3.6.3 Subgruppenentdeckung

Zur Einführung von Subgruppen ein Beispiel: Der Arzt eines Krankenhauses möchte aus einem Bericht über seine Patienten des letzten Jahrzehnts Mengen an Patienten mit gemeinsamen Eigenschaften finden, bei denen die Sterblichkeit deutlich erhöht ist.

Subgruppen sind Instanzen eines Objekts, die bestimmte Attributwerte gemeinsam haben und bezüglich eines bestimmten Zielattributs von der gesamten Menge an Instanzen, der Population, abweichen (vgl. [5, S. 17]). Die Güte der Abweichung wird durch eine „Qualitätsfunktion“ bewertet. Diese und weitere Parameter wie z.B. die maximale Größe von gemeinsamen Attributwertkombinationen einer Subgruppe bestimmen die Gründlichkeit der Suche und damit Laufzeit des Algorithmus. Nur eine vollständige und damit zeitaufwändige Suche garantiert, die Subgruppe mit der größten Abweichung zu finden. Unabhängig von diesen Parametern besitzen die entdeckten Subgruppen mit der Subgruppengröße, der Attributwertkombination als Beschreibung der Subgruppe, den Informationen zum Zielattribut in der Subgruppe und den Informationen zum Zielattribut in der Population korrekte Informationen, die zudem übersichtlich in einer Tabelle dargestellt werden können.

3.6.4 Lernen eines Entscheidungsbaums

Zur Einführung von Entscheidungsbäumen ein Beispiel: Ein Leiter einer Bank möchte aus einem Bericht, der Eigenschaften der Kunden sowie ihre Kredithistorie nennt, mögliche Erklärungen für negatives Kreditverhalten erhalten.

Ein Entscheidungsbaum ist ein ungerichteter Graph. Er beschreibt die Beziehungen zwischen mehreren Eingabeattributen und einem Klassenattribut (vgl. [50, S. 300]). In jedem seiner Knoten werden Mengen an Instanzen anhand des Werts eines ausgewählten Attributs voneinander getrennt. Jeder Knoten enthält eine Menge an Instanzen; diese wiederum besitzen eine bestimmte Verteilung der Werte des Ausgabeattributs. In den Blättern des Baums weisen die Instanzen entweder im Idealfall den gleichen Wert des Ausgabeattributs oder auch hier wieder eine Verteilung auf. Ein Entscheidungsbaum liefert somit für jede Instanz eine Erklärung für den Wert des Ausgabeattributs. Welches Attribut in welchem Knoten zur Trennung der Instanzen verwendet wird, ist abhängig vom verwendeten Algorithmus, daher stellt ein Entscheidungsbaum stets nur eine mögliche Erklärung für ein Ausgabeattribut dar. Dennoch stellt er eine häufig genutzte Technik dar, um Nachbarschaften zwischen Attributen und Objekten festzustellen. Das Prinzip, nach denen sie erstellt werden, ist nicht nur leicht verständlich, die Baumdarstellung bietet weiterhin eine intuitive Visualisierung des Ergebnis.

3.6.5 Lernen von Assoziationsregeln

Auch hier zur Einführung ein Beispiel: Aus einer Auflistung an verkauften Autos mit ihren mitverkauften Extras möchte ein Autohändler erfahren, welche Zusätze eines Autos vermehrt zusammen gekauft werden, um diese kostensparend in Bundles anzubieten.

Beim Lernen von Assoziationsregeln werden Zusammenhänge zwischen Attributen eines Objekts in Form von *Wenn-Dann*-Regeln gesucht (vgl. [38, S. 344]). Eine solche Regel besteht aus einer Attributwertkombination als Vorbedingung, sowie dem Wert eines Attributs als Nachbedingung. Das Maß der Gültigkeit einer Regel innerhalb der Instanzen eines Objekts kann durch verschiedene Zahlen ausgedrückt werden, z.B. „accuracy“ [38, S. 344], dem Verhältnis zwischen der Anzahl Instanzen, bei denen Vorbedingung und Nachbedingung gelten zur Anzahl Instanzen, bei denen zumindest die Vorbedingung gilt. Ähnlich wie bei der Subgruppenentdeckung kann durch Parameter bestimmt werden, wie gründlich nach Regeln gesucht wird. Jede entdeckte Regel steht für sich alleine und kann von einem Entscheidungsträger verstanden werden. Der Apriori-Algorithmus verlangt für seine Ausführung binäre Attribute, die nur die Werte „0“ und „1“ aufweisen. Kategorische oder Zeitgebende Attribute können in diese Form gebracht werden, indem aus jedem Attributwert ein neues binäres Attribut erstellt wird, das kennzeichnet, ob die Instanz den Wert besitzt. Numerische Attribute müssen zuvor durch Diskretisierung in Kategorische Attribute transformiert werden. Diese besondere Kodierung der Attribute kann bei der Erstellung des ER-Modells berücksichtigt werden.

3.6.6 Segmentierung

Auch für die Segmentierung (bzw. Clustering) nenne ich zunächst ein Beispiel: In einem Bericht der Kundendatei eines einzelnen Geschäftsbereichs möchte ein Unternehmer wohldefinierte Gruppen seiner Kunden identifizieren, um sie besser in speziellen Marketingstrategien ansprechen zu können.

Hierbei werden Instanzen in Segmente eingeteilt. Instanzen innerhalb eines Segments besitzen maximale, Instanzen aus verschiedenen Segmenten minimale Ähnlichkeit. Manche Implementierungen erfordern die Eingabe der Anzahl an Segmenten, die entdeckt werden sollen. Außerdem unterscheiden sie sich häufig in der Definition von Ähnlichkeit. Die einfachste Definition sieht ein binäre Ähnlichkeit vor – identische Attributwerte sind *ähnlich*, unterschiedliche Attributwerte sind *unähnlich* – und ist auch für den Entscheidungsträger verständlich. Im „K-Means-Algorithmus“ (vgl. [32, S. 402]) wird jedes Segment durch eine Mittelwertinstanz repräsentiert, mit der die zum Segment zugehörigen Instanzen bestimmt werden. Dieser Repräsentant kann zur Interpretation der Segmentierung verwendet werden.

In Tabelle 3.3 fasse ich die genannten Techniken zusammen und begründe ihre Auswahl anhand

der genannten Kriterien.

Technik	**nachvollziehbar**	**ausführbar**	**interpretierbar**	**verfügbar**
Diagramme, Sortierungen, Kennzahlen	Eigenschaften, Beziehungen und ungewöhnliche Instanzen oder Werte.	Je nach Diagramm nur Attribute eines bestimmten Datentyps.	Kann durch Zusatzfunktionen in Werkzeugen noch verstärkt werden.	z.B. Scatter-Plot, Box-Plot, Histogramm [32, S. 56]
Korrelationstabelle	Abhängigkeit zwischen Attributen.	Nur Numerische Attribute.	Hervorhebung stark abhängiger Attribute in Visualisierung. Verdeutlichung in Scatter-Plot.	z.B. „Pearson's product moment coefficient“ [32, S. 67]
Subgruppenentdeckung	Beziehungen zwischen Attributen.	Beliebiges Zielattribut. Erklärende Numerische Attribute müssen diskretisiert werden.	Nennung in Tabelle.	z.B. SD-Map [5, S.47], SD-Map* [4]
Entscheidungsbaumlernen	Beziehungen zwischen Attributen. Unterschiede und Gemeinsamkeiten zwischen Instanzen	Numerische Attribute müssen diskretisiert werden.	Darstellung als Baum.	z.B. ID3 [55], J48 [56], CART [12]
Assoziationsregellernen	Beziehungen zwischen Attributen.	Numerische Attribute müssen diskretisiert werden.	Wenn-Dann-Regeln.	z.B. Apriori-Algorithmus [2]
Segmentierung	Unterschiede und Gemeinsamkeiten zwischen Instanzen. Ungewöhnliche Instanzen.	Beliebige Attribute.	Repräsentanten der Segmente.	z.B. K-Means [32, S. 402]

Tab. 3.3: Data-Mining-Techniken

Exporte der Berichte sind meist über CSV-, PDF- oder XLS-Dateien möglich. PDF- und XLS-Dateien besitzen ein proprietäres Format und werden von wenigen Werkzeugen unterstützt. CSV stellt ein simples Dateiformat dar, das von vielen Werkzeugen eingelesen werden kann. Sein Nachteil besteht im Informationsverlust bei der Konvertierung: In den Rohdaten sind Numerische, Kategorische und Zeitgebende Attribute enthalten, ggf. mit Missing-Values (vgl. [32, S.

62]). Diese werden in der Datenbank als Integers oder Reals, Strings sowie Dates oder Datetimes gespeichert; Missing-Values werden durch *null* dargestellt. In der CSV-Datei werden diese Werte z.B. als „null“ abgelegt. Sie können nicht einfach durch „“ ersetzt werden, da dieser Wert dem expliziten Leerwert entspricht. In einer CSV-Datei werden alle Attributwerte als Zeichenketten ohne Typ gespeichert. Daher kann es notwendig sein, eine CSV-Datei in das ARFF-Dateiformat (vgl. [10]) zu konvertieren: Dieses spezifiziert den Wert „?“ als Missing-Value und besitzt für jeden Typ aus der Datenbank eine Entsprechung. Bei der Konvertierung der CSV-Datei in eine ARFF-Datei werden Missing-Values in „?“ umgewandelt, außerdem wird für jedes Attribut ein ARFF-Datentyp in Kopfzeilen der Datei festgelegt.

Nachdem von einer Technik konkrete Muster entdeckt worden sind, gilt es diese „Resultate“ (siehe Data-Mining in Abschnitt 2.4.3.2 des Prozessmodells) auf interessante Muster zu untersuchen. Diese lassen sich auf die Wirklichkeit übertragen und bieten eine nützliche Information im Bezug auf die Fragestellung, die in der Anforderung behandelt wurde. Interessante Muster können dann im Deployment, siehe Abschnitt 2.4.3.3, verwendet werden.

Bevor eine solche Resultatsanalyse vorgenommen werden kann, muss jedoch sichergestellt sein, dass die Anforderung vollständig erfüllt wurde und die ihrbezüglich durchgeführten Techniken im Data-Assay, Data-Warehouse, Reporting und Data-Mining keine Unklarheiten aufweisen. Daher ist auch im Schritt des Data-Mining die Evaluation der bisherigen Entwicklungen nötig, siehe Abschnitt 2.4.2.3 des Prozessmodells.

3.7 Dokumentations- und Wissensmanagementsystem

Nun beschreibe ich meinen Ansatz zur Dokumentation und zum Wissensmanagement in den Data-Mining-Projekten, die in den Abschnitten 2.4.1 und 2.4.2.2 als Aufgaben des Data-Mining genannt werden; dieser Ansatz besteht aus einem *KDDM-Wiki* und einem Dokumentenversionierungssystem.

Auch wenn Data-Mining schwer erlernbar ist, kann es dennoch erleichtert werden. Das Wissensmanagement, wie es in Abschnitt 2.4.1.1 genannt wird, bezieht sich dabei auf den Wissensaustausch zwischen einzelnen Data-Mining-Projekten. Data-Mining-Projekte können viele Informationen enthalten, die bei der Durchführung weiterer Projekte eine Rolle spielen. Als Hauptinformationsquelle für diese Informationen kann die Dokumentation der Projekte dienen. Je nach Dokumentationsstrategie ist es leichter, nötige Informationen zu extrahieren.

Problematisch ist die einfache Dokumentation mit einzelnen Dokumenten in zweierlei Hinsicht. Einerseits ist die Dokumentation an sich erschwert, können schwerlich Referenzen gebildet und Redundanzen vermieden werden. Dieser Meinung ist auch Becker [8], die darauf hinweist, dass die herkömmliche Methode mittels simplen Textdokumenten nicht ausreicht, weil sie beispielsweise

keine effiziente Querverweise unterstützt – diese jedoch essentiell sind. Außerdem ist das Suchen nach Informationen und damit das Wissensmanagement erschwert.

3.7.1 KDDM-Wiki

Nur wenn die Dokumentation der Data-Mining-Projekte formalisiert und vereinheitlicht ist, kann sie sinnvoll zum Wissensmanagement verwendet werden. Becker [8], Bartlmae und Riemenschneider [7], Euler [26] und Britos et al. [13] haben Möglichkeiten genannt, wie die Durchführung von Data-Mining-Projekten formalisiert und dokumentiert werden kann. Mit ihren Anstößen zum Thema habe ich eine eigene Objektstruktur entworfen. Ihre einzelnen Objekte werden in Abbildung 3.6 in einem Klassendiagramm dargestellt.

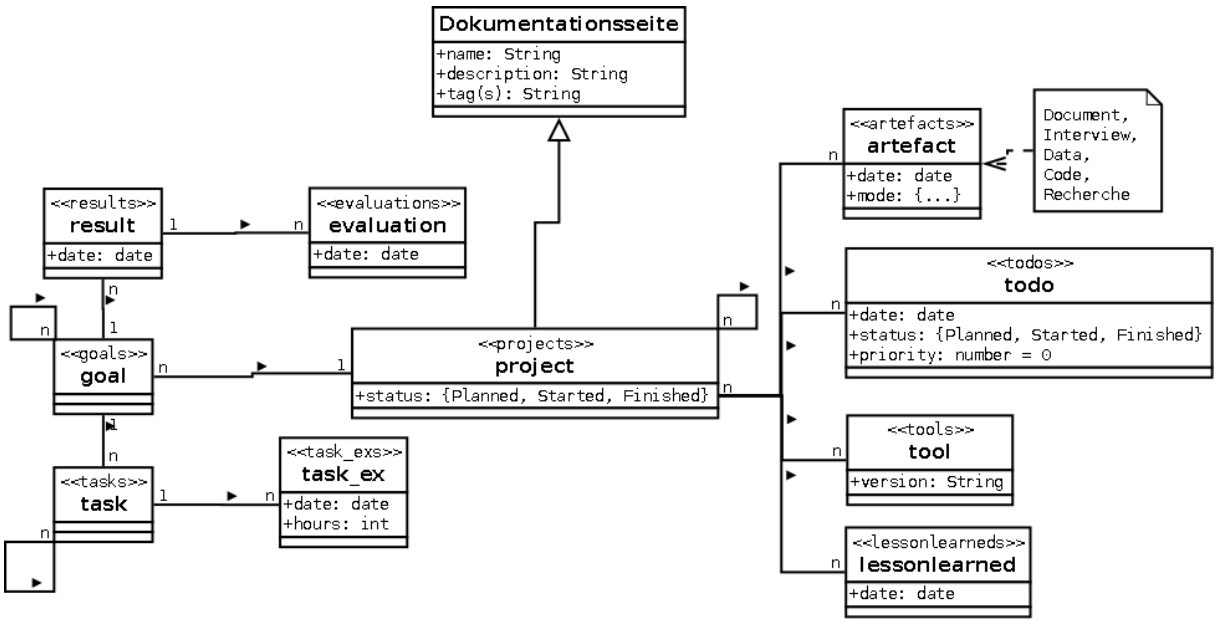


Abb. 3.6: Struktur der Dokumentation

Von jedem Objekt können Instanzen – Inhaltsseiten in der Dokumentation – erstellt werden. Diese besitzen neben einem eindeutigen Namen und einem beliebig langen Beschreibungstext ein Attribut „Tag“, in das Schlagworte eingegeben werden können, nach denen in der Dokumentation explizit gesucht werden kann. Im Beschreibungstext können über den eindeutigen Namen Verlinkungen auf andere Dokumentationsseiten eingefügt werden. Auch „Artefakte“, „Todos“, „Werkzeuge“ und „Lessons-Learned“ können jeder Instanz zugeordnet werden. In der Abbildung 3.6 werden diese Vererbungs- und Beziehungsstrukturen aus Gründen der Übersichtlichkeit für das Objekt „Projekt“ dargestellt. Im Folgenden einige weitere Informationen zu den Objekten:

Projekt Projekte besitzen einen Status. Außerdem können ihnen eine beliebige Anzahl an Zielen und Unterprojekten zugeordnet werden.

Ziel Zielen können Unterziele, Aufgaben und Ergebnisse zugeordnet werden.

Aufgabe Aufgaben können Teilaufgaben und Durchführungen zugeordnet werden.

Durchführung Durchführungen besitzen ein Datum sowie die Anzahl an Stunden, die für sie in Anspruch genommen worden sind.

Resultat Ergebnisse besitzen ein Datum, an dem sie erzielt worden sind. Außerdem werden ihnen beliebig viele Evaluationen zugeordnet.

Evaluation Evaluationen besitzen ein Datum, an dem sie durchgeführt worden sind.

Artefakt Ein Artefakt kann ein Dokument, ein Gespräch, Daten, Code oder eine Recherche sein. Außerdem besitzen sie ein Datum, an dem sie dem Team als Informationsquelle zur Verfügung gestellt wurden.

Todo Todos beschreiben Vorschläge für Aufgaben. Sie besitzen das Datum, an dem sie erstellt worden sind, und einen Status. Die Höhe einer Zahl gibt die Priorität des Vorschlags an.

Tool Werkzeuge besitzen eine Version, mittels der nach einer aktuellen Version eines Werkzeugs gesucht werden kann.

Lessons-Learned Lessons-Learned besitzen das Datum, an dem sie beschrieben worden sind.

Becker [8] hat für die Dokumentation ein eigenes System entwickelt, das nicht frei verfügbar ist; ich habe mich für semantische Wikis entschieden. In einem Wiki können beliebig viele Seiten erstellt und mittels einfacher Syntax mit Freitextinhalt gefüllt werden. Auch Referenzen sind durch Hyperlinks möglich. Wikis werden über Webbrowser bedient und unterstützen den Mehrbenutzerbetrieb. Semantische Wikis erweitern diese Wikifunktionalität um die Möglichkeit, Objekte mit Attributen zu definieren. Abbildung 3.7 zeigt einen Ausschnitt der Startseite eines solchen Wikis.

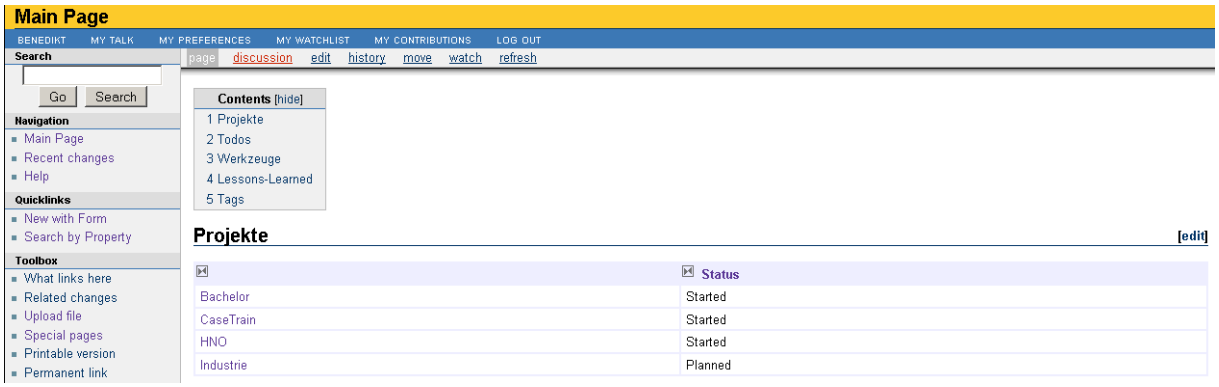


Abb. 3.7: Startseite KDDM-Wiki

Um dieses KDDM-Wiki mit Inhalten zu füllen, soll es während der Durchführung eines Data-

Mining-Projekts zu dessen Dokumentation verwendet werden. Informationen für den Business-Case, jegliche Ergebnisse aus der Entwicklung in Data-Assay, Data-Warehouse, Reporting und Data-Mining, sowie die Resultate für die Business-Story soll das Team darin eingeben. Die Dokumentation wird in Abschnitt 2.4.2.2 des Prozessmodells als wichtig erklärt.

Neue Einträge können über Formulare vereinfacht eingegeben werden. Mittels der semantischen Informationen automatisch erstellte Übersichtstabellen über Instanzen von Objekten, z.B. Projektlisten, Aufgabenlisten, siehe auch in Abbildung 3.7, verbessern die Übersicht.

Mit Inhalten gefüllt, kann das Wiki als Informationsquelle für neue Projekte verwendet werden. Das Suchen nach Informationen erfolgt entweder über eine Navigation durch die Referenzen, z.B. in den Übersichtstabellen. Oder über die Suche. Dabei kann entweder rein textuell oder nach speziellen Attributwerten, semantischen Annotationen wie den Tags, gesucht werden. So lässt sich über das Wiki ein Wissensmanagement innerhalb einer Organisation betreiben.

3.7.2 Dokumentenversionierungssystem

Das Wiki eignet sich nur begrenzt zum Dokumentenmanagement. So können zwar theoretisch beliebige Dateien im Wiki gespeichert werden; die Handhabung ist jedoch umständlich, außerdem können die Inhalte dieser Dateien nicht durchsucht werden. Stattdessen wird für das Dokumentenmanagement ein separates Versionierungssystem eingesetzt und die jeweiligen Daten im KDDM-Wiki lediglich referenziert, entweder über eine direkte Verlinkung zur Datei im Versionierungssystem oder eine Angabe des Dateinamen und -pfads. Diese Art der Speicherung ermöglicht auch, spezielle Benutzerrechte zu vergeben. So müssen im KDDM-Wiki keine sicherheitsbedenklichen Daten wie Passwörter gespeichert, sondern können ausgelagert und nur befugten Personen zugänglich gemacht werden.

3.8 Software-Komponenten

Die Bewertung der Werkzeuge muss laut Abschnitt 2.4.3.1 des Prozessmodells geschehen. Das Prozessmodell hat die Aufgaben – das Was – beschrieben, die bisherigen Abschnitte in diesem Kapitel haben die Techniken – das Wie – vorgestellt. In diesem Abschnitt möchte ich zeigen, mit welchen Werkzeugen – das Womit – diese Techniken durchgeführt werden können.

3.8.1 Definition von Komponenten

In den vorherigen Abschnitten wurde ein Grundstock an Techniken genannt, die in einem Data-Mining-Projekt benötigt werden. Diese Techniken lassen sich in fünf Komponenten gruppieren: ETL, Data-Warehouse, Reporting, Data-Mining und Dokumentation. Diese Komponenten

möchte ich zunächst definieren und anschließend verwenden, um verschiedene Werkzeuge zu kategorisieren. Über die Verwendung von Komponenten soll die Methodologie möglichst unabhängig von konkreten Werkzeugen sein. So können auch neue Werkzeuge kategorisiert und im Rahmen der Methodologie eingesetzt werden.

3.8.1.1 ETL-Komponente

Für die Extraktion, die Transformierung, und das Laden bzw. Einlesen von Daten sind ETL-Werkzeuge geeignet. Mit diesen Werkzeugen sollen Teile des Data-Assay und des Data-Warehouse umgesetzt werden.

Drei Techniken spielen hier eine Rolle: Das Einlesen der Datenquellen, ihre Umwandlung zu tabellarischen Daten und ihre Speicherung in einer relationalen Datenbank. Außerdem die Umsetzung des ER-Modells. Und schließlich die Umsetzung des MD-Modells.

Für die erste Aufgabe sind folgende Funktionen nötig:

- ETL-Werkzeuge sollten Daten aus verschiedenen Datenquellen einlesen können, z.B. aus Dateien mit Standard-Formaten (CSV, TXT, ARFF), aus proprietären Dateiformaten (z.B. Microsoft Excel, XBase), oder Datenbanken (z.B. MySQL, PostgreSQL, Microsoft SQL Server).
- Auch die Ausgabe der Daten muss in diese Formate funktionieren, vor allem in relationale Datenbanken.
- Anpassung von beliebigen Daten in tabellarische Form.
- Normierung von Daten, darunter das Auftrennen von Zeichenketten.
- Angleichen von Attributwerten, z.B. Ersetzen von Ersatzwerten durch leere Werte in Numerischen Attributen, das Angleichen von Datumsformaten, das Umwandeln von Numerischen Attributen in Kategorische Attribute (bzw. umgekehrt das Umwandeln von Kategorischen Attributen in Numerische Attribute).

Um das ER-Modell umzusetzen sind insbesondere folgende Funktionen nötig:

- Kombination verschiedener Datenquellen.
- Beliebiges Anpassen von Instanzen zur Erstellung von Objekten.
- Beliebiges Anpassen von Attributwerten zur Erstellung von Attributen.

Um das MD-Modell umzusetzen sind insbesondere folgende Funktionen nötig:

- Das Erstellen von Dimensionstabellen.

Gute ETL-Werkzeuge zeichnen sich durch die Möglichkeit der *Visuellen Prozessmodellierung* aus. Dabei erstellt der Benutzer einen Graphen aus Knoten und Pfeilen. Die Pfeile dienen dem Transport von Daten zwischen Knoten; diese führen je nach Knotentyp Aktionen (z.B. Filterung) auf die Daten aus. Man kann Datenflüsse intuitiv modellieren (vgl. [50, S. 199]). Nach einer gewissen Eingewöhnungszeit ist dieser Ansatz einfach, aber dennoch sehr mächtig (vgl. [50, S. 393]).

Was außerdem sehr nützlich ist: Die ETL-Werkzeuge speichern diese Datenflüsse zur jederzeitigen wiederholten Ausführung. Das wiederholte Einlesen aktueller Daten, das Ausmerzen eines Fehlers oder das Hinzufügen eines neuen abgeleiteten Attributs ist direkt möglich. Wenn in den Einzelfallstudien von „mehreren Schritten“ die Rede ist, wird es sich häufig nicht um manuelle Ausführungen, sondern verkettete Knoten des *ETL*-Werkzeugs handeln, die „auf Knopfdruck“ ausgeführt werden können. Daneben ermöglichen ETL-Werkzeuge häufig die Kommentierung und das Debugging, ähnlich wie in Programmiersprachen.

3.8.1.2 Data-Warehouse-Komponente

Data-Warehouse-Werkzeuge ermöglichen das performante Speichern und definierte Abfragen von Daten. Folgende Techniken sollen durch die Data-Warehouse-Komponente abgedeckt werden:

- Die Werkzeuge sollen die Erstellung und Verwaltung von relationalen Datenbanken ermöglichen. Auf diese habe ich mich im Data-Assay festgelegt.
- Sie sollten die Abfragesprache SQL implementieren und erlauben, darüber Anfragen an die Datenbanken zu stellen, da diese Funktion im Data-Assay zur Beschreibung der Daten und im Reporting zur Erstellung von Berichten genutzt wird.
- Sie sollten die benutzerfreundliche Verwaltung von Primärschlüsseln, Indices der Fremdschlüssel und Views ermöglichen, die für das ER-Modell und im Reporting eine Rolle spielen.
- Sie sollten die Erstellung und Verwaltung von Multidimensionalen Datenbanken unterstützen, die ein Multidimensionales Modell implementieren können. Ein solches wird bei der Erstellung des Data-Warehouse konzeptioniert. Sie sollten weiterhin eine Abfrage der Multidimensionalen Datenbanken, z.B. über MDX, ermöglichen.

3.8.1.3 Reporting-Komponente

Werkzeuge zum Reporting enthalten folgende Funktionen:

- Sie ermöglichen die Erstellung eines individuellen Berichts, z.B. mit Zusatzbeschreibungen.

- Sie ermöglichen das Füllen eines Berichts mittels Abfragen auf das ER-Modell, z.B. SQL.
- Sie ermöglichen das Füllen eines Berichts mittels Abfragen auf das MD-Modell, z.B. MDX.
- Sie ermöglichen das jederzeitige Ausführen eines Berichts, z.B. über eine webbasierte Benutzerschnittstelle.
- Sie ermöglichen das Speichern in verschiedene Formate, z.B. CSV, XLS, PDF.
- Sie sollten eine Entscheidungsträger-freundliche Erstellung von Abfragen und Sicht auf die Daten ermöglichen – durch intuitive Erstellung von SQL-Abfragen und interaktives OLAP.

3.8.1.4 Data-Mining-Komponente

Werkzeuge zum Data-Mining ermöglichen das Durchführen von Techniken, die im Data-Assay und Data-Mining nötig sind:

- Sie ermöglichen das Einlesen von Tabellarischen Daten aus CSV- und ARFF-Dateien.
- Sie implementieren Data-Mining-Techniken aus Abschnitt 3.6.

Ich habe verschiedene Informationsquellen berücksichtigt, um mögliche Werkzeuge für die Komponenten zu identifizieren. Im Folgenden beschreiben ich die drei wichtigsten Quellen:

- Gregory Piatetsky-Shapiro, einer der Autoren der KDDM-Definition [27], die ich in Kapitel 2 zitiert habe, veröffentlicht seit 1997 in regelmäßigen Abständen ein vielbeachtetes Informationsblatt per E-Mail. Über seine Webseite werden auch Umfragen durchgeführt. Eine Umfrage mit der Frage „What data mining tools have you used for a real project (not just for evaluation) in the past 6 months?“ [52] wurde als eine Informationsquelle für Werkzeuge, insbesondere für die Data-Mining-Komponente verwendet.
- Der Zusammenfassungsbericht der „2nd Annual Data Miner Survey“ [60] von Rexer Analytics nennt weitere mögliche Data-Mining-Werkzeuge.
- Die Studie zur „Open Source Business Intelligence“ [29] nennt freie Werkzeuge aus dem Bereich der Business-Intelligence und damit mögliche Werkzeuge für die ETL- Data-Warehouse- und Reporting-Komponenten.

Daneben wurden Werkzeuge in die Auswahl einbezogen, die von Herstellern als geeignete Ergänzungen zu ihren Werkzeugen empfohlen wurden, z.B. MySQL, PostgreSQL. Ich habe Werkzeuge ausgewählt, die entweder in mehreren Informationsquellen enthalten waren oder besonders häufig empfohlen wurden. Diese Werkzeuge wurden anschließend daraufhin untersucht, ob sie die Anforderungen von Komponenten erfüllen; dazu wurden größtenteils Webseiten der Hersteller konsultiert.

3.8.1.5 Dokumentationskomponente

Werkzeuge zur Dokumentation, wie sie in dieser Methodologie empfohlen werden, besitzen eine Wiki-Funktionalität sowie die Möglichkeit, beliebige Dokumente zu archivieren und versionieren. Beide Dienste sollen dabei über eine Client-Server-Architektur, z.B. per Webbrowser, angeboten werden.

Für jede der Komponenten konnten mehrere Werkzeuge gefunden werden, die diese erfüllen. Eine Tabelle 3.4 gibt einen Überblick über diese Werkzeuge und ihre Kategorisierung in ETL, Data-Warehouse (DW), Reporting (RP) und Data-Mining (DM). Wenn ein Werkzeug für eine Komponente geeignet ist, wird das durch „+“ gekennzeichnet, wenn sie nicht geeignet ist durch „-“ und wenn sie beschränkt geeignet ist durch „(+)“. Für jedes Werkzeug werden Werkzeuge eingeschlossen, die der verbesserten Nutzung dienen, z.B. HeidiSQL oder Pentaho Schema Workbench. Auch nicht eingefügt habe ich allgemeine Hilfswerkzeuge, von denen es sehr viele am Markt gibt, z.B. der Open-Source-Texteditor *Notepad++*[5]. Am Markt zeigt sich die Aufteilung in *Open-Source*- und *Closed-Source*-Werkzeuge, die sich in ihren Konzepten stark unterscheiden. Es zeigt sich, dass die Open-Source-Werkzeuge einzelne oder nur Teile der Komponenten erfüllen, wohingegen die Closed-Source-Systeme den Anspruch besitzen, vollintegriertes Data-Mining in einem Werkzeug anzubieten.

Im Folgenden stelle ich diese Konzepte kurz vor und beschreibe die gefundenen Werkzeuge.

3.8.2 Open-Source-Werkzeuge

Laut der Open Source Initiative[6], einer gemeinnützigen Organisation, zeichnet es Open-Source-Werkzeuge aus, dass sie frei verfügbar erhalten (z.B. über das Internet), modifiziert, genutzt und weitergegeben werden können. Auch bei einer kommerziellen Nutzung dieser Werkzeuge fallen keine direkten Kosten an. Um strenggenommen als Open-Source zu gelten, muss das Werkzeug eine entsprechende Lizenz besitzen, aktuelle Beispiele für solche Lizenzen sind „AGPLv3“[7], „GPLv3“[8] und „LGPLv3“[9]. Für die vollständige freie Verwendung innerhalb einer Organisation, wie sie in dieser Arbeit behandelt wird, können jedoch auch andere Lizenzen als Open-Source gesehen und dementsprechend genutzt werden. Eine wichtige Voraussetzung für ein langfristig konkurrenzfähiges Open-Source-Werkzeug ist die *Community*. Darunter werden die

[5] Notepad++, http://notepad-plus.sourceforge.net/de/site.htm, Dezember 2009
[6] The Open Source Definition, http://www.opensource.org/docs/osd, Dezember 2009
[7] GNU AFFERO GENERAL PUBLIC LICENSE v3, http://www.opensource.org/licenses/agpl-v3.html, Dezember 2009
[8] GNU General Public License version 3 (GPLv3), http://www.opensource.org/licenses/gpl-3.0.html, Dezember 2009
[9] The GNU Lesser General Public License (LGPLv3), http://www.opensource.org/licenses/lgpl-3.0.html, Dezember 2009

Werkzeug	**ETL**	**DW**	**RP**	**DM**	**Doku**
Open-Source					
Kettle	+	-	-	-	-
Talend Open Studio	+	-	-	-	-
MySQL	-	+	-	-	-
PostgreSQL	-	+	-	-	-
Mondrian OLAP Server	-	+	-	-	-
Palo OLAP Server, Excel	-	+	+	-	-
Pentaho Reporting	-	-	+	-	-
Jasper BI-Suite	-	-	+	-	-
VIKAMINE	-	-	-	+	-
Weka	-	-	-	+	-
Rattle	-	-	-	+	-
RapidMiner	(+)	-	-	+	-
Semantic MediaWiki	-	-	-	-	+
Subversion	-	-	-	-	+
Closed-Source					
Excel 2007	(+)	-	+	(+)	-
SPSS Clementine	+	-	+	+	-
SAS-Enterprise Miner	+	-	+	+	-
STATISTICA Data Miner	+	-	+	+	-
Microsoft SQL Server	+	+	+	+	-

Tab. 3.4: Komponenten Werkzeuge Übersicht

Personen verstanden, die aktiv an der Entwicklung des Werkzeugs teilhaben, z.B. über Beiträge in Foren oder Entwicklungen am Quellcode. Im KDDM-Umfeld zeigt sich der Trend, dass Open-Source-Werkzeuge von etablierten Firmen gefördert werden – mit der Erwartung, Umsatz über individuelle Beratung von Nutzern der Werkzeuge zu erzielen (vgl. [9]). Im Folgenden werden für jede Komponente mindestens zwei Werkzeuge vorgestellt, die für die Verwendung innerhalb einer Organisation als Open-Source angesehen werden können.

3.8.2.1 Pentaho Data Integration

Dieses, ehemals „Kettle“ genannte, Werkzeug in Version „v3.2.0 stable„ und mit LGPL-Lizenz implementiert die Funktionen der ETL-Komponenten mittels Visueller Prozessmodellierung und folgender Konzepte:

Step ist die Bezeichnung eines Knoten.

Hop ist die Bezeichnung eines Pfeils.

Transformation wird die Zusammenfassung von Knoten und Pfeils in einer ausführbaren Datei genannt.

Job wird die Zusammenfassung von Transformationen in einer ausführbaren Datei genannt.

Pentaho Data Integration ist grundsätzlich intuitiv zu bedienen, die Erklärung mancher Steps ist dennoch hilfreich[10]. Weitere Informationen zu Pentaho Data Integration finden sich auf der Webseite von Pentaho[11].

3.8.2.2 Talend Open Studio

Auch dieses Werkzeug (mit GPL-Lizenz) implementiert die ETL-Komponente mittels Visueller Prozessmodellierung. Talend Open Studio in Version „v3.1.3„ kann als funktional gleichwertig zu Pentaho Data Integration eingeschätzt werden. Ihr Anbieter, Talend, hat 2009 eine Kapitalerhöhung von 12 Mio. Dollar erhalten[12]. Nähere Informationen finden sich im Benutzerhandbuch von Talend[13].

3.8.2.3 MySQL Server

MySQL Server in Version „v5.0.51“ und mit GPL-Lizenz ist ein relationales Datenbanksystem und implementiert damit einen Teil der Techniken für das Data-Warehouse. Ihre Abfragesprache setzt den SQL-Standard um. Die Verwaltung von Tabellen, Views, Primärschlüsseln und Indices funktioniert z.B. über die Open-Source-Werkzeuge *HeidiSQL*[14] in Version „v4.0“ und mit GPL-Lizenz und *MySQL Workbench*[15] in Version „v5.1.16“ und mit GPL-Lizenz. Weitere Informationen zu MySQL finden sich auf der Webseite[16].

3.8.2.4 PostgreSQL

Auch PostgreSQL in Version „8.4.1“ und mit GPL-Lizenz ist ein relationales Datenbanksystem, das die Standardabfragesprache SQL umsetzt und damit die Anforderungen an diesen Teil der

[10] Pentaho Data Integration v3.2. Steps, http://wiki.pentaho.com/display/EAI/Pentaho+Data+Integration+v3.2.+Steps, Dezember 2009
[11] siehe http://kettle.pentaho.org/
[12] Open Source Anbieter gewinnt 12 Mio. Dollar Venture Capital, http://mittelfranken.business-on.de/open-source-talend-nuernberg-datenintegration-bertrand-diard-_id969.html, Dezember 2009
[13] Talend User Guide, http://www.talend.com/resources/documentation.php, Dezember 2009
[14] HeidiSQL, http://www.heidisql.com/, Dezember 2009
[15] MySQL Workbench, http://www.mysql.de/products/workbench/, Dezember 2009
[16] MySQL, http://www.mysql.de/, Dezember 2009

Data-Warehouse-Komponente erfüllt. Nähere Informationen sind auf der Webseite zu PostgreSQL verfügbar[17]

3.8.2.5 Mondrian OLAP Server

Der Mondrian OLAP Server in Version „3.1.2.“ mit CPL-Lizenz in Kombination mit einer relationalen Datenbank, z.B. MySQL oder PostgreSQL, ermöglicht die Umsetzung eines Multidimensionalen Modells in einer multidimensionalen Datenbank.

Dazu werden mit dem Hilfswerkzeug *Mondrian Schema Workbench*[18] in der Version „3.1.1-stable“ und mit CPL-Lizenz die Data-Cubes mit ihren Dimensionen, Hierarchien, Levels, Fakten und Kennzahlen aus dem MD-Modell beschrieben, mit Dimensions- und Faktentabellen einer relationalen Datenbank verknüpft und in einer XML-Datei gespeichert. Dieses sog. „Schema“ verwendet der Mondrian OLAP Server, um MDX-Abfragen zu interpretieren und in SQL-Abfragen auf die Datenbank zu übersetzen. Gemeinsame Dimensionen mehrerer Data-Cubes, können in Mondrian mit *Virtuellen Data-Cubes* umgesetzt werden. Nähere Informationen zu Mondrian und Workbench siehe Dokumentation auf der Community-Webseite[19].

3.8.2.6 Palo OLAP Server

Der Palo OLAP Server von Jedox in Version „3.0“ und „free of licensing“[20] speichert direkt ein Multidimensionales Modell – anders als Mondrian, das multidimensionale Daten über den Umweg einer relationalen Datenbank anbietet. Die Verarbeitung der Daten findet hier im Speicher statt und kann besonders schnell geschehen (vgl. [33, S. 30]). Der Aufbau der Data-Cubes und das Einlesen sowie die Abfrage der Daten erfolgt über ein Plugin für Microsoft Excel und ist somit nicht vollständig kostenfrei umsetzbar. Es kann davon ausgegangen werden, dass der Palo OLAP Server vergleichbare Funktionalität, bis hin zum benutzerfreundlichen Zugriff auf die Daten über eine webbasierte Benutzerschnittstelle (vgl. [33, S. 29]), bietet.

3.8.2.7 Pentaho Reporting

Pentaho Reporting umfasst unter anderem die Werkzeuge *Business Intelligence Server*, *Report Designer* und *Design Studio*. In der Version „3.5.0-stable„ und mit CPL-Lizenz ermöglichen sie es, die Techniken aus dem Reporting umzusetzen.

[17] PostgreSQL, http://www.postgresql.org/, Dezember 2009
[18] Mondrian Schema Workbench, http://mondrian.pentaho.org/documentation/workbench.php, Dezember 2009
[19] Mondrian Overview, http://mondrian.pentaho.org/documentation/doc.php, Dezember 2009
[20] Palo, http://www.jedox.com/de/home/uebersicht.html, Dezember 2009

BI-Server stellt eine webbasierte Benutzerschnittstelle mit Authentifizierung zur Verfügung. Dort können Berichte „veröffentlicht“ und jederzeit ausgeführt und z.B. als CSV, PDF oder XLS-Datei gespeichert werden. Desweiteren bietet er über MDX und OLAP einen Zugriff auf das MD-Modell im Mondrian OLAP Server sowie eine Endnutzer-freundliche Möglichkeit zur Erstellung von SQL-Abfragen des ER-Modells. Report Designer ermöglicht das Erstellen von individuellen Berichten aus beliebigen SQL- oder MDX-Abfragen, die anschließend auf dem BI-Server veröffentlicht werden können. Design Studio ermöglicht das automatisierte Verarbeiten von Berichten, z.B. zur Ausführung mehrerer Berichte oder zum Versenden per E-Mail. Nähere Informationen zu jedem dieser Werkzeuge werden auf der Community-Webseite von Pentaho angeboten[21], von der aus auch auf zum Forum und Wiki als erste Anlaufstelle bei Problemen gelangt werden kann.

3.8.2.8 Jasper Business Intelligence Suite

Auch die BI-Suite von JasperSoft erfüllt mit seinen Werkzeugen *JasperServer* und *iReport*, lizenziert über GPL, die Kriterien einer Reporting-Komponente. Eine webbasierte Benutzerschnittstelle; der Zugriff auf MD-Modelle wie Mondrian OLAP Server über MDX; der Zugriff auf ER-Modelle in relationalen Datenbanken über SQL; die individuelle und Endnutzer-freundliche Erstellung von Berichten; und das Versenden von Berichten per E-Mail, sind mit JasperSoft genauso möglich wie mit Pentaho Reporting – wenn sich die Werkzeuge in der Bedienung auch stark unterscheiden. Nähere Informationen finden sich auf der Community-Webseite von JasperSoft[22].

3.8.2.9 VIKAMINE

Das Data-Mining-Werkzeug VIKAMINE[23] besitzt eine LGPL-Lizenz und unterstützt das Einlesen von CSV- und ARFF-Dateien. Es bietet einen eigenen Ansatz zur Beschreibung von Attributen und Objekten in einer interaktiven „Zoomtable“[24]. VIKAMINE ist auf die Subgruppenentdeckung spezialisiert und implementiert deren Technik unter Anwendung unterschiedlicher Algorithmen.

[21] Pentaho Community, http://community.pentaho.com/index.php, Dezember 2009
[22] JasperForge, http://jasperforge.org/, Dezember 2009
[23] VIKAMINE, http://www.vikamine.org/, Dezember 2009
[24] VIKAMINE – An Overview, http://vikamine.sourceforge.net/data/VikamineTutorial.pdf, Dezember 2009

3.8.2.10 Weka

Mit Weka[25] wurde auch das ARFF-Format eingeführt. Außerdem unterstützt es das Einlesen von CSV-Dateien. Dieses Werkzeug in Version „3.7.0“ und GPL-Lizenz bietet verschiedene Diagramme zur Beschreibung der Daten, implementiert außerdem viele verschiedene Data-Mining-Techniken, unter anderem das Assoziationsregellernen, das Entscheidungsbaumlernen und die Segmentierung.

3.8.2.11 Rattle

Rattle [68] bietet einen benutzerfreundlichen, grafischen Zugang zur statistischen Sprache R[26]. Neben dem detaillierten Beschreiben von Attributen aus CSV- und ARFF-Dateien, z.B. in Box-Plots oder Histogrammen, implementiert Rattle in Version „2.5.0“ und mit GPL-Lizenz verschiedene Data-Mining-Techniken, z.B. die Korrelationstabelle.

3.8.2.12 RapidMiner

RapidMiner in Version „4.4“ und mit AGPL-Lizenz ermöglicht das Einlesen von CSV- und ARFF-Dateien und deren umfangreiche Beschreibung in verschiedenen Diagrammen, z.B. Säulendiagrammen, Balkendiagrammen, Kurvendiagrammen oder Kreisdiagrammen. RapidMiner besitzt mit „Operatorketten“[27] einen eigenen Ansatz zur grafischen Darstellung von Datenflüssen, der für die ETL-Komponente eingesetzt werden kann, jedoch nicht die Intuitivität der Visuellen Prozessmodellierung besitzt.

3.8.2.13 Semantic MediaWiki

MediaWiki[28] und seine Erweiterung Semantic MediaWiki[29] sind „free software“ und erfüllen die Anforderungen an das KDDM-Wiki.

3.8.2.14 Subversion

Das Open-Source-Versionierungssystem Subversion[30] ermöglicht das Dokumentenmanagement, wie es für die Dokumentationskomponente verlangt wird.

[25] Weka, http://www.cs.waikato.ac.nz/ml/weka/, Dezember 2009
[26] The R Project for Statistical Computing, http://www.r-project.org/, Dezember 2009
[27] Multi-Layered Data View Concept, http://rapid-i.com/content/view/15/69/lang,de/, Dezember 2009
[28] MediaWiki, http://www.mediawiki.org/wiki/MediaWiki/de, Dezember 2009
[29] Semantic MediaWiki, http://semantic-mediawiki.org/wiki/Semantic_MediaWiki, Dezember 2009
[30] Subversion, http://subversion.tigris.org/, Dezember 2009

3.8.3 Closed-Source-Werkzeuge

Anbieter von Closed-Source-Werkzeugen werden dadurch gekennzeichnet, dass sie den Quellcode ihrer Software nicht offen legen. Eine vollständige Nutzung wird nur für einen bestimmten Zweck, für einen bestimmten Zeitraum und nach Bezahlen einer Lizenzgebühr möglich, über die sich die Hersteller finanzieren. Diese Lizenzgebühr kann von ca. 100 Euro für die Microsoft Office Suite, für einen Computer, bis zu mehrere Tausend Euro für den Microsoft SQL Server[31] betragen. Unabhängig davon, wie hoch der Preis für solche Closed-Source-Data-Mining-Produkte tatsächlich ist, er ist niemals symbolisch, erfordert stets eine gewisse Investition.

Im Folgenden gebe ich einen Überblick über häufig verwendete Systeme und nenne Komponenten, die sie erfüllen.

3.8.3.1 Microsoft Excel 2007

Dieses Werkzeug kann zum Data-Mining verwendet werden, wenn es die Anforderungen zulassen. Innerhalb einer Machbarkeitsstudie zu einem der Projekte haben wir Excel beispielsweise folgendermaßen eingesetzt:

Wir haben die Datenquelle, in Form mehrerer CSV-Dateien, mit *Pentaho Data Integration* eingelesen und jeweils in eine Tabelle von *MySQL* gespeichert. Wir haben mit SQL einen View erstellt, der in einem Verbund mehrere Datenquellen verbindet und in einer einzelnen Tabelle anzeigt. Diese Tabelle haben wir als CSV-Datei exportiert und mit Excel eingelesen. Durch manuelles Kopieren und Einfügen, automatisches Suchen und Ersetzen sowie unter Verwendung von einfachen Tabellenkalkulationen haben wir die Rohdaten vorverarbeitet.

Anschließend haben wir die Daten mittels der Pivot-Funktion (vgl. [25]) von Excel angezeigt. In Pivot-Tabellen konnten wir Zeilen und Spalten beliebig umpositionieren. Filter oder Sortierungen haben die Sicht auf Instanzen mit bestimmten Attributwerten gelenkt. Über Mengen von Instanzen mit identischen Attributwerten konnten wir Summen oder ähnliche Aggregationsberechnungen anstellen. Aus diesen Sichten haben wir einzelne Berichte erstellt, sie nach den Vorgaben des Entscheidungsträgers aufbereiten und als CSV, PDF oder XLS exportiert. Desweiteren haben wir direkt in Excel verschiedene Diagramme erstellt und zum Entdecken von Mustern genutzt.

Die Vorteile dieses Vorgehens liegen in der intuitiven Bedienung von Microsoft Excel. Einlesen und Vorverarbeiten sind unkompliziert möglich. Eine Pivot-Tabelle bietet als eine vereinfachte Form des OLAP die interaktive Beschreibung der Rohdaten und kann durch Diagramme visualisiert werden. Selbst „ungeübten Anwendern“ [25] erlaubt Excel somit eine einfache Form des Data-Mining.

[31] SQL Server 2008 Pricing, http://www.microsoft.com/sqlserver/2008/en/us/pricing.aspx, Dezember 2009

Die Nachteile des Data-Mining mit Excel liegen im hohen Aufwand bei erweiterten Anforderungen. Die Vorverarbeitungsschritte, der Aufbau der Pivottabellen sowie die Erstellung der Diagramme müssten bei neuen Daten stets vollständig wiederholt werden. In der Theorie sollte dieser Aufwand geringer sein, solange die Struktur der Datenquelle bestehen bleibt, in der Praxis konnte sich das nicht bestätigen; neue oder leicht abgeänderte Daten erforderten stets den gleichen Aufwand. Und auch wenn sich manche dieser Aufgaben über Zusatzfunktionen von Excel, z.B. Macros, automatisieren lassen, spätestens bei geringen Änderungen der Struktur ist zu erwarten, dass sich keine Vorarbeiten wiederverwenden lassen. Für jegliche automatisierte Ansätze ist Excel damit nicht geeignet.

Zusammenfassend kann man die Verwendung von Excel zum Data-Mining als *Quick & Dirty*-Lösung bezeichnen. Einfache Probleme lassen sich schnell behandeln, bei aufwändigeren Anforderungen sollte sich der Einarbeitungsaufwand anderer Werkzeuge relativ schnell amortisieren. Nur als Ergänzung zu anderen Werkzeugen (z.B. Palo, MySQL) empfiehlt sich Excel auf für umfangreichere Projekte ([45]).

Es sei anzumerken, dass Excel zur Präsentation der Ergebnisse in der Business-Story genutzt werden kann.

3.8.3.2 SPSS Clementine

Mittlerweile wurde SPSS Inc. von IBM aufgekauft, das seit 1993 entwickelte SPSS Clementine in PASW Modeler[32] umbenannt. Es wird über die Visuelle Prozessmodellierung bedient ([50, S. 199]). Bereits in PASW Modeler integriert bzw. über Erweiterungsmodule (z.B. PASW Data Preparation) nachrüstbar sind Knoten, mit denen ETL-Aufgaben durchgeführt werden können. Dieses Werkzeug setzt voraus, dass die Daten abrufbereit zur Verfügung stehen und sieht keine Speicherung in einem Data-Warehouse vor. Diese kann über zusätzliche Werkzeuge realisiert werden. Man kann Daten auch betrachten und mittels Berichte exportieren. Eine Vielzahl von Data-Mining-Techniken stehen zur Verfügung (vgl. [50, S. 198]).

3.8.3.3 SAS-Enterprise Miner

Dieses von SAS Institute Inc. entwickelte Werkzeug kann ähnlich wie SPSS Clementine mittels Visueller Prozessmodellierung bedient werden. SAS-Enterprise Miner[33] unterstützt ETL-Techniken, Reporting-Funktionalität und verschiedene Data-Mining-Algorithmen. Eine Zwischenspeicherung in einem Data-Warehouse ist auch hier nicht vorgehesehen (vgl. [50, S. 203-213]).

[32] PASW Modeler, http://www.spss.com/software/modeling/modeler/, Dezember 2009
[33] SAS Analytics, http://www.sas.com/technologies/analytics/index.html, Dezember 2009

3.8.3.4 STATISTICA Data Miner

Dieses Werkzeug von StatSoft Inc. ist das dritte Werkzeug der „Three Most Common Data Mining Software Tools“ [50, S. 197] und bietet eine identische Grundfunktionalität. Es integriert ETL-Aufgaben, Reporting-Funktionalität und Data-Mining-Techniken, allerdings keine Zwischenspeicherung in einem Data-Warehouse (vgl. [50, S. 214-234]). In STATISTICA Data Miner[34] kann der Aufbau von SQL-Ausdrücken grafisch erfolgen (vgl. [50, S. 114]) .

3.8.3.5 Microsoft SQL Server

Es ist zu erwarten, dass Hersteller von Datenbanksystemen in naher Zukunft Data-Mining-Funktionalität in ihre Produkte integrieren werden. Denn es liegt nahe, die effiziente Speicherung und Abfrage von Daten mit Analysemöglichkeiten zu ergänzen. Der Microsoft SQL Server unterstützt neben der Erstellung von relationalen Datenbanken ETL-Funktionen zum Einlesen von Daten sowie die Erstellung von Berichten. An Data-Mining-Funktionalität werden unter anderem Assoziationsregellernen, Segmentierung und Entscheidungsbaumlernen unterstützt[35]. Auch das Erstellen eines Data-Cube und das Analysieren seiner Inhalte mittels Data-Mining-Techniken ist möglich. Der SQL Server[36] von Microsoft ist damit eine vollintegrierte Data-Mining-Suite.

3.9 Hardware-Komponenten

Für die Projekte wurden drei herkömmliche Arbeitsplatzrechner verwendet. Sie geben die Infrastruktur, siehe Abschnitt 2.4.1.2, während der Projekte vor und gehören als Teil der Ressourcen zur Planung, siehe Abschnitt 2.4.2.1:

Arbeitsplatz-Rechner Ein herkömmlicher Rechner (Pentium IV, 3 GHz, 2 GB RAM) mit Windows XP Professional, auf dem die ETL- und Data-Mining-Komponenten ausgeführt werden. Unter der Voraussetzung, dass Daten jeweils nur für eine Sitzung verwendet und anschließend gelöscht oder archiviert werden, ist dieser Rechner weniger sicherheitskritisch als die folgenden.

Data-Warehouse-Server Ein etwas leistungsstärkerer Rechner (Pentium Dual-Core, 3 GHz, 4 GB RAM) mit Ubuntu Linux Server Edition in Version „8.04 LTS“, auf dem das Data-Warehouse gespeichert und die Berichte ausgeführt werden. Dieser Rechner wird so konfiguriert, dass er nur innerhalb des Intranets der Organisation erreicht werden kann, um

[34] STATISTICA Data Mining and Predictive Analytics Software, http://www.statsoft.com/products/data-mining-solutions/, Dezember 2009
[35] Data Mining Algorithms, http://msdn.microsoft.com/en-us/library/ms175595.aspx, Dezember 2009
[36] SQL Server 2008, http://www.microsoft.com/sqlserver/2008/en/us/R2.aspx, Dezember 2009

Übergriffe von Vorneherein ausschließen zu können. Anders herum wird dieser Server so konfiguriert, dass E-Mails nicht versehentlich an unbestimmte Adressen außerhalb der Organisation verschickt werden. Dieser Rechner wird hauptsächlich „fern-gewartet“ und kann daher räumlich getrennt vom Arbeitsplatz-Rechner stehen, auch um einen unberechtigten Zugriff zu erschweren.

Dokumentationsserver Dieser Rechner (Pentium IV, 3 GHz, 1 GB RAM) mit Ubuntu Linux Server Edition in Version „8.04 LTS“ erfordert weniger Kapazitäten. Auf ihm werden das KDDM-Wiki und das Versionierungssystem intalliert. Auch dieser Rechner wird hauptsächlich fern-gewartet.

Abbildung 3.8 zeigt als Überblick über die Infrastruktur die Komponenten und ihre Schnittstellen in einem Diagramm (vgl. [46]). Jede Komponente wird dem Rechner zugeordnet, auf dem ihre Hauptfunktionalität liegt, z.B. werden Berichte zwar meist auf dem Arbeitsplatz-Rechner zusammengestellt, anschließend werden sie jedoch auf dem Data-Warehouse-Server „veröffentlicht“, um von dort jederzeit ausgeführt werden zu können. HTTP-Verbindungen per Webbrowser zur Bedienung des Data-Warehouse- und Dokumentationsservers werden aus Gründen der Übersichtlichkeit nicht dargestellt.

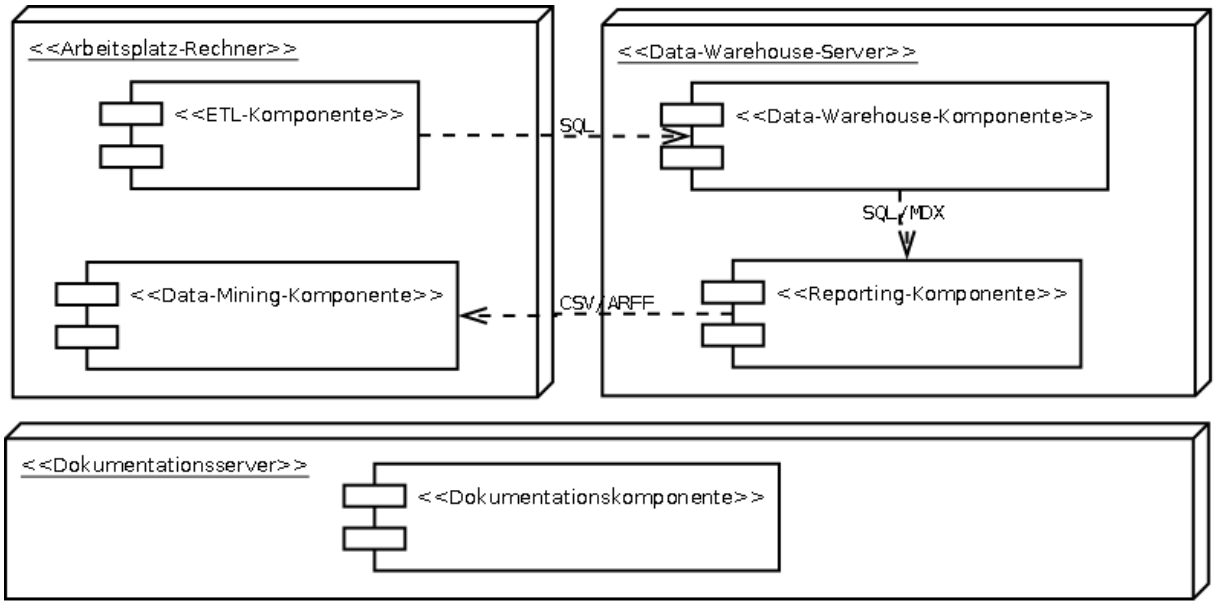


Abb. 3.8: Übersicht Infrastruktur

4 Design der Einzelfallstudien

Das folgende Kapitel beschreibt den Aufbau der beiden Einzelfallstudien, die in den darauffolgenden Kapiteln enthalten sind und die Anwendung der Methodologie aus dem vorherigen Kapitel demonstrieren sollen.

4.1 Forschungsfrage und Beobachtungspunkte

Folgende Forschungsfrage soll durch die Einzelfallstudien beantwortet werden: Wie kann Deskriptives Data-Mining mit der Entscheidungsträger-verständlichen Methodologie konkret durchgeführt werden?

Die Einzelfallstudien sind an potenzielle Entscheidungsträger und Entwicklerteams gerichtet und sollen beschreiben und erklären, wie Data-Mining mit der Methodologie aus dem vorherigen Kapitel in der Praxis betrieben werden kann. Der Leser soll einen Eindruck davon erhalten, wie auch weitere Probleme mit der Methdologie behandelt werden können. Fallstudien sind laut dem seit 2005 regelmäßig stattfindenden Workshop „Data Mining Case Studies"[1], laut Becker [8] und als zentraler Teil der Methode des Fallbasierten Schließens (vgl. [26]; [7]) zum Nachvollziehen von realen Data-Mining-Projekten gut geeignet.

In einer ersten Einzelfallstudie besteht der Fall aus dem Data-Mining-Projekt „Bachelor", in dem die Bachelorstudiengänge der Universität Würzburg mittels Prüfungsdaten bewertet werden. In der zweiten Einzelfallstudie wird der Fall „CaseTrain" behandelt, bei dem der Nutzen eines fallbasierten Lehrsystems an der Universität Würzburg beurteilt wird. Beide Fallstudien werden nicht anonymisiert. Die Ergebnisse der Projekte werden aus datenschutzrechtlichen Gründen nicht vollständig offen gelegt, sondern durch Beispiele veranschaulicht.

Diese Einzelfallstudien sind keine *Schritt-für-Schritt-Anleitung* zum vollständigen Nachvollziehen und wiederholtem Durchführen des jeweiligen Data-Mining-Vorhabens. Dies zeigt sich in zweierlei Hinsicht. Zum einen werden die Einzelfallstudien nicht chronologisch beschrieben, sondern nach Relevanz für Entscheidungsträger oder Team in zwei Teile geteilt:

[1] Workshop on Data Mining Case Studies and Practice Prize, http://www.dataminingcasestudies.com/, Dezember 2009

1. Zunächst werden Business-Case und Business-Story beschrieben. Sie sind hauptsächlich für einen Entscheidungsträger, aber auch für das Entwicklerteam interessant.
2. Anschließend werden die Teile der Umsetzung (Data-Assay, Data-Warehouse, Reporting und Data-Mining) beschrieben. Sie sind insbesondere für das Team relevant, können je nach technischem Hintergrund aber auch einen Entscheidungsträger interessieren.

Durch Referenzen vom Umsetzungsteil mit seinen Techniken und Werkzeugen zu den Anforderungen und Ergebnissen im Entscheidungsträger-verständlichen Teil der Einzelfallstudie sollen die kausalen Abhängigkeiten deutlich werden. Abbildung 4.1 zeigt das Design der Einzelfallstudien.

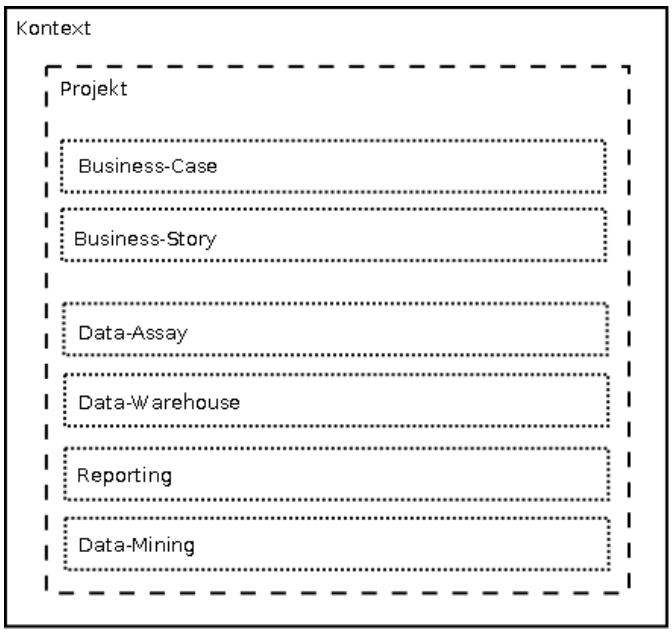


Abb. 4.1: Analyseeinheiten der Einzelfallstudien

Zum anderen wird meist auf die Beschreibung der konkreten Durchführung verzichtet und lediglich ein Hinweis gegeben, wenn eine Technik mit einem bestimmten Werkzeug durchgeführt werden konnte. Denn auf Grund der hohen Entwicklungsgeschwindigkeit der verwendeten Open-Source-Werkzeuge, kann sich die Bedienung der Funktionen jederzeit ändern. Außerdem sollen die verwendeten Werkzeuge lediglich Beispiele darstellen, wie die Komponenten der Methodologie ausgefüllt werden können.

4.2 Informationsquellen

Die Data-Mining-Projekte wurden in einem Team aus Data-Mining-, Daten- und Domänen-Experten durchgeführt, weshalb ich in den Einzelfallstudien „wir“ schreibe, wenn das Data-

Mining-Team gemeint ist. Wie in der Mehrfachfallstudie, wird auch in den Einzelfallstudien die Kommunikation im Data-Mining-Team nicht behandelt, weshalb eine Unterscheidung nicht notwendig ist.

Als Informationsquelle der Fälle dient das KDDM-Wiki mit der Dokumentation zum jeweiligen Data-Mining-Projekt. Außerdem hatte ich als jeweiliger Data-Mining-Experte im Team die Möglichkeit der „teilnehmenden Beobachtung“ (vgl. [71, S. 111]). Einerseits ermöglicht sie Einsichten, die ein Außenstehender eines Data-Mining-Projekts nicht haben kann, z.B. zum Aufwand oder zur Komplexität. Andererseits birgt die teilnehmende Beobachtung das Risiko des „Bias“ [71, S. 112]. Dieses Risiko besteht jedoch nur begrenzt, weil das Data-Mining im Team durchgeführt wurde, jedes Projekt durch die Vorgaben eines Entscheidungsträgers bestimmt war und insgesamt wenig Kontrolle über die Ereignisse eines Data-Mining-Projekts möglich ist, z.B. Programmierfehler in Open-Source-Werkzeugen oder Qualitätsprobleme mit den Daten (vgl. [71, S. 8]).

Die Einzelfallstudien setzen Kenntnisse zu Techniken und Komponenten der Methodologie voraus. Begriffe, die an anderer Stelle in der Arbeit eingeführt werden, werden in den Einzelfallstudien *hervorgehoben* und dem Index hinzugefügt. Die Inhalte der Einzelfallstudien wurden nicht wortwörtlich aus dem KDDM-Wiki oder erstellten Dokumenten entnommen. Zitate stammen dagegen wortwörtlich aus der Dokumentation, die im Anhang enthalten ist.

5 Ergebnisse der Einzelfallstudie: Bachelor

Dieses Kapitel beschreibt die Einzelfallstudie zum Projekt Bachelor. In diesem Projekt sollen die neu eingeführten Bachelorstudiengänge an der Universität Würzburg bewertet werden. Das Kapitel ist wie folgt strukturiert:

Ein Management-Summary fasst die wichtigsten Ergebnisse des Projekts zusammen. Ein Business-Case beschreibt den Hintergrund und die Ziele des Projekts. In einer Business-Story werden anschließend die Ergebnisse beschrieben. Die darauffolgenden Abschnitte beschreiben das Vorgehen im Projekt zum Data-Assay, Data-Warehouse, Reporting und Data-Mining.

5.1 Management-Summary

Ziel des Projekts ist die Bewertung der im Rahmen des Bologna-Prozesses neu eingeführten Bachelorstudiengänge an der Universität Würzburg mittels elektronisch erfasster Prüfungsdaten. Nicht zuletzt zahlreiche und wochenlang andauernde Bildungsstreiks der Studierenden haben die Relevanz dieses Themas bestätigt.

Im Folgenden einige wichtige Fragestellungen, die wir behandelt haben:

1. Wie entwickeln sich die wichtigsten Kennzahlen der Studiengänge über die Zeit?
2. Wie entwickeln sich die wichtigsten Kennzahlen der Prüfungsveranstaltungen?
3. Welche relevanten Leistungsdaten weisen die aktuell Studierenden auf?

Fragestellung 1 beantworten wir für jedes Studienfach durch eine Übersichtstabelle (für ein Beispiel, siehe Abbildung 5.1 in der Business-Story). Sie zeigt für jedes Studiensemester (z.B. Wintersemester 2007) und Fachsemester der Studierenden die Anzahl an eingeschriebenen Studenten, deren durchschnittlich erreichten ECTS-Punkte sowie deren Noten. Hier wird das Verhältnis zwischen Anfängern, Abbrechern und Absolventen deutlich, ein wichtiges Qualitätsindiz eines Studiengangs.

Fragestellung 2 wird von uns durch zwei Sichten auf die Prüfungsveranstaltungen eines Studienfachs behandelt. Erstere zeigt für jede Veranstaltung von wievielen und in welchen Fachsemestern der aktuell Studierenden sie bestanden worden ist (für ein Beispiel, siehe Abbildung 5.2). Die

zweite wirft einen Blick auf die aktuellen Fachsemester, in denen sich die Studierenden befinden. Sie gibt an, wieviel Prozent der Studierenden in einem Fachsemester die Veranstaltung bereits bestanden haben, ergänzt um deren Durchschnittsnote (für ein Beispiel, siehe 5.3). Hier wird gezeigt, wann Veranstaltungen normalerweise bestanden werden, möglicherweise deutlich später/früher als der Studienverlaufsplan vorsieht.

Fragestellung 3 haben wir durch eine komprimierte und eine detaillierte Studierendentabelle pro Studienfach beantwortet. In beiden werden Zahlen genannt, die die aktuelle Situation eines Studierenden unmissverständlich beschreiben, darunter das Fach- und Hochschulsemester, die durchschnittlichen ECTS-Punkte pro Fachsemester, die durchschnittliche Note, sowie die erreichten ECTS-Punkte und Noten in Prüfungsbereichen, z.B. in Pflicht- oder Wahlpflichbereichen sowie Schlüsselqualifikationen (für ein Beispiel, siehe 5.5). In der detaillierten Version werden zusätzlich die Noten der Studierenden in einzelnen Veranstaltungen des Pflichtbereichs aufgelistet – für eine noch bessere Aussage zur aktuellen Situation der Studierenden (für ein Beispiel, siehe 5.6).

Von jedem dieser insgesamt fünf Berichte pro Studiengang können jederzeit Auszüge als PDF- oder Excel-Dateien erstellt und in DIN-A-4-Größe ausgedruckt werden. Wenn aktuelle Daten verfügbar werden, können sie auch berücksichtigt werden, bis hin zu tagesaktuellen Berichten. Die Berichte werden auf eine passwortgeschützte Webplattform gestellt, über die die Entscheidungsträger einen bequemen Zugang zu den ausschließlich für sie interessanten Berichten erhalten.

Dennoch gibt es Raum für Verbesserungen. Zwei Beispiele: Ausnahmen, wie 2-Fach- oder 3-Fach-Studierende sowie Frühstudenten, können im Vergleich zu anderen Studierenden deutlich andere Kennzahlen aufweisen und Statistiken verfälschen. Sie sollten separat betrachtet werden, sind jedoch nicht leicht zu identifizieren. Auch ist bereits Interesse an erweiterten Sichten bekundet worden, z.B. eine nach weiteren Veranstaltungen aufgeschlüsselte Studierendentabelle.

Bereits jetzt haben wir Lösungsansätze entwickelt, um mittelfristig auch folgende Fragen zu beantworten:

- Aus welchen Gründen brechen viele Studierende ihr Studium ab?
- Welche Bedeutung haben einzelne Veranstaltungen für den Erfolg des Studiums?
- In welchen wichtigen Kennzahlen unterscheiden sich die Studiengänge?

Abbrecher eines Studiengangs sind, bzw. waren Studierende mit bestimmten Leistungsdaten. Anhand dieser Daten lassen sich Abbrecher charakterisieren und Gründe für Abbrüche identifizieren. Wir haben mit der Data-Mining-Technik der sog. „Subgruppenentdeckung“ eine Möglichkeit gefunden, um dies zu tun – allerdings unter der Voraussetzung, dass wir verständliche und gleichzeitig aussagekräftige Abbrecherberichte, ähnlich zu den obigen Studententabellen, entwerfen.

Genauso empfehlen wir vorzugehen, um Gründe für „kritische“ Lehrveranstaltungen zu finden – einzelne Veranstaltungen, deren Bestehen oder Nicht-Bestehen über den Gesamterfolg eines Studierenden entscheiden können. Außerdem zeigen wir, dass auch komplette Studiengänge durch Kennzahlen beschrieben und verglichen werden können, ein Schritt mehr zum angestrebten Standardisierten Hochschulraum mit seinen antizipierten Vorteilen und möglichst wenig Nachteilen.

5.2 Business-Case

Im folgenden wird der Business-Case des Projekts Bachelor beschrieben. Hervorgehobene Begriffe wurden im Originaldokument in einem Glossar definiert, der auch hier unter Abschnitt 5.2.6 zu finden ist, aber auch über den Index dieser Arbeit erreicht werden kann.

Die Inhalte des Business-Case haben sich in Interviews zwischen den Projekt-Beteiligten sowie während der Projektdurchführung ergeben.

Zu Beginn des Projekts Bachelor haben wir eine Machbarkeitsstudie durchgeführt. Dafür haben wir vorab einen Teil der Daten erhalten, die uns während des Projekts zur Verfügung gestellt worden sind. Die Machbarkeitsstudie hat ca. 50 Stunden in Anspruch genommen. Folgende Ergebnisse für den Business-Case haben wir erzielt:

- Wir haben die Qualität der Daten überprüft, z.B., ob bei der Herstellung des Datenauszugs Fehler aufgetreten waren. Eine Datei war fehlerhaft und konnte erst nach einigen Modifikationen eingelesen werden.
- Wir haben Konzepte in den Daten identifiziert und ihnen Bezeichnungen gegeben, z.B. „Kohorte“, „Scheinstudenten“, „Abbrecher“ oder „Module“.
- Wir haben mögliche Schwierigkeiten identifiziert, z.B. Studienwechsler, die von Abbrechern nicht zu unterscheiden sind, oder Studenten, die Leistungen in Studiensemestern erbracht haben, in denen sie nicht eingeschrieben waren.
- Wir haben den Hintergrund, die Motivation und erste Fragestellungen des Projekts besser verstanden und Ideen zu möglichen Lösungen entwickelt, z.B. der Vergleich von Studenten, Kohorten, oder Lehrveranstaltungen.

Auch während des Projekts wurden mehrere Interviews mit den Entscheidungsträgern geführt, in denen wir gemeinsam die Anforderungen verfeinert haben.

5.2.1 Hintergrund und Motivation

Der Bologna-Prozess strebt eine Standardisierung des Hochschulraums vieler europäischer Staaten an, zur Förderung von Mobilität, internationaler Wettbewerbsfähigkeit und letztendlich

Beschäftigungsfähigkeit der Studierenden. In dessen Folge hat die Universität Würzburg die meisten Diplom- und Magisterstudiengänge auf das zweistufige System mit Bachelor und Master umgestellt.

Angesichts möglicher negativer Implikation – die zudem schwer zu überschauen sind – rücken die Ziele aus dem Bologna-Prozess in den Hintergrund. Beispiele für oft genannte Probleme sind: Verschulung des Studiums mit zu vielen Pflichtkursen, von denen einzelne eine überproportional hohe Belastung bedeuten; abschlussrelevante Prüfungen ab dem ersten Semester, die wenige Möglichkeiten für Auslandssemester, Praktika und außeruniversitäres Engagement lassen; erschwerte Bedingungen, um einen Masterstudiengang zu beginnen und das fachliche Niveau eines Diploms zu erlangen.

In Folge dieser Kritikpunkte könnten die Studentenzahlen sinken, mit negativen Auswirkungen auf den Hochschulstandort. Allerdings entstammen sie häufig pauschalisierten Meinungen einzelner Personen, gebildet aus subjektiven Erfahrungen. Um Änderungen zur Verbesserung der Studienbedingungen zu rechtfertigen, müssen Probleme belegt werden; eine möglichst objektive Beurteilung der Studiengänge ist nötig. Eine Voraussetzung dafür ist durch die vollständige elektronische Erfassung aller Bachelor/Master-Prüfungsdaten gegeben.

5.2.2 Problemstellung und Möglichkeiten

Folgende Aspekte sind wichtig, um den Erfolg eines Bachelorstudiengangs zu beurteilen:

- Das Verhältnis zwischen Anfängern, Abbrechern und Absolventen.
- Praktische Umsetzung von Vorgaben aus der Studienordnung.
- Zusammenspiel zwischen dem Arbeitsaufwand, dem Schwierigkeitsgrad und den resultierenden Abschlussnoten.
- Das Maß, in dem die Studenten über ihre Studiensituation aufgeklärt sind.
- Unterschiede verschiedener Studiengänge in allgemeingültigen Kennzahlen.

Aus diesen Kriterien wurden vier Problemstellungen abgeleitet, die behandelt werden sollen:

5.2.2.1 Analyse der Studienabbrecher

Es sollen die Gründe für Studienabbrecher untersucht werden. Zu beantworten sind die Fragen:

1. Wie entwickeln sich die Studierendenzahlen über die Zeit?
2. Wie entwickeln sich die Anzahl der *ECTS-Punkte* und Durchschnittsnoten der Studierenden?

3. Lassen sich bei Abbrechern wiederkehrende Gründe finden?

Manche Ursachen für Abbrüche können vermieden werden, wenn sie bekannt sind.

5.2.2.2 Analyse der Lehrveranstaltungen

Im Bezug auf Lehrveranstaltungen, sog. *Module*, sind folgende Fragestellungen interessant:

1. Wann werden die Lehrveranstaltungen bestanden und mit welcher Note?
2. Wie kritisch sind einzelne Veranstaltungen für den Studienerfolg?

Es stellt sich möglicherweise heraus, dass der Schwierigkeitsgrad mancher Veranstaltungen zu hoch und anzupassen ist. Auch kann mit solchen Informationen überprüft werden, ob der Studienverlaufsplan korrekt ist.

5.2.2.3 Analyse der Studierenden

Auch Studierende werfen Fragen auf:

1. Wieviele ECTS-Punkte erreichen sie, in welchen Lehrveranstaltungen und mit welcher Note?
2. Welche Studierende gelten als gefährdet?

Bei Abbrechern ist es meist zu spät, gefährdete Studenten dagegen können Hilfe, z.B. in Form von Beratungsgesprächen, erhalten.

5.2.2.4 Vergleich von Studiengängen

Bezüglich des Vergleichs von Studiengängen ist folgende Fragestellung zu beantworten:

1. Welche Studiengänge weichen in allgemeingültigen „Kennzahlen“ deutlich vom Durchschnitt ab?

Die für einen Studiengang ermittelten Kennzahlen lassen sich oft im Vergleich zu ähnlichen Studiengängen besser interpretieren. So lassen sich Ursachen für nicht-tolerierbare Abweichungen, z.B. zum Anspruch der Studiengänge, identifizieren und vermindern.

5.2.3 Aktuelle Situation und Datenlage

Für die Lösung der Fragestellung stehen elektronisch erfasste Studentendaten, Stammblätter, Prüfungsdaten und Prüfungsleistungen aller Bachelorstudiengänge aus der Universität, sowie

die Allgemeine (und ggf. jeweils fachspezifische) Studien- und Prüfungsordnung zur Verfügung. Im Folgenden werden diese Daten näher erläutert.

5.2.3.1 Studentendaten

Die Studentendaten enthalten für jeden Studierenden unter anderem die Staatsangehörigkeit, das Geburtsdatum und das Geschlecht. Außerdem die Art der Hochschulzugangsberechtigung (z.B. Abitur), das Jahr und die Note.

5.2.3.2 Stammblätter

Die Stammblätter enthalten für jeden eingeschriebenen Studenten pro Studiensemester folgende Daten: Studien- und Fachsemester, Informationen zum Studiengang sowie die geltende Prüfungsordnungsversion.

Ein Wort zum Datenschutz: Soweit es nicht für die Analysen andersweitig erforderlich ist, werden die Matrikelnummern in den Daten verschlüsselt, die Studenten also anonymisiert. Datenschutzrechtliche Bedenken gibt es daher nicht. Eine Zuordnung der anonymisierten Matrikelnummern zu Studenten — etwa zur Warnung von gefährdeten Studenten -– kann bei Bedarf durch die Auftraggeber erfolgen. Die Verschlüsselung ordnet identischen Matrikelnummern stets den selben Schlüssel zu, so dass nachträglich verschlüsselte, neue Daten, stets zu den bereits vorhandenen Daten hinzugefügt werden können.

5.2.3.3 Prüfungsdaten

In den Prüfungsdaten sind z.B. Prüfungsbezeichnungen (lang/kurz), die Prüfungsart (*Modul*, Teilmodul), veranschlagte Semesterwochenstunden und *ECTS-Punkte* sowie die geltende Prüfungsordnungsversion gespeichert.

5.2.3.4 Prüfungsleistungen

Für Prüfungsleistungen eines Studenten werden unter anderem Note, Datum, Prüfungssemester und Status (z.B. bestanden, nicht bestanden, endgültig nicht bestanden) gespeichert.

5.2.3.5 Studien- und Prüfungsordnung

Informationen zur Struktur eines Studiums (z.B. Unterscheidung zwischen Pflicht- und Wahlpflichtkursen) sind elektronisch erfasst und können entweder nach Bedarf manuell oder möglicherweise automatisch eingelesen werden.

Die gesamten Daten werden direkt von Universitätsmitarbeitern in das Hochschulinformationssystem eingegeben und dort automatisch auf Fehler überprüft. Es kann daher davon ausgegangen werden, dass sie eine hohe Qualität aufweisen.

Die vorgestellten Lösungen sollen keine Studiengänge speziell behandeln, sondern sich allgemein auf alle Bachelorstudiengänge beziehen. Des Weiteren soll es möglich sein, jederzeit aktualisierte Daten einlesen und die Lösungen wiederholen zu können.

5.2.4 Alternative und empfohlene Lösungen

Im Folgenden werden Anforderungen beschrieben, die die Fragestellungen behandeln und während des Projekts realisiert werden sollen. Außerdem wird geklärt, inwiefern alternative Lösungen, ggf. unter Verwendung von zusätzlichen Daten, möglich sind.

5.2.4.1 Anforderungen: Analyse der Studienabbrecher

In einer ersten Anforderung soll für jeden Studiengang eine Übersichtstabelle erstellt werden, die die Fragestellungen 1 und 2 aus Abschnitt 5.2.2.1 behandelt. Der Bericht dieser Anforderung zeigt pro Studiensemester, wieviele Studierende sich in welchem Fachsemester befunden haben, außerdem, wieviele ECTS-Punkte sie bis dahin im Durchschnitt erreicht und welche Noten sie im Durchschnitt erhalten haben. Noten sollen dabei nach der Anzahl an ECTS gewichtet werden – genauso, wie es häufig bei der Notenberechnung am Ende eines Studiums gemacht wird. Demnach zählt eine Note, die 16 ECTS-Punkte gebracht hat, doppelt so viel zum Durchschnitt, wie eine Note, die 8 ECTS-Punkte gebracht hat. Wichtig ist hierbei auch, dass Teilmodule, deren Module noch nicht bestanden werden, in die Durchschnittszahlen einfließen. Ansonsten könnten insbesondere die ECTS-Summen einen falschen Eindruck über die Studiensituation der Studierenden vermitteln.

Der Hauptzweck dieser Anforderung besteht darin, die Entwicklung von Studierendenzahlen, sowie durchschnittliche ECTS-Punkte und Noten der Studierenden darzustellen. Auch ein Eindruck über das Ausmaß an Abbrechern wird so erhalten.

Der Bericht soll für jeden Studiengang in Auszügen als PDF- und Excel-Dateien erstellt und auf DIN-A-4-Größe ausgedruckt werden können, um später an die Dekane der Universität weitergeleitet zu werden. Es wird angestrebt, dass die Dekane den Bericht direkt interpretieren können, weshalb keine Muster darin gesucht werden. Das Gleiche gilt für die Berichte aus den Anforderungen 5.3, 5.4, 5.6 und 5.7, die in späteren Abschnitten behandelt werden. Über eine passwortgeschützte Webplattform, erreichbar per Webbrowser, erhalten die Dekane einen bequemen Zugriff auf die Berichte; jeder Dekan kann dabei nur auf die Berichte seines Studiengangs zugreifen.

Name:	**Übersicht - Eingeschriebene Studenten mit Note und ECTS-Punkten**
Eingabe:	Studentendaten, Stammblätter, Prüfungsdaten, Prüfungsleistungen
Bericht:	Tabellen: Studienfach
	Zeilen: Fachsemester
	Spalten: Für jedes Studiensemester, die Anzahl/die kumulierte durchschnittliche ECTS-Punkte-Anzahl/die kumulierte Durchschnittsnote der im Studienfach eingeschriebenen Studierenden.
Muster:	-

Tab. 5.1: Anforderung: Übersicht Eingeschriebene Studenten

In einer weiteren Anforderung sollen die Gründe für Abbrecher untersucht werden und damit Frage 3 aus Abschnitt 5.2.2.1 beantwortet werden. Dazu soll ein Bericht erstellt werden, der eine detaillierte Betrachtung der Studierenden ermöglicht. Sie werden mit Eigenschaften beschrieben, von denen angenommen wird, dass sie den Grund eines Abbruchs beschreiben können. Vorab ist anzumerken, dass auch mehrere, sich beeinflussende Gründe für einen Abbruch verantwortlich sein können.

Mittels der Prüfungsleistungen lassen sich Abbrecher grob nach dem Hauptgrund für ihr Abbrechen unterteilen. Scheinstudenten (z.B. „parkende“ Studenten, die auf die Zulassung zu einem anderen Studiengang warten), ohne ernsthafte Studienabsicht, erbringen keinerlei Prüfungsleistungen und brechen ihr Studium ab, wenn in der Studienordnung genannte Mindeststudienleistungen erforderlich werden. Wieder andere Gründe haben Studenten, die abbrechen, obwohl sie die erforderten Leistungen des Studiums, etwa genügend ECTS-Punkte, erreichen. Beispiele sind Studiengangs- oder Hochschulwechsler. Da nur Studentendaten aus der Universität Würzburg vorliegen, kann ein Wechsel an eine andere Hochschule nicht eindeutig festgestellt werden. Scheinstudenten und Studienwechsler lassen sich möglicherweise anhand ihrer Leistungen in Prüfungen erkennen. Daher soll für die Studierenden im Bericht enthalten sein, wieviele ECTS-Punkte sie in ihren Fachsemestern erreicht haben. Dabei werden die ECTS-Punkte pro Studiensemester in aussagekräftige Kategorien diskretisiert, nämlich in die Gruppen „< 5“, „$5 <= X < 15$“, „$15 <= X < 25$“, „$25 <= X < 35$“ und „$>= 35$“.

Ein weiterer Grund für ein Abbrechen sind einzelne, zu komplizierte Pflichtveranstaltungen, die ein Studierender trotz genügender Zeit nicht besteht, und die ihn zum Gesamtabbruch seines Studiums zwingen. Eine Veranstaltung kann aus verschiedenen Ursachen zu kompliziert sein, z.B. fehlende Sprachkenntnisse, fehlende Grundkenntnisse oder zu hohe Anforderungen. Auch diese Ursachen gilt es zu untersuchen:

Der Bericht der Anforderung soll für jedes Modul die Information darüber enthalten, ob der

Student das Modul bereits bestanden hat. Möglicherweise kristallisieren sich Veranstaltungen heraus, die von sehr wenigen Abbrechern bestanden werden.

Sprachkenntnisse können heuristisch anhand der Staatszugehörigkeit aus den Studentendaten abgeschätzt werden. Studenten, z.B. aus Deutschland oder Österreich, werden im Bericht als „deutschsprachig“, andere Studenten als „nicht deutschsprachig“ aufgelistet.

Die Hochschulzugangsberechtigung kann Aussagen zum Ausgangsniveau der Studierenden enthalten. Daher soll die Note der Hochschulzugangsberechtigung im Bericht angegeben sein, allerdings gerundet auf ganzzahlige Werte. Auch das Alter bei Studienbeginn gibt Auskunft zu möglichen Vorkenntnissen eines Studierenden, z.B. aus sonstigen Weiterbildungen. Hier soll zwischen einem Alter unter 21, zwischen 21 und 25 und über 25 unterschieden werden.

Auch übermäßig anspruchsvolle Prüfungsvorleistungen, z.B. in Übungsveranstaltungen oder Präsenzaufgaben, sind möglich. Als ergänzende Datenquelle können daher Prüfungsvorleistungen dienen. Solche Daten werden nicht zwingend elektronisch aufgezeichnet und können eine niedrige Qualität aufweisen. Zudem unterscheiden sich die Prüfungsvorleistungen zwischen einzelnen Studiengängen und sogar zwischen einzelnen Veranstaltungen eines Studiengangs. Daher werden diese Daten vorläufig nicht berücksichtigt.

Ein weiterer Grund für Abbrecher ist eine zu hohe Arbeitsbelastung. Trotz theoretisch machbarer *Module* bleibt solchen Studenten neben Beschäftigungen wie Jobs, Kindern oder Hobbies zu wenig Zeit für ihr Studium. In Folge dessen brechen sie dieses ab, weil sie die Vorgaben aus der Studienordnung, etwa die Einzelprüfungen in der Studienhöchstdauer, nicht erreichen.

Zur Unterscheidung dieser Abbrecher kann die zeitliche Belastung der Studenten erfasst werden, z.B. die Zeit für außeruniversitäre Aktivitäten. Diese sind nur durch Zusatzdaten aus Umfragen festzustellen. Ähnliche Umfragen werden bisher nur in einzelnen Studiengängen durchgeführt, und selbst dort nur mit geringen Rücklaufquoten und niedriger Repräsentativität.

Innerhalb des Berichts sollen anschließend Ausprägungen von Eigenschaften entdeckt werden, die als Gründe für Abbrecher bekräftigt werden können. Tabelle 5.2 zeigt die gesamte Anforderung.

5.2.4.2 Anforderungen: Analyse der Lehrveranstaltungen

Um Fragestellung 1 aus Abschnitt 5.2.2.2 zu behandeln, haben wir zwei Anforderungen entwickelt, die jeweils eine Sicht auf Lehrveranstaltungen bieten.

In der ersten Anforderung wird dabei ein Bericht erstellt, der für jedes Modul und Teilmodul angibt, wieviele der aktuell Studierenden es in welchem Fachsemester bestanden haben. Ursprünglich war geplant, diese Zahl prozentual zur Anzahl der aktuell Studierenden anzugeben. Allerdings hatten Studierende, die sich momentan in einem Fachsemester befinden, meist noch nicht die Möglichkeit, diese Veranstaltung im jeweiligen Semester bestanden zu haben; die Pro-

Name:	**Abbrecher Eigenschaften**
Eingabe:	Studentendaten, Stammblätter, Prüfungsdaten, Prüfungsleistungen
Bericht:	Tabellen: Studienfach
	Zeilen: Abbrecher und Nicht-Abbrecher in Studienfach
	Spalten: Kohorte; Geschlecht; Sprache; Hochschulzugangsart; -note; Alter bei Studienbeginn; ob Abbrecher; für jedes Fachsemester die Anzahl an ECTS-Punkten (diskretisiert); für jedes Modul, ob es bestanden wurde
Muster:	Attributwerte, mit denen sich Abbrecher charakterisieren lassen.

Tab. 5.2: Anforderung: Abbrecher Eigenschaften

zentzahlen wären über diese Studierenden vermindert worden und schwer zu interpretieren. Die kumulierten Zahlen dagegen sind korrekt, geben jedoch per se keine Auskunft darüber, wieviele der aktuell Studierenden eine Veranstaltung bisher bestanden haben. Deshalb werden in zwei weiteren Spalten diese Gesamtanzahl und die Anzahl an Studenten im ersten Fachsemester angegeben – die zweite Spalte aus dem Grund, dass diese Studierenden in der Regel noch keine Veranstaltungen bestanden haben können.

Tabelle 5.3 enthält eine Zusammenfassung dieser Anforderung.

Name:	**Lehrveranstaltungen nach kumulierten Fachsemestern**
Eingabe:	Studentendaten, Stammblätter, Prüfungsdaten, Prüfungsleistungen
Bericht:	Tabellen: Studienfach
	Zeilen: Modul oder Teilmodul
	Spalten: Prüfungsname; Prüfungskennzeichen; Gesamtzahl der aktuell Studierenden (Nicht-Abbrecher); Gesamtzahl der aktuell Studierenden im ersten Fachsemester; je Fachsemester Anzahl der aktuell Studierenden mit bestandener Veranstaltung
Muster:	Zuerst sollen Module und Teilmodule aus einem Pflichtbereich, dann erst die restlichen Module und Teilmodule angezeigt werden. Jeweils alphabetisch sortiert.

Tab. 5.3: Anforderung: Lehrveranstaltungen nach kumulierten Fachsemestern

In der zweiten Anforderung soll detaillierter das aktuelle Semester betrachtet werden. Ein Bericht gibt in einer ersten Menge an Spalten an, wieviele Studierende, die sich momentan in einem bestimmten Fachsemester befinden, eine Veranstaltung bereits bestanden haben, zudem mit welcher Durchschnittsnote. In weiteren Spalten wird die Anzahl auch prozentual von der Gesamtzahl der aktuell Studierenden, die sich in diesem Fachsemester befinden, angegeben. Des

Weiteren wird in zusätzlichen Spalten für jedes Fachsemester die Durchschnittsnote angegeben.

Tabelle 5.4 fasst diese Anforderung zusammen.

Name:	**Lehrveranstaltungen nach aktuellen Fachsemestern**
Eingabe:	Studentendaten, Stammblätter, Prüfungsdaten, Prüfungsleistungen
Bericht:	Tabellen: Studienfach
	Zeilen: Modul oder Teilmodul
	Spalten: Für jedes Fachsemester der aktuell Studierenden, Anzahl Studierende, die Veranstaltungen bisher bestanden haben/Prozentsatz an eingeschriebenen Studierenden/Durchschnittsnote
Muster:	Zuerst sollen Module und Teilmodule aus einem Pflichtbereich, dann erst die restlichen Module und Teilmodule angezeigt werden. Jeweils alphabetisch sortiert.

Tab. 5.4: Anforderung: Lehrveranstaltungen nach aktuellen Fachsemestern

Von besonderer Bedeutung sind immer die Pflichtveranstaltungen der Studierenden, weshalb diese als erstes in den beiden Berichten gezeigt werden sollen. Da wir keine Möglichkeit gefunden haben, um Pflichtveranstaltungen eindeutig als solche zu identifizieren, verwenden wir eine Heuristik. Demnach gilt ein Bereich als pflichtig, wenn sein Name mit „Pflicht“ beginnt.

Auch diese beiden Berichte sollen auf DIN-A-4-Größe passen, weshalb die Länge der Spalten begrenzt ist. Für jedes Modul oder Teilmodul werden der Kurzname, und ein Kennzeichen, bestehend aus drei Kleinbuchstaben verwendet. Eine Legende wird dem Bericht beigefügt, mit der, wenn die Spaltenbreite zu klein ist, um den Kurznamen anzuzeigen, zumindest über das Kennzeichen das Modul erkannt werden kann. Daneben soll im Kurznamen der Module und Teilmodule die Studienfachkennzeichnung, ausgedrückt durch eine Zahl am Anfang, aus Platzgründen entfernt werden. Dies soll grundsätzlich für alle Berichte gemacht werden. Von beiden Berichten wird angenommen, dass sie direkt verständlich sind; weitere Muster müssen nicht darin gefunden werden.

Werden bestimmte Pflichtveranstaltungen seltener bzw. später als im Durchschnitt oder in der Studienordnung vorgesehen bestanden, ggf. mit einer schlechteren Note, weist dies auf übermäßig anspruchsvolle oder aufwändige Veranstaltungen hin. Solche kritischen Verstanstaltungen können zu einem Abbruch des Studiums führen — wie bereits in Abschnitt 5.2.4.1 erklärt. Um nun Fragestellung 2 aus Abschnitt 5.2.2.2 zu behandeln und herauszufinden, ob bestimmte Veranstaltungen kritisch sind, kann ihr Einfluss auf Abbrecher untersucht werden.

Daher wird in einer weiteren Anforderung, siehe Tabelle 5.5, ein Bericht erstellt, der für jedes Modul eines Studienfachs den Prozentsatz an Abbrechern unter den Studierenden errechnet, die das Modul bestanden, außerdem den Prozentsatz an Abbrechern unter den Studierenden

errechnet, die das Modul nicht bestanden haben. Zur Verdeutlichung der Relevanz werden außerdem absolute Zahlen genannt. Module, die von vielen Abbrechern nicht bestanden worden sind, werden im Bericht hervorgehoben. Die Prozentzahlen in diesem Bericht geben vergleichbar der Trennschärfe aus der Statistik einen Eindruck darüber, wie stark das Bestehen einer Prüfung zwischen Abbrechern und Nicht-Abbrechern trennt.

Name:	**Veranstaltungen Trennschärfe**
Eingabe:	Studentendaten, Stammblätter, Prüfungsdaten, Prüfungsleistungen
Bericht:	Tabellen: Studienfach
	Zeilen: Modul
	Spalten: Anzahl Studierende, die Modul bestanden; Anzahl Nicht-Abbrecher, die Modul bestanden; Anzahl Abbrecher, die Modul bestanden; Prozentsatz Abbrecher von Studierenden, die bestanden; Prozentsatz Abbrecher von Studierenden, die nicht bestanden
Muster:	Absteigend sortiert nach Prozentsatz Abbrecher, die nicht bestanden.

Tab. 5.5: Anforderung: Veranstaltungen Trennschärfe

5.2.4.3 Anforderungen: Analyse der Studierenden

Um die Fragestellung 1 aus Abschnitt 5.2.2.3 zu behandeln, werden in zwei Anforderungen die aktuell Studierenden näher betrachtet.

Im Bericht der ersten Anforderung werden für jeden Studierenden zunächst das aktuelle Fachsemester, das Einstiegssemester und das Hochschulsemester angegeben. Diese Kennzahlen lassen auf Studierende, die direkt in einem höheren Fachsemester einsteigen sowie Urlaubssemester schließen.

Dann werden die ECTS-Punkte angegeben, zunächst die Gesamtsumme. Anschließend der Durchschnitt pro Fachsemester. Diese Zahl wird meist missverständlich sein, denn normalerweise hat ein Student im aktuellen Fachsemester noch keine ECTS-Punkte erreicht. Daher wird dieser Durchschnitt zusätzlich mit der um eins verminderten Anzahl Fachsemester berechnet. Wie auch in Anforderung 5.1 ist es wichtig, Teilmodule, deren Module noch nicht bestanden wurden, einzubeziehen. Neben den ECTS-Berechnungen wird zusätzlich der Notendurchschnitt angegeben. In weiteren Spalten werden für jeden Prüfungsbereich die ECTS-Punkte und der Notendurchschnitt angegeben. Tabelle 5.6 fasst die Anforderung zusammen.

Dieser Bericht soll primär absteigend nach dem aktuellen Fachsemester sortiert werden, um Studierende in einem höheren Semester weiter oben anzuzeigen. Innerhalb eines Fachsemesters soll absteigend nach ECTS-Punkten sortiert werden, damit die besten einer *Kohorte* sichtbar

werden.

Name:	**Studenten Komprimiert**
Eingabe:	Studentendaten, Stammblätter, Prüfungsdaten, Prüfungsleistungen
Bericht:	Tabellen: Studienfach
	Zeilen: Studierender in Studienfach
	Spalten: Aktuelles Fachsemester; Einstiegssemester; Hochschulsemester; Summe ECTS-Punkte; ECTS-Durchschnitt pro Fachsemester; ECTS-Durchschnitt pro beendetes Fachsemester; Durchschnittsnote; für jeden Bereich, ECTS-Summe; für jeden Bereich, Durchschnittsnote.
Muster:	Absteigend nach dem aktuellen Fachsemester sortiert. Für jedes Fachsemester absteigend nach Summe ECTS-Punkte sortiert.

Tab. 5.6: Anforderung: Studenten Komprimiert

Um näher zu zeigen, in welchen Pflichtveranstaltungen die Studierenden ihre ECTS-Punkte erreichen, sollen solche Informationen in einer detaillierteren Version des obigen Berichts hinzugefügt werden. Wie in Tabelle 5.7 beschrieben, enthält dieser Bericht die Noten, die der Student in Modulen bzw. in nicht-abgeschlossenen Teilmodulen aus dem Pflichtbereich erhalten hat. Der Name eines solchen Bereichs beginnt mit „Pflicht“.

Name:	**Studenten – Detailliert**
Eingabe:	Studentendaten, Stammblätter, Prüfungsdaten, Prüfungsleistungen
Bericht:	Tabellen: Studienfach
	Zeilen: Student aus Studienfach
	Spalten: siehe Anforderung 5.6; Für jeden Pflichtbereich, für jedes Modul oder Teilmodul die Note
Muster:	siehe Anforderung 5.6

Tab. 5.7: Anforderung: Studenten Detailliert

Für den Fall, dass die Namen der Veranstaltungen oder Bereiche auf Grund der begrenzten Spaltenbreite im DIN-A-4-Dokument nicht mehr lesbar sind, sollen für beide auch Kennzeichen angegeben werden, die in einer Legende zum Bericht nachgeschlagen werden können.

Der Bachelorstudiengang sieht durchschnittlich 30 *ECTS-Punkte* pro Fachsemester vor. Nach sechs Semestern können die benötigten 180 ECTS-Punkte für den Bachelorabschluss erreicht werden. Als kritische Grenze werden ca. 20 ECTS-Punkte pro Semester gesehen, bei denen das Studium noch innerhalb der Maximaldauer von 9 Semestern erfolgreich absolviert werden kann. Studierende, die unter diesem Schnitt liegen, sind gefährdet, ihr Studium abbrechen zu müssen; dies wird in Frage 2 aus Abschnitt 5.2.2.3 angesprochen. Aus den Berichten der Tabellen 5.6 und

5.7 können die ECTS-Punkte der Studierenden abgefragt werden. Um den Grad der Gefährdung eines Studenten besser einschätzen zu können, soll in einer weiteren Anforderung, siehe Tabelle 5.8, speziell die Verteilung der durchschnittlichen ECTS-Punkte pro Semester für Abbrecher und Nicht-Abbrecher eines Studienfachs untersucht werden.

Name:	**ECTS-Verteilung**
Eingabe:	Studentendaten, Stammblätter, Prüfungsdaten, Prüfungsleistungen
Bericht:	Tabellen: Studienfach
	Zeilen: Student aus Studienfach
	Spalten: ob Abbrecher; Anzahl ECTS-Punkte pro eingeschriebenes Fachsemester
Muster:	Darstellung der Verteilung der ECTS-Punkte.

Tab. 5.8: Anforderung: ECTS-Verteilung

5.2.4.4 Anforderungen: Vergleich von Studiengängen

Frage 1 aus Abschnitt 5.2.2.4 nennt den Vergleich der Studiengänge als interessante Informationsquelle. Deshalb soll in einer Anforderung (siehe Tabelle 5.9) herausgefunden werden, welche Studienfächer hinsichtlich einiger wichtiger Kennzahlen deutlich vom Durchschnitt abweichen: Diese sind die Durchschnittsnoten, Abbrecherquoten, durchschnittlichen *ECTS-Punkte* und Prüfungswiederholungen der Studierenden. Um den Vergleich zu vereinfachen, sollen sie auf eine prozentuale Abweichung vom Durchschnitt normiert werden.

Name:	**Vergleich Studiengänge**
Eingabe:	Studentendaten, Stammblätter, Prüfungsdaten, Prüfungsleistungen
Bericht:	Zeilen: Studienfach
	Spalten: Anzahl Studenten; Note Durchschnitt normiert; Abbrecher Prozentsatz normiert; ECTS-Summe pro Student und Studiensemester normiert; Wiederholungen Durchschnitt normiert.
Muster:	Nach Bedarf nach einer der Kennzahlen sortiert.

Tab. 5.9: Anforderung: Vergleich Studiengänge

Wir vermuten, dass Unterschiede zwischen Studiengängen besonders deutlich werden, wenn Studierende verschiedener Studienfächer dieselbe Veranstaltung besuchen. Deshalb werden wir in einer weiteren Anforderung für solche interdisziplinären Module die Leistungen der Studierenden aus verschiedenen Studienfächern auflisten und, normiert auf die prozentuale Abweichung vom Durchschnitt, die durchschnittliche Note und Wiederholungsanzahl angeben, siehe Tabelle

5.10. Diese Anforderung kann Indizien liefern, dass Studierende bestimmter Studienrichtungen in einem Modul über- oder unterfordert werden.

Name:	**Interdisziplinäre Module**
Eingabe:	Studentendaten, Stammblätter, Prüfungsdaten, Prüfungsleistungen
Bericht:	Tabellen: Interdisziplinäres Modul
	Zeilen: Studienfach
	Spalten: Note Durchschnitt; Note Durchschnitt normiert; Wiederholungen Durchschnitt; Wiederholungen Durchschnitt normiert.
Muster:	Nach Bedarf nach einer der Kennzahlen sortiert.

Tab. 5.10: Anforderung: Interdisziplinäre Module

5.2.5 Projektplanung

5.2.5.1 Beteiligte

Entscheidungsträger und Auftraggeber sind Professor Frank Puppe und Professor Jürgen Wolff von Gudenberg. Indirekte Entscheidungsträger sind die Studiendekane der Universität, denen Teile der Ergebnisse vorgelegt werden sollen. Professor Puppe besitzt als Studiendekan Domänen-Expertise. Daten-Experten sind Marianus Ifland und Florian Lemmerich, Mitarbeiter seines Lehrstuhls, die im Kontakt zu einem Administrator des Hochschulinformationssystems der Universität stehen und darüber das nötige Hintergrundwissen zu den zu analysierenden Daten besitzen. Der Autor dieser Arbeit dient als Data-Mining-Experte.

5.2.5.2 Ressourcen

Es werden ausschließlich kostenfrei nutzbare Open-Source-Werkzeuge eingesetzt. Als Hardware werden drei herkömmliche Arbeitsplatzrechner benötigt, einer für ein Data-Warehouse, ein zweiter für die Analysen, ein dritter zur Verwaltung der Dokumentation.

5.2.5.3 Projektplan

Der Projektplan teilt sich in folgende Meilensteine auf:

Data-Assay Es werden die Daten beschrieben und aufbereitet, so dass sie für die Anforderungen geeignet sind.

Data-Warehouse Es wird ein Data-Warehouse erstellt, das Abfragen ermöglicht, um die angeforderten Berichte zu erstellen.

Reporting Es werden die angeforderten Berichte erstellt.

Data-Mining Es werden in den Berichten die angeforderten Muster gesucht.

Anpassungen der Anforderungen werden dem Entscheidungsträger vorgelegt. Der Entscheidungsträger kann während des Projekts stets den Status Quo in Form von Beschreibungen des Data-Assay, Demonstrationen des Data-Warehouse, Auszüge der Reports und Demonstrationen des Data-Mining erhalten.

Am Ende des Projekts erhält der Entscheidungsträger die Ergebnisse der Anforderungen in Form einer Business-Story.

5.2.6 Glossar

5.2.6.1 ECTS-Punkte

European-Credit-Transfer-System-Punkte werden beim Bestehen eines *Moduls* erhalten und beschreiben die studentischen Leistungen.

5.2.6.2 Kohorte

Als Kohorte wird in diesem Zusammenhang eine Menge an Studenten bezeichnet, die im selben Semester begonnen haben, ein Studienfach an der Universität zu studieren. Diese Studenten müssen nicht zwingend mit dem ersten Fachsemester beginnen. Eine Kohorte wird durch das Studiensemester gekennzeichnet.

5.2.6.3 Modul

Module sind Einheiten eines Studiengangs, für die ein Student *ECTS-Punkte* und eine Note erlangen kann. Sie bestehen aus einigen wenigen Lehrveranstaltungen, an deren Ende eine studienbegleitende Prüfung zum Thema des Moduls steht. Module können weiter in Teilmodule aufgeteilt werden, für die eigene ECTS-Punkte und Noten erhalten werden und in ihrer Gesamtheit das Ergebnis im Modul ausmachen.

5.3 Business-Story

Die Business-Story wird dem Auftraggeber als Ergebnis des Projekts übergeben. Informationen darüber, wie die Ergebnisse erzeugt wurden, sind in den darauffolgenden Kapiteln enthalten.

Zur Business-Story gehört ein Management-Summary, das dieser Einzelfallstudie als Einleitung vorangesetzt wurde, siehe Abschnitt 5.1.

5.3.1 Ziel des Projekts Bachelor

Ziel des Projekts ist eine Bewertung der neu eingeführten Bachelorstudiengänge der Universität Würzburg mittels elektronisch erfassten Studiendaten, um mögliche negative Auswirkungen der Umstellung von Diplom- auf Bachelor-/Masterabschlüsse auf die Studienqualität möglichst früh feststellen und beheben zu können.

5.3.2 Ergebnisse

Im Folgenden wird auf jedes der Teilziele anhand der Anforderungen aus dem Business-Case eingegangen.

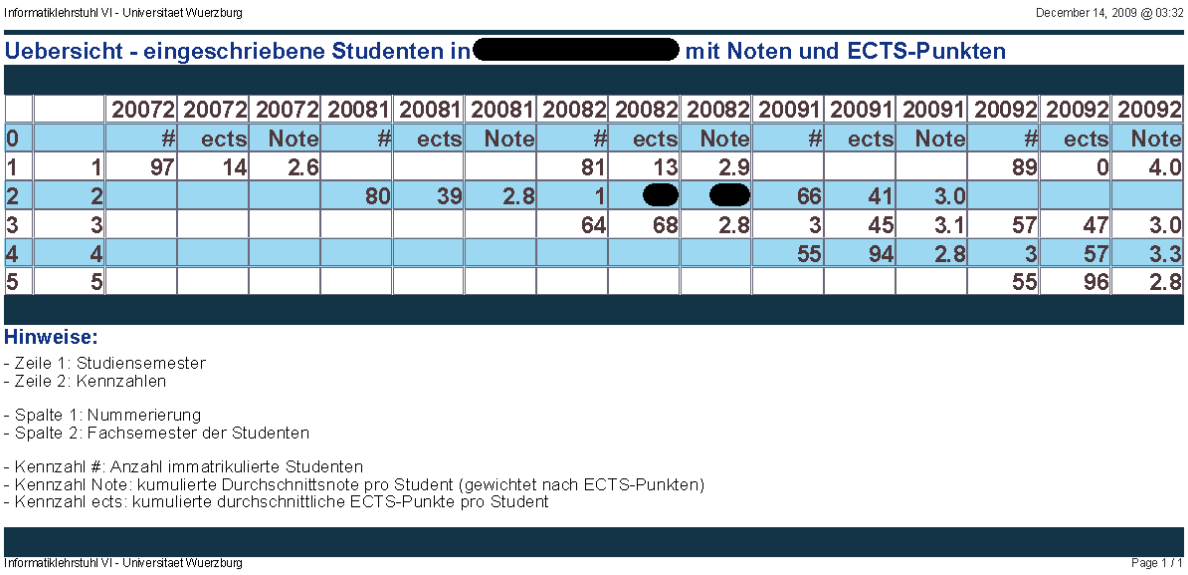
Am 19. November 2009 wurden uns Daten zu insgesamt 33 Studiengängen übergeben, für die wir die Anforderungen erfüllt haben, die im Business-Case festgelegt worden sind.

5.3.2.1 Ergebnis: Analyse der Studienabbrecher

Abbildung 5.1 zeigt ein anonymisiertes Beispiel des Berichts, den wir nach Tabelle 5.1 umgesetzt haben, um zu zeigen, wie sich die Anzahl Studierende, ihre erreichten ECTS-Punkte und ihre Noten über die Zeit entwickeln.

In diesem Bericht kann man eine *Kohorte* im Zeitverlauf von links oben nach rechts unten verfolgen. Beispielsweise entwickelt sich in der Abbildung 5.1 die Kohorte 20072, deren Studenten im Wintersemester 2007 mit dem Studium begonnen haben, folgendermaßen: Die Anzahl an Studierenden reduziert sich von anfangs 97 Studenten, im zweiten Semester auf 80, dann auf 64, und dann noch einmal auf 55. Den Übergang ins 5. Semester haben dann alle 55 Studierende geschafft. Die ECTS-Punkte der Studierenden werden vom ersten auf das zweite Semester zunächst mehr als verdoppelt, die anschließenden ECTS-Sprünge fallen jedoch geringer aus. Das letzte Studiensemester, hier 20092, ist normalerweise nicht aussagekräftig, da es noch nicht abgeschlossen ist und dementsprechend wenig Leistungen elektronisch erfasst sein können. Die Note verschlechtert sich im zweiten Semester von 2,6 auf 2,8 und bleibt dann konstant. Es sei anzumerken, dass Studenten auch in einem Fachsemester stagnieren können, z.B. nach einem Urlaubssemester; deshalb sind die Kohorten nicht vollständig verfolgbar.

Der Bericht ist allgemein gut geeignet, um Regelmäßigkeiten zu erkennen. Zum Beispiel könnte sich zeigen, dass es anfangs viele Scheinstudenten ohne jegliche ECTS-Punkte gibt, diese ihr



Informatiklehrstuhl VI - Universitaet Wuerzburg — December 14, 2009 @ 03:32

Uebersicht - eingeschriebene Studenten in [redacted] mit Noten und ECTS-Punkten

		20072	20072	20072	20081	20081	20081	20082	20082	20082	20091	20091	20091	20092	20092	20092
0		#	ects	Note	#	ects	Note	#	ects	Note	#	ects	Note	#	ects	Note
1	1	97	14	2.6				81	13	2.9				89	0	4.0
2	2				80	39	2.8	1	[redacted]	[redacted]	66	41	3.0			
3	3							64	68	2.8	3	45	3.1	57	47	3.0
4	4										55	94	2.8	3	57	3.3
5	5													55	96	2.8

Hinweise:

- Zeile 1: Studiensemester
- Zeile 2: Kennzahlen

- Spalte 1: Nummerierung
- Spalte 2: Fachsemester der Studenten

- Kennzahl #: Anzahl immatrikulierte Studenten
- Kennzahl Note: kumulierte Durchschnittsnote pro Student (gewichtet nach ECTS-Punkten)
- Kennzahl ects: kumulierte durchschnittliche ECTS-Punkte pro Student

Informatiklehrstuhl VI - Universitaet Wuerzburg — Page 1 / 1

Abb. 5.1: Beispiel Übersicht - Eingeschriebene Studenten mit Note und ECTS-Punkten

Studium aber schnell beenden. Manche Studienfächer sind möglicherweise anfangs etwas schwerer – mit entsprechend hohen Abbrecherquoten – werden mit der Zeit jedoch leichter. Über mehrere Kohorten hinweg lassen sich auch Tendenzen hinsichtlich der Abbrecherquote feststellen.

Damit Dekane diese Berichte weiterleiten können, dürfen keine personenbezogenen Daten enthalten sein. Studiensemester, in denen für ein Fachsemester nur ein Student eingeschrieben ist, lassen leicht auf eine Person schließen. Die ECTS-Punkte und Noten solcher Studenten dürfen nicht angezeigt werden. Dies muss konsequent auch für weiterführende Berichte gelten. Eine solche Adaption der Berichte ist noch umzusetzen, weshalb wir sie in Abbildung 5.1 manuell vorgenommen haben.

Wir haben erfahren, dass in manchen Studiengängen die Abschlussnoten nicht gewichtet nach ECTS-Punkten der Lehrveranstaltungen berechnet werden, sondern durch andere Formeln, z.B. nach individuell gewichteten Bereichen. Je nach Komplexität einer Formel lässt sich das direkt in den Berichten berücksichtigen, oder erst nach Änderungen im Data-Warehouse.

Nach diesem Überblick über Studierende und Abbrecher haben wir über die Anforderung aus Tabelle 5.2 auch eine Möglichkeit erhalten, um Gründe für Abbrecher festzustellen. Dabei suchen wir nach Gruppen von Studierenden, die gemeinsame Eigenschaften aufweisen und eine hohe Wahrscheinlichkeit haben, ihr Studium abzubrechen. Ein Beispiel für solche Gruppen bietet Tabelle 5.11. Sie dient der Verdeutlichung möglicher Ergebnisse und wurde daher anonymisiert. Wir haben darin den Bericht der Anforderung für ein spezielles Studienfach erstellt und folgende Subgruppen gefunden: Studierende, die Modul 1 nicht bestanden haben, brechen demnach besonders häufig auch ihr Studium ab, und zwar zu 42,1 Prozent unter den 38 Studierenden der

Subgruppe, im Gegensatz zu 20,3% unter allen 123 Studierenden des Studienfachs. Ein solches Modul kann entweder früh im Studium angesetzt sein und deshalb ein Grund für das Abbrechen sein; oder es kann spät im Studium angesetzt und vielmehr deshalb noch nicht bestanden worden sein, weil Abbrecher meist früh und damit vor dem möglichen Bestehen dieses Moduls abbrechen. Diese Muster in den Daten müssen also meist mit Hintergrundwissen betrachtet werden. Die Anzahl der Studenten (# Subgruppe), die in dieser Subgruppe zusammengefasst sind, können die Bedeutung einer Subgruppe relativieren, vor allem im Vergleich mit der Gesamtanzahl an Studenten (# Population). Zwei weitere Subgruppen wurden gefunden. Erstens solche Studenten, die in ihrem ersten Semester zwischen 5 und 15 ECTS-Punkte erreicht haben. Außerdem weibliche Studierende, die Modul 2 nicht bestanden haben.

Subgruppen-Beschreibung	**# Population**	**# Subgruppe**	**% Population**	**% Subgruppe**
$Modul1 = 0$	123	38	20,3%	42,1%
$ECTS\_Sem\_1 = 5 <= X < 15$	123	27	20,3%	37,0%
$Person = W$ AND $Modul2 = 0$	123	22	20,3%	49,3%

Tab. 5.11: Beispiel Subgruppenentdeckung zu Abbrecher Eigenschaften

Die bisherigen Ergebnisse enthalten Einschränkungen: Abbrecher und Absolventen werden von uns nicht unterschieden. Daher kann der Bericht für Studienfächer, in denen es bereits Absolventen gibt, nicht verwendet werden. Um diese Einschränkung aufzuheben, müsste eine Möglichkeit gefunden werden, um Absolventen eindeutig zu erkennen, und das Data-Warehouse dementsprechend erweitert werden. Absolventen heuristisch über die Anzahl an erreichten ECTS-Punkten zu identifizieren, halten wir für unzureichend, da insbesondere späte Abbrecher, mit ggf. vielen ECTS-Punkten, ein ernstes Problem in einem Studiengang darstellen und nicht ignoriert werden dürfen. Des Weiteren wurden für die ECTS-Summen bisher nur Leistungen in Modulen berücksichtigt. Leistungen in Teilmodulen, deren Modul noch nicht bestanden ist, müssen jedoch für eine korrekte Einschätzung inkludiert werden. Dies ist in anderen Anforderungen bereits umgesetzt worden.

Wenn diese Einschränkungen behoben werden, kann die Anforderung auch erweitert werden; im Folgenden einige Beispiele:

- Hinzufügen neuer möglicher Einflussfaktoren auf die Abbrecherquote, z.B. aus Ergebnissen von Umfragen der Studierenden.
- Untersuchen von studiengangsübergreifenden Teilmengen von Studierenden, z.B. Geisteswissenschaftler, Studierende der ersten G8-Abiturjahrgänge.

- Untersuchen von Einflussfaktoren auf weitere Eigenschaften, z.B. Note, ECTS-Punkte, Prüfungswiederholungen, Studiendauer.
- Untersuchen von speziellen Studierendengruppen, z.B. Frühstudenten, Quereinsteiger, 2-Fach oder 3-Fach-Studierende, einzelne Kohorten oder Studierende aus einer konkreten Prüfungsveranstaltung.
- Automatisches Einteilen der Studierende in Kategorien, z.B. Scheinstudenten, „gute“ Studierende, „gefährdete“ Studierende. Diese können anschließend näher untersucht werden.

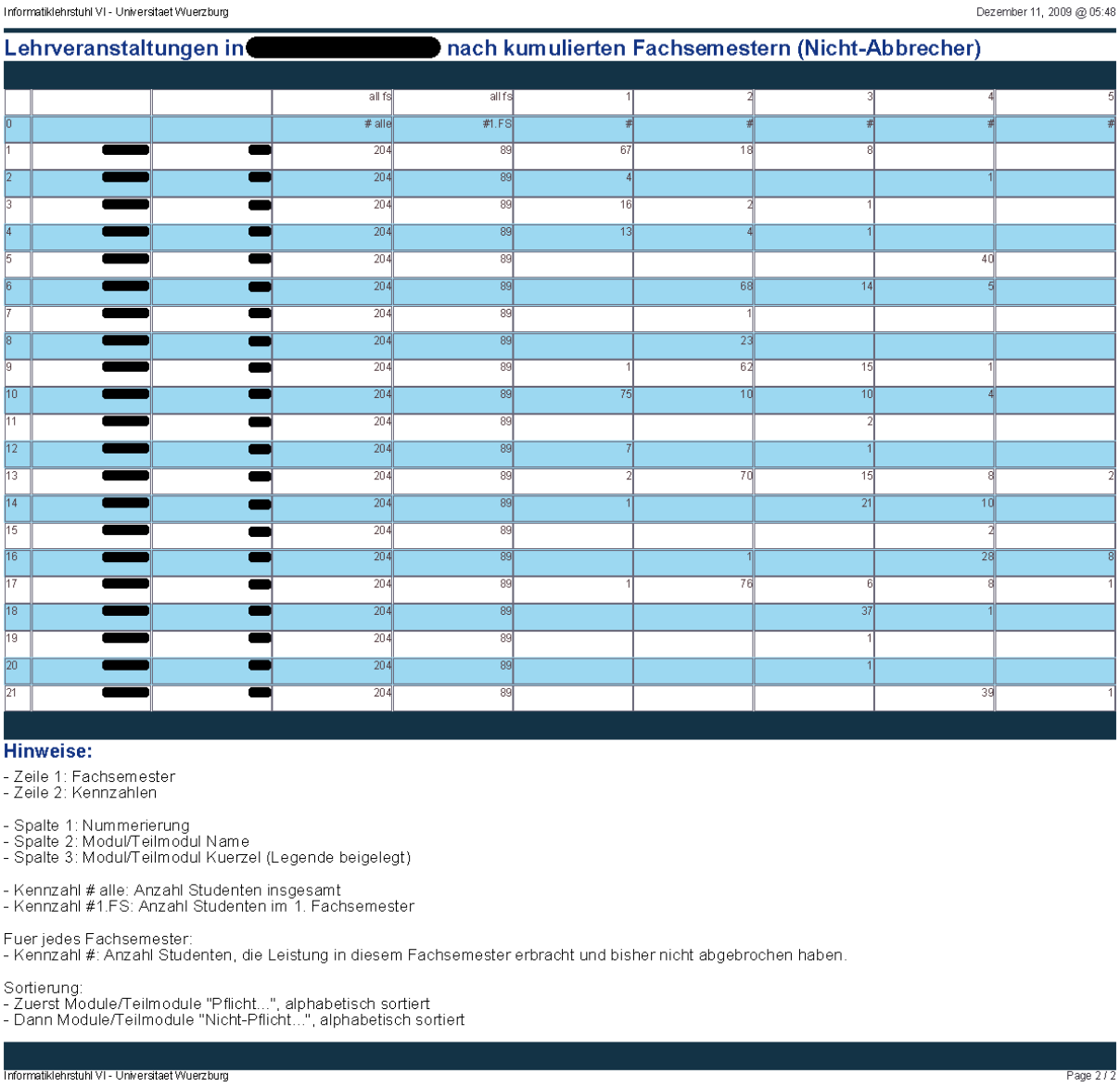
5.3.2.2 Ergebnis: Analyse der Lehrveranstaltungen

Unser erstes Ziel der Analyse von Lehrveranstaltungen war es, herauszufinden, wann und mit welchen Noten sie bestanden werden. Dies erreichen wir mit zwei Berichten. Diese können von Dekanen verwendet werden, um den Studienverlaufsplan bezüglich Veranstaltungen auf Unstimmigkeiten zu überprüfen.

Abbildung 5.2 zeigt ein Beispiel des umgesetzten Berichts aus Tabelle 5.3. Sowohl die Namen der Veranstaltungen als auch ihre Kennzeichen wurden darin aus datenschutzrechtlichen Gründen anonymisiert. Laut diesem Bericht wurde die Veranstaltung mit Zeilennummer 1 von 67 Studenten in ihrem ersten, von 18 in ihrem zweiten und von 8 in ihrem dritten Fachsemester bestanden. Von insgesamt 204 Studenten haben sie demnach 93 Studenten bereits bestanden, wobei nur 115 die Veranstaltung bereits bestanden haben können, da 89 der 204 Studenten sich aktuell im ersten Fachsemester befinden. Dieses Modul oder Teilmodul wurde folglich größtenteils bestanden, zudem meist in frühen Fachsemestern. Der Bericht gibt erstens einen Endruck davon, in welchen Fachsemestern Veranstaltungen besonders häufig bzw. besonders selten bestanden werden, außerdem, wieviele der aktuell Studierenden die Veranstaltung bisher bestanden haben.

Der Bericht aus Tabelle 5.4 zeigt dagegen die Veranstaltungen aus Sicht der Studierenden in ihrem jeweils aktuellen Fachsemester. In Abbildung 5.3 wurde das anonymisierte Beispiel aus Abbildung 5.2 für diesen Bericht weitergeführt. Hier zeigt sich, dass die Veranstaltung in Zeile 1 von 70% der Drittsemestler, von 100% der Viertsemestler und von 90% der Studenten im fünften Fachsemester bisher bestanden worden ist. Die Erstgenannten haben einen Notendurchschnitt von 2,8, die Zweitgenannten von 3,1 und die Drittgenannten von 2,7. Auch in Abbildung 5.1 wird dieses Beispiel verwendet. Dort sieht man, dass aktuell keine Studierenden im zweiten Fachsemester sind, was der Grund dafür ist, dass dieses Fachsemester für alle Veranstaltungen nur leere Werte enthält und deshalb im Bericht nicht angezeigt wird. Mit seiner speziellen Sicht kann dieser Bericht z.B. schwache Kohorten aufzeigen.

In den Berichten sind Leistungen in Teilmodulen, deren Module bereits bestanden worden sind, nicht enthalten. Möglicherweise sind sie dennoch interessant, da es Module gibt, für die aus



Informatiklehrstuhl VI - Universitaet Wuerzburg — Dezember 11, 2009 @ 05:48

Lehrveranstaltungen in [redacted] nach kumulierten Fachsemestern (Nicht-Abbrecher)

			all fs	all fs	1	2	3	4	5
0			# alle	#1.FS	#	#	#	#	#
1	[redacted]	[redacted]	204	89	67	18	8		
2	[redacted]	[redacted]	204	89	4			1	
3	[redacted]	[redacted]	204	89	16	2	1		
4	[redacted]	[redacted]	204	89	13	4	1		
5	[redacted]	[redacted]	204	89				40	
6	[redacted]	[redacted]	204	89		68	14	5	
7	[redacted]	[redacted]	204	89		1			
8	[redacted]	[redacted]	204	89		23			
9	[redacted]	[redacted]	204	89	1	62	15	1	
10	[redacted]	[redacted]	204	89	75	10	10	4	
11	[redacted]	[redacted]	204	89			2		
12	[redacted]	[redacted]	204	89	7		1		
13	[redacted]	[redacted]	204	89	2	70	15	8	2
14	[redacted]	[redacted]	204	89	1		21	10	
15	[redacted]	[redacted]	204	89				2	
16	[redacted]	[redacted]	204	89		1		28	8
17	[redacted]	[redacted]	204	89	1	76	6	8	1
18	[redacted]	[redacted]	204	89			37	1	
19	[redacted]	[redacted]	204	89			1		
20	[redacted]	[redacted]	204	89			1		
21	[redacted]	[redacted]	204	89				39	1

Hinweise:

- Zeile 1: Fachsemester
- Zeile 2: Kennzahlen

- Spalte 1: Nummerierung
- Spalte 2: Modul/Teilmodul Name
- Spalte 3: Modul/Teilmodul Kuerzel (Legende beigelegt)

- Kennzahl # alle: Anzahl Studenten insgesamt
- Kennzahl #1.FS: Anzahl Studenten im 1. Fachsemester

Fuer jedes Fachsemester:
- Kennzahl #: Anzahl Studenten, die Leistung in diesem Fachsemester erbracht und bisher nicht abgebrochen haben.

Sortierung:
- Zuerst Module/Teilmodule "Pflicht...", alphabetisch sortiert
- Dann Module/Teilmodule "Nicht-Pflicht...", alphabetisch sortiert

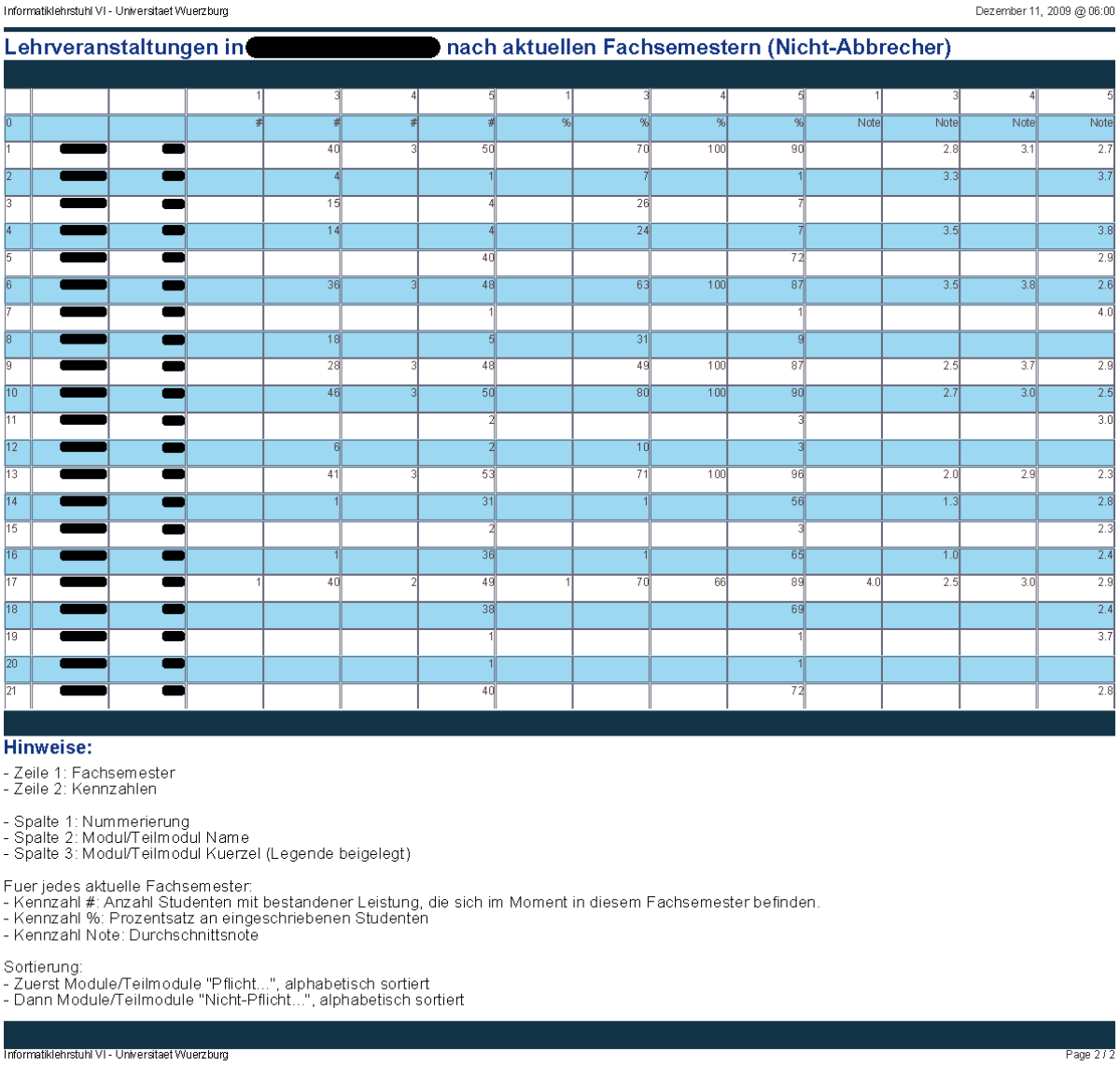
Informatiklehrstuhl VI - Universitaet Wuerzburg — Page 2 / 2

Abb. 5.2: Beispiel Lehrveranstaltungen nach kumulierten Fachsemestern

verschiedenen Teilmodulen ausgewählt werden kann.

Kritische Veranstaltungen untersuchen wir mit dem Bericht aus Tabelle 5.5. Abbildung 5.4 zeigt die Ausgabe dieses Berichts an einem anonymisierten Beispiel. Hier zeigt sich, dass Studierende, die das erste Modul bestanden haben, zu 10%, Studierende, die es nicht bestanden haben, zu 75% ihr Studium abgebrochen haben. Insgesamt haben 90 Studenten das Modul bisher bestanden. Tabellenzellen ohne Werte repräsentieren im Bericht den Wert „0“.

Dieser Bericht weist noch Einschränkungen auf. So fehlen zur intuitiven Interpretation die Zahlen derjenigen Abbrecher und Nicht-Abbrecher, die ein Modul nicht bestanden haben. Des Weiteren



Informatiklehrstuhl VI - Universitaet Wuerzburg — Dezember 11, 2009 @ 06:00

Lehrveranstaltungen in [redacted] nach aktuellen Fachsemestern (Nicht-Abbrecher)

			1	3	4	5	1	3	4	5	1	3	4	5
0			#	#	#	#	%	%	%	%	Note	Note	Note	Note
1	[redacted]	[redacted]		40	3	50		70	100	90		2.8	3.1	2.7
2	[redacted]	[redacted]		4		1		7		1		3.3		3.7
3	[redacted]	[redacted]		15		4		26		7				
4	[redacted]	[redacted]		14		4		24		7		3.5		3.8
5	[redacted]	[redacted]				40				72				2.9
6	[redacted]	[redacted]		36	3	48		63	100	87		3.5	3.8	2.6
7	[redacted]	[redacted]				1				1				4.0
8	[redacted]	[redacted]		18		5		31		9				
9	[redacted]	[redacted]		28	3	48		49	100	87		2.5	3.7	2.9
10	[redacted]	[redacted]		46	3	50		80	100	90		2.7	3.0	2.5
11	[redacted]	[redacted]				2				3				3.0
12	[redacted]	[redacted]		6		2		10		3				
13	[redacted]	[redacted]		41	3	53		71	100	96		2.0	2.9	2.3
14	[redacted]	[redacted]		1		31		1		56		1.3		2.8
15	[redacted]	[redacted]				2				3				2.3
16	[redacted]	[redacted]		1		36		1		65		1.0		2.4
17	[redacted]	[redacted]	1	40	2	49	1	70	66	89	4.0	2.5	3.0	2.9
18	[redacted]	[redacted]				38				69				2.4
19	[redacted]	[redacted]				1				1				3.7
20	[redacted]	[redacted]				1				1				
21	[redacted]	[redacted]				40				72				2.8

Hinweise:

- Zeile 1: Fachsemester
- Zeile 2: Kennzahlen

- Spalte 1: Nummerierung
- Spalte 2: Modul/Teilmodul Name
- Spalte 3: Modul/Teilmodul Kuerzel (Legende beigelegt)

Fuer jedes aktuelle Fachsemester:
- Kennzahl #: Anzahl Studenten mit bestandener Leistung, die sich im Moment in diesem Fachsemester befinden.
- Kennzahl %: Prozentsatz an eingeschriebenen Studenten
- Kennzahl Note: Durchschnittsnote

Sortierung:
- Zuerst Module/Teilmodule "Pflicht...", alphabetisch sortiert
- Dann Module/Teilmodule "Nicht-Pflicht...", alphabetisch sortiert

Informatiklehrstuhl VI - Universitaet Wuerzburg — Page 2 / 2

Abb. 5.3: Beispiel Lehrveranstaltungen nach aktuellen Fachsemestern

muss, wie bereits in Abschnitt 5.3.2.1 behandelt, auch für diesen Bericht die Unterscheidung zwischen Abbrechern, Nicht-Abbrechern und Absolventen sichergestellt sein, um auch für Studienfächer verwendet werden zu können, die bereits Absolventen aufweisen.

Wenn diese Einschränkungen aufgehoben werden, kann dieser Bericht den Dekanen aller Studienfächer dienen, um Hinweise auf kritische Veranstaltungen zu erhalten. Außerdem kann er – genauso wie der Bericht aus Tabelle 5.2 verwendet wird, um Abbrecher zu charakterisieren – verwendet werden, um Veranstaltungen zu charakterisieren. Der Bericht könnte dazu um Spalten erweitert werden, die weitere Eigenschaften und Kennzahlen zu den Veranstaltungen auflisten. Darunter zum Beispiel: die erwerbbaren ECTS-Punkte, die Anzahl Semesterwochen-

	Studenten Anzahl			Abbrecher von Bestanden	Abbrecher von Nicht-Bestanden
pruefung	All isAbbrechers	0	1	All isAbbrechers	All isAbbrechers
[redacted]	90	81	9	10,00%	75,00%
[redacted]	36	35	1	2,78%	66,67%
[redacted]	85	76	9	10,59%	50,00%
[redacted]	77	72	5	6,49%	28,57%
[redacted]	73	65	8	10,96%	28,00%
[redacted]	92	83	9	9,78%	20,00%
[redacted]	62	58	4	6,45%	10,34%
[redacted]	50	47	3	6,00%	8,33%
[redacted]	32	31	1	3,13%	,00%
[redacted]	25	25			
[redacted]	27	27			
[redacted]	41	41			

Abb. 5.4: Spalten: Anzahl Studenten mit bestandener Leistung; davon Anzahl Nicht-Abbrecher; davon Anzahl Abbrecher; davon Prozentsatz Abbrecher; Prozentsatz Abbrecher von Studenten ohne bestandene Leistung

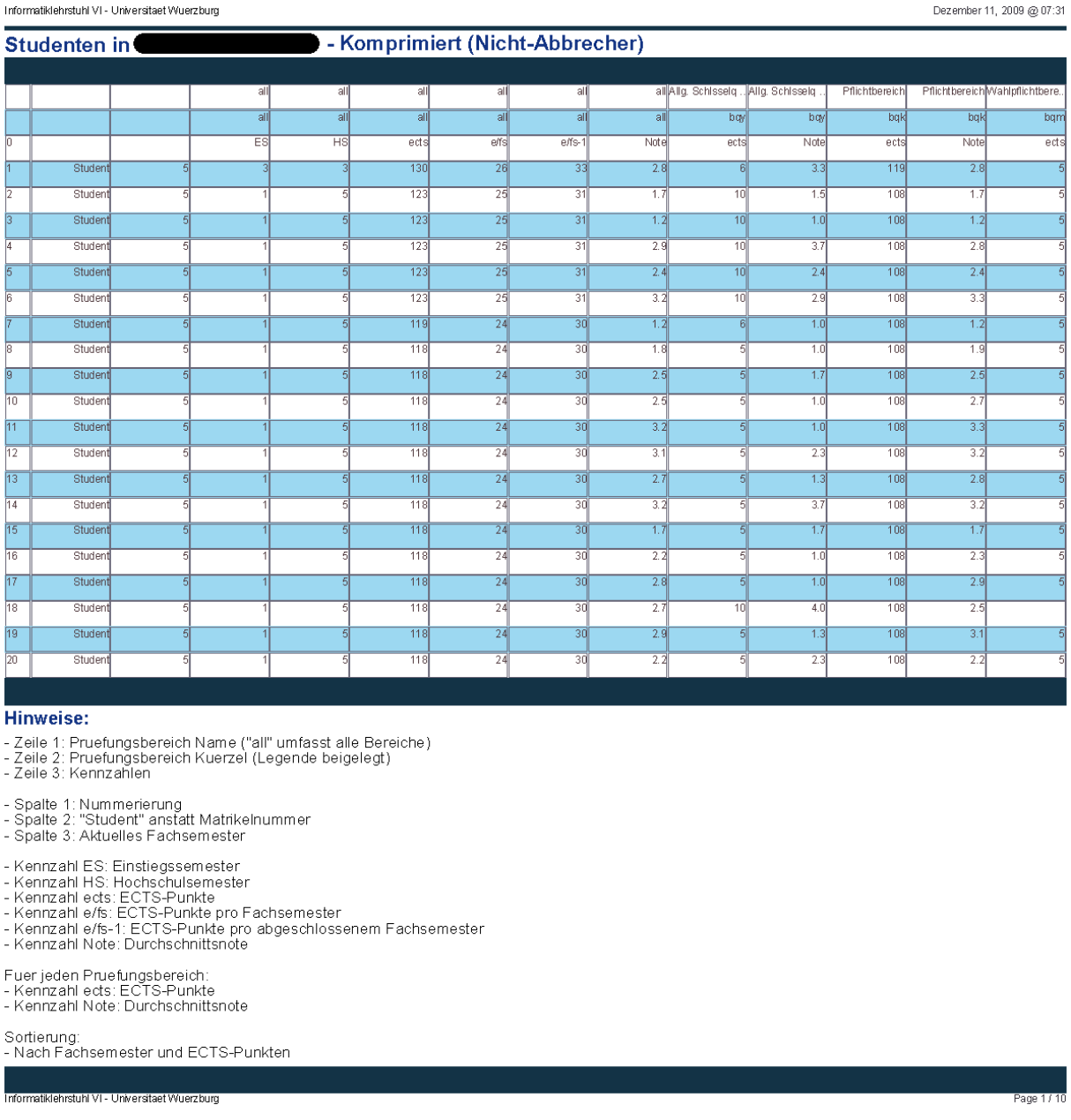
stunden, der Notendurchschnitt, die Nicht-Bestanden-Quote oder die durchschnittliche Anzahl an nötigen Wiederholungen. Kennzahlen können dabei auch nach Studienfächern, Fachsemestern oder einzelnen Kohorten aufgegliedert werden, um eine noch detailliertere Sicht zu bieten. Des Weiteren könnten auch konkrete Veranstaltungen eines Studiensemesters betrachtet werden. Diese lassen sich durch weitere Spalten im Bericht beschreiben, z.B. die erwartete, angemeldete und letztendlich zur Prüfung zugelassene Teilnehmerzahl. Erwartete Teilnehmer sind diejenigen aktuell Studierenden, die eine Pflichtveranstaltung noch nicht bestanden haben. In einem um Eigenschaften und Kennzahlen erweiterten Bericht könnten dann z.B. folgende Fragestellungen mittels der Data-Mining-Technik „Subgruppenentdeckung“ beantwortet werden:

- Welche Gründe können vorliegen, weshalb eine Veranstaltung häufig mit schlechten Noten, nach vielen Wiederholungen oder grundsätzlich selten bestanden wird?
- Welche Gründe können vorliegen, dass eine Veranstaltung von bestimmten Studierendengruppen selten bestanden wird, z.B. von Abbrechern, international Studierenden, weiblichen Studierenden?

5.3.2.3 Ergebnis: Analyse der Studierenden

Es ist unser Ziel, einen fundierten Einblick in die Studiensituation der Studierenden zu bieten, was wir in zwei Berichten umgesetzt haben. Der erste Bericht listet die wichtigsten Kennzahlen eines Studierenden auf. Laut anonymisiertem Beispiel in Abbildung 5.5 befindet sich der Student aus Zeile 1 aktuell in seinem fünften Fachsemester, ist mit dem dritten Fachsemester ins Studium eingestiegen – möglicherweise als Hochschulwechsler – und hat seit seinem Einstieg kein Urlaubssemester genommen, wie die Anzahl von drei Hochschulsemestern dieses Studienfachs verrät. Er

hat bereits 130 ECTS-Punkte erreicht, was 26 ECTS-Punkte im Schnitt pro Fachsemester und 33 ECTS-Punkte pro abgeschlossenem Fachsemester entspricht. Seine Durchschnittsnote beträgt 2,8. Seine ECTS-Punkte hat er in den Bereichen „Allg. Schlüsselqualifikation“ „Pflichtbereich“ und „Wahlpflichtbereich“ erhalten, Letzteres dabei ohne Note. Zu sehen sind auch die Kennzeichen der Prüfungsbereiche, z.B. „bqy“ für Allgemeine Schlüsselqualifikationen.



Informatiklehrstuhl VI - Universitaet Wuerzburg — Dezember 11, 2009 @ 07:31

Studenten in [redacted] - Komprimiert (Nicht-Abbrecher)

			all	all	all	all	all	all	Allg. Schlsselq ..	Allg. Schlsselq ..	Pflichtbereich	Pflichtbereich	Wahlpflichtbere..
			all	all	all	all	all	all	bqy	bqy	bqk	bqk	bqm
0			ES	HS	ects	e/fs	e/fs-1	Note	ects	Note	ects	Note	ects
1	Student	5	3	3	130	26	33	2.8	6	3.3	119	2.8	5
2	Student	5	1	5	123	25	31	1.7	10	1.5	108	1.7	5
3	Student	5	1	5	123	25	31	1.2	10	1.0	108	1.2	5
4	Student	5	1	5	123	25	31	2.9	10	3.7	108	2.8	5
5	Student	5	1	5	123	25	31	2.4	10	2.4	108	2.4	5
6	Student	5	1	5	123	25	31	3.2	10	2.9	108	3.3	5
7	Student	5	1	5	119	24	30	1.2	6	1.0	108	1.2	5
8	Student	5	1	5	118	24	30	1.8	5	1.0	108	1.9	5
9	Student	5	1	5	118	24	30	2.5	5	1.7	108	2.5	5
10	Student	5	1	5	118	24	30	2.5	5	1.0	108	2.7	5
11	Student	5	1	5	118	24	30	3.2	5	1.0	108	3.3	5
12	Student	5	1	5	118	24	30	3.1	5	2.3	108	3.2	5
13	Student	5	1	5	118	24	30	2.7	5	1.3	108	2.8	5
14	Student	5	1	5	118	24	30	3.2	5	3.7	108	3.2	5
15	Student	5	1	5	118	24	30	1.7	5	1.7	108	1.7	5
16	Student	5	1	5	118	24	30	2.2	5	1.0	108	2.3	5
17	Student	5	1	5	118	24	30	2.8	5	1.0	108	2.9	5
18	Student	5	1	5	118	24	30	2.7	10	4.0	108	2.5	
19	Student	5	1	5	118	24	30	2.9	5	1.3	108	3.1	5
20	Student	5	1	5	118	24	30	2.2	5	2.3	108	2.2	5

Hinweise:

- Zeile 1: Pruefungsbereich Name ("all" umfasst alle Bereiche)
- Zeile 2: Pruefungsbereich Kuerzel (Legende beigelegt)
- Zeile 3: Kennzahlen

- Spalte 1: Nummerierung
- Spalte 2: "Student" anstatt Matrikelnummer
- Spalte 3: Aktuelles Fachsemester

- Kennzahl ES: Einstiegssemester
- Kennzahl HS: Hochschulsemester
- Kennzahl ects: ECTS-Punkte
- Kennzahl e/fs: ECTS-Punkte pro Fachsemester
- Kennzahl e/fs-1: ECTS-Punkte pro abgeschlossenem Fachsemester
- Kennzahl Note: Durchschnittsnote

Fuer jeden Pruefungsbereich:
- Kennzahl ects: ECTS-Punkte
- Kennzahl Note: Durchschnittsnote

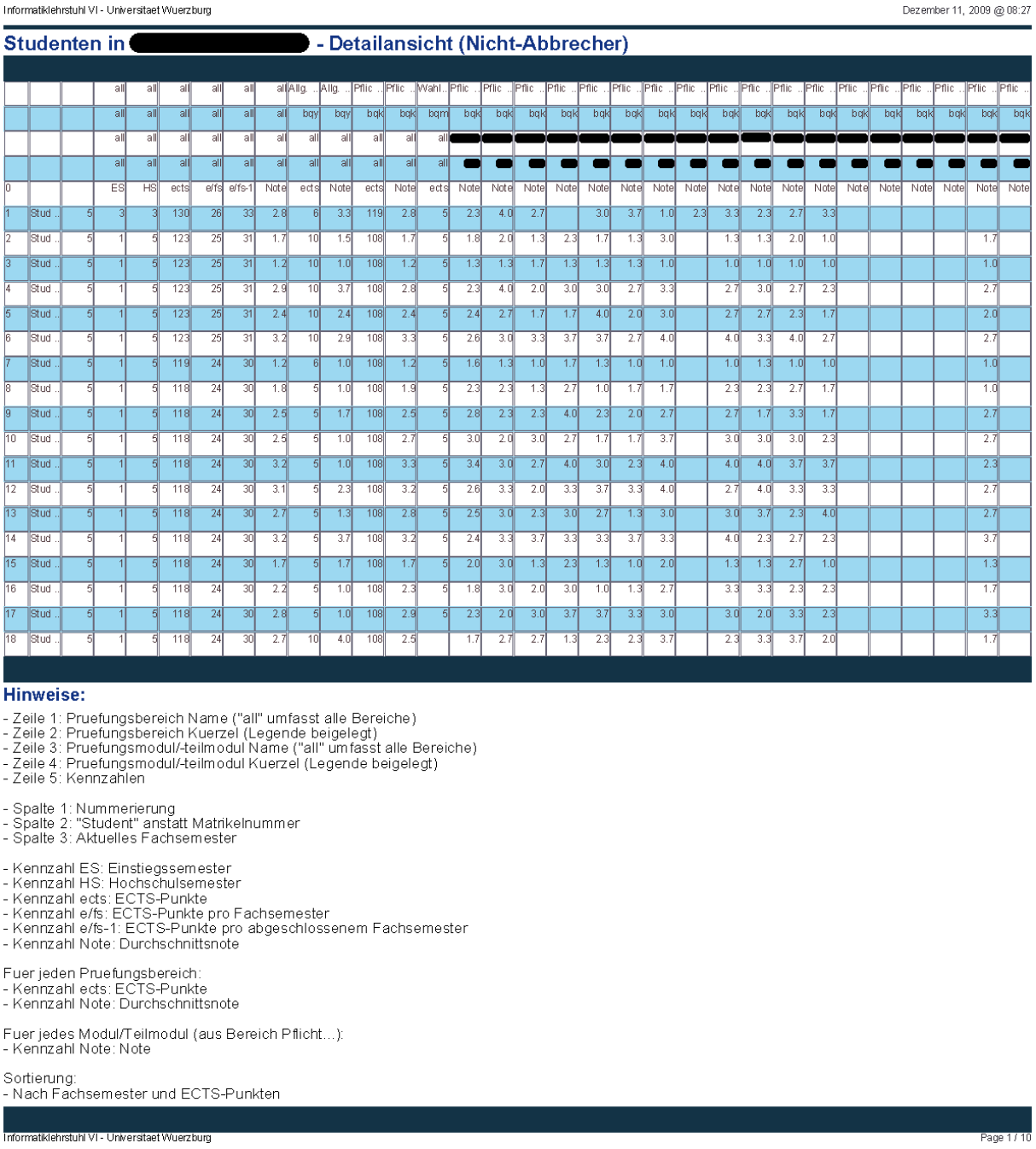
Sortierung:
- Nach Fachsemester und ECTS-Punkten

Informatiklehrstuhl VI - Universitaet Wuerzburg — Page 1 / 10

Abb. 5.5: Beispiel Bericht Studenten Komprimiert

Der zweite Bericht schlüsselt darüber hinaus den Pflichtbereich in Leistungen einzelner Module und Teilmodule auf. Im weitergeführten Beispiel werden – beginnend mit 2,3, endend mit

3,3 – die Noten angezeigt, die der Student in einzelnen, hier jedoch anonymisierten, Modulen und Teilmodulen erreicht hat. Auch die Kennzeichen der Veranstaltungen wurden unkenntlich gemacht.



Informatiklehrstuhl VI - Universitaet Wuerzburg — Dezember 11, 2009 @ 08:27

Studenten in [redacted] - Detailansicht (Nicht-Abbrecher)

			all	all	all	all	all	all	Allg. ..	Allg. ..	Pflic ..	Pflic ..	Wahl..	Pflic ..	Pflic ..	Pflic ..	Pflic ..	Pflic ..	Pflic ..	Pflic ..	Pflic ..	Pflic ..	Pflic ..	Pflic ..	Pflic ..	Pflic ..	Pflic ..	Pflic ..	Pflic ..	Pflic ..	Pflic ..
			all	all	all	all	all	all	bqy	bqy	bqk	bqk	bqm	bqk	bqk	bqk	bqk	bqk	bqk	bqk	bqk	bqk	bqk	bqk	bqk	bqk	bqk	bqk	bqk	bqk
			all	all	all	all	all	all	all	all	all	all	all	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
			all	all	all	all	all	all	all	all	all	all	all	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
0			ES	HS	ects	e/fs	e/fs-1	Note	ects	Note	ects	Note	ects	Note	Note	Note	Note	Note	Note	Note	Note	Note	Note	Note	Note	Note	Note	Note	Note	Note
1	Stud ..	5	3	3	130	26	33	2.8	6	3.3	119	2.8	5	2.3	4.0	2.7		3.0	3.7	1.0	2.3	3.3	2.3	2.7	3.3						
2	Stud ..	5	1	5	123	25	31	1.7	10	1.5	108	1.7	5	1.8	2.0	1.3	2.3	1.7	1.3	3.0		1.3	1.3	2.0	1.0					1.7	
3	Stud ..	5	1	5	123	25	31	1.2	10	1.0	108	1.2	5	1.3	1.3	1.7	1.3	1.3	1.3	1.0		1.0	1.0	1.0	1.0					1.0	
4	Stud ..	5	1	5	123	25	31	2.9	10	3.7	108	2.8	5	2.3	4.0	2.0	3.0	3.0	2.7	3.3		2.7	3.0	2.7	2.3					2.7	
5	Stud ..	5	1	5	123	25	31	2.4	10	2.4	108	2.4	5	2.4	2.7	1.7	1.7	4.0	2.0	3.0		2.7	2.7	2.3	1.7					2.0	
6	Stud ..	5	1	5	123	25	31	3.2	10	2.9	108	3.3	5	2.6	3.0	3.3	3.7	3.7	2.7	4.0		4.0	3.3	4.0	2.7					2.7	
7	Stud ..	5	1	5	119	24	30	1.2	6	1.0	108	1.2	5	1.6	1.3	1.0	1.7	1.3	1.0	1.0		1.0	1.3	1.0	1.0					1.0	
8	Stud ..	5	1	5	118	24	30	1.8	5	1.0	108	1.9	5	2.3	2.3	1.3	2.7	1.0	1.7	1.7		2.3	2.3	2.7	1.7					1.0	
9	Stud ..	5	1	5	118	24	30	2.5	5	1.7	108	2.5	5	2.8	2.3	2.3	4.0	2.3	2.0	2.7		2.7	1.7	3.3	1.7					2.7	
10	Stud ..	5	1	5	118	24	30	2.5	5	1.0	108	2.7	5	3.0	2.0	3.0	2.7	1.7	1.7	3.7		3.0	3.0	3.0	2.3					2.7	
11	Stud ..	5	1	5	118	24	30	3.2	5	1.0	108	3.3	5	3.4	3.0	2.7	4.0	3.0	2.3	4.0		4.0	4.0	3.7	3.7					2.3	
12	Stud ..	5	1	5	118	24	30	3.1	5	2.3	108	3.2	5	2.6	3.3	2.0	3.3	3.7	3.3	4.0		2.7	4.0	3.3	3.3					2.7	
13	Stud ..	5	1	5	118	24	30	2.7	5	1.3	108	2.8	5	2.5	3.0	2.3	3.0	2.7	1.3	3.0		3.0	3.7	2.3	4.0					2.7	
14	Stud ..	5	1	5	118	24	30	3.2	5	3.7	108	3.2	5	2.4	3.3	3.7	3.3	3.3	3.7	3.3		4.0	2.3	2.7	2.3					3.7	
15	Stud ..	5	1	5	118	24	30	1.7	5	1.7	108	1.7	5	2.0	3.0	1.3	2.3	1.3	1.0	2.0		1.3	1.3	2.7	1.0					1.3	
16	Stud ..	5	1	5	118	24	30	2.2	5	1.0	108	2.3	5	1.8	3.0	2.0	3.0	1.0	1.3	2.7		3.3	3.3	2.3	2.3					1.7	
17	Stud ..	5	1	5	118	24	30	2.8	5	1.0	108	2.9	5	2.3	2.0	3.0	3.7	3.7	3.3	3.0		3.0	2.0	3.3	2.3					3.3	
18	Stud ..	5	1	5	118	24	30	2.7	10	4.0	108	2.5		1.7	2.7	2.7	1.3	2.3	2.3	3.7		2.3	3.3	3.7	2.0					1.7	

Hinweise:

- Zeile 1: Pruefungsbereich Name ("all" umfasst alle Bereiche)
- Zeile 2: Pruefungsbereich Kuerzel (Legende beigelegt)
- Zeile 3: Pruefungsmodul/-teilmodul Name ("all" umfasst alle Bereiche)
- Zeile 4: Pruefungsmodul/-teilmodul Kuerzel (Legende beigelegt)
- Zeile 5: Kennzahlen

- Spalte 1: Nummerierung
- Spalte 2: "Student" anstatt Matrikelnummer
- Spalte 3: Aktuelles Fachsemester

- Kennzahl ES: Einstiegssemester
- Kennzahl HS: Hochschulsemester
- Kennzahl ects: ECTS-Punkte
- Kennzahl e/fs: ECTS-Punkte pro Fachsemester
- Kennzahl e/fs-1: ECTS-Punkte pro abgeschlossenem Fachsemester
- Kennzahl Note: Durchschnittsnote

Fuer jeden Pruefungsbereich:
- Kennzahl ects: ECTS-Punkte
- Kennzahl Note: Durchschnittsnote

Fuer jedes Modul/Teilmodul (aus Bereich Pflicht...):
- Kennzahl Note: Note

Sortierung:
- Nach Fachsemester und ECTS-Punkten

Informatiklehrstuhl VI - Universitaet Wuerzburg — Page 1 / 10

Abb. 5.6: Beispiel Bericht Studenten Detailliert

Für beide Berichte gibt es Vorschläge für Erweiterungen:

Worauf wir beim Ergebnis zu Tabelle 5.1 bereits eingegangen sind, muss hier auch berücksichtigt

werden: Studenten, die als einzige für ein aktuelles Fachsemester eingeschrieben sind, können leicht eindeutig identifiziert werden, weshalb deren Daten nicht in den Berichten enthalten sein dürfen. Anderenfalls können Dekane diese Berichte nicht weiterleiten, was für sie jedoch nötig ist, um Maßnahmen zu begründen.

Auch in diesen beiden Berichten sollte die Gesamtdurchschnittsnote, ähnlich wie im Ergebnis zu Tabelle 5.1 gewichtet errechnet werden.

Auch die Aufschlüsselung nach weiteren Bereichen kann interessant sein, wenn z.B. überprüft werden soll, welche Wahlfächer aus welchen Gründen (z.B. Anspruch, Noten) bevorzugt genommen werden. So könnte der zweite Bericht für Bereiche, die den Begriff „Schlüssel“ enthalten, angepasst werden. Ein weiterer Bericht würde die restlichen Veranstaltungen enthalten, es sei denn, eine weitere Aufteilung ist aus Platzgründen nötig.

Die Berichte können problemlos dazu verwendet werden, um auch spezielle Gruppen an Studierenden zu beschreiben. Ein solcher Bericht würde beispielsweise ausschließlich Abbrecher, Frühstudenten oder weibliche Studierende eines Studienfachs auflisten – vorausgesetzt allerdings, diese Studierende sind in den Daten dementsprechend gekennzeichnet. Manche Kennzahlen hätten in diesem Fall eine andere Bedeutung, z.B. bei Abbrechern wäre das aktuelle Fachsemester als das Abbruchsemester des Studenten zu verstehen.

In den Berichten können Studierende enthalten sein, die einen 2-Fach- oder 3-Fach-Bachelorabschluss anstreben. Für diese Abschlüsse gelten abweichende Bedingungen an die Anzahl zu erreichender ECTS-Punkte. Die Informationen zu solchen Studierenden können im Vergleich zu anderen Studierenden missinterpretiert werden, weshalb sie separat betrachtet werden sollten. Beispielsweise kann überlegt werden, ob solche Studienfachkominationen als eigene „Studienfächer“ in den gesamten Berichten enthalten sein sollten.

Ein ähnliches Problem stellen sog. „Frühstudenten“ dar. Sie erscheinen in den Daten erst dann, wenn sie ihren Hochschulzugang erreicht haben, und können sich häufig bereits in ihrem ersten Fachsemester, als regulär Studierende, überdurchschnittlich viele ECTS-Punkte aus dem Frühstudium anrechnen lassen. Daher verfälschen sie jegliche Statistiken und sollten getrennt von anderen Studierenden betrachtet werden.

Mit Abbildung 5.7 soll nun veranschaulicht werden, wie die ECTS-Punkteverteilung eines Studienfachs aussehen kann. Der Wert „All“ entspricht hier der Verteilung über alle Studenten, der Wert „0“ den Nicht-Abbrechern und der Wert „1“ den Abbrechern. In der Abbildung sind zwei Diagramme enthalten, die jeweils eine unterschiedliche Sicht auf die Verteilung bieten. Ein sog. Box-Plot zeigt, dass der Mittelwert der ECTS-Punkte pro Fachsemester bei den Abbrechern, wie zu erwarten, deutlich unter 20 ECTS-Punkten liegt. Aber auch die Nicht-Abbrecher erreichen nur knapp die kritische Grenze von 20 ECTS-Punkten, ein mögliches Zeichen dafür, dass noch einige Abbrüche bevorstehen. Im zweiten Diagramm, einer Verteilung, werden weitere Details

sichtbar, z.B., dass es einige Abbrecher gibt, die zwischen 10 und 20 ECTS-Punkte erreicht haben; bei solchen Studenten hätten sich möglicherweise Beratungsgespräche gelohnt. Einige überdurchschnittliche Nicht-Abbrecher könnten sich als Frühstudenten entpuppen.

Die Verteilung der ECTS-Punkte ist möglicherweise nicht nur für ein gesamtes Studienfach interessant, sondern auch für einzelne Studierendengruppen, z.B. Kohorten. Mit ihr kann empirisch eine ECTS-Punktegrenze gefunden werden, die in einem *Frühwarnsystem* verwendet wird, um gefährdete Studenten automatisch zu identifizieren.

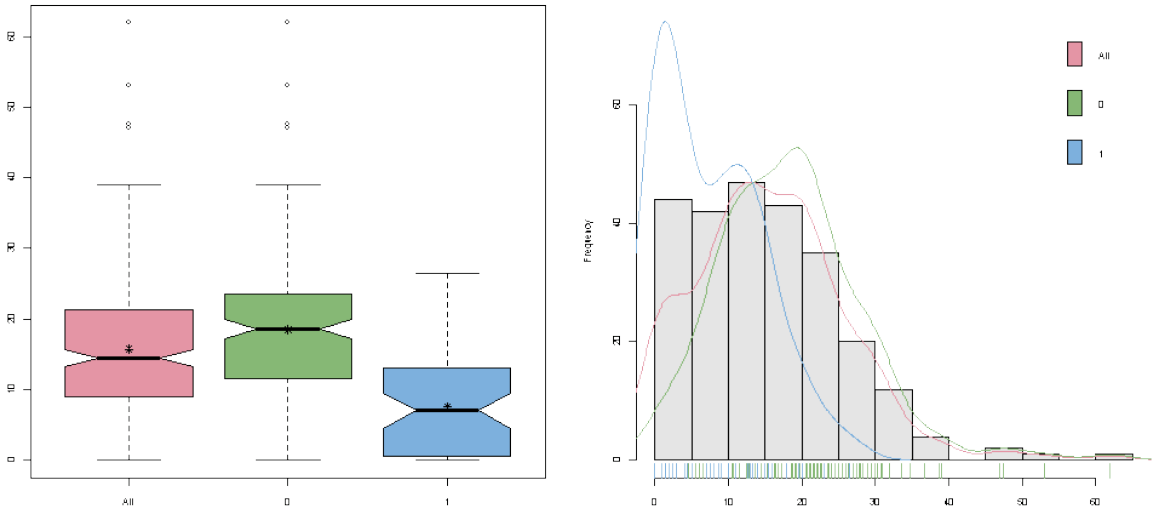


Abb. 5.7: Box-Plot und Verteilung: „All“ (gesamte Studenten), „0“ (Nicht-Abbrecher), „1“ (Abbrecher)

Die Analyse zu den durchschnittlichen ECTS pro Fachsemester enthält bisher allerdings zwei Einschränkungen: Sie betreffen einerseits die Berechnung der ECTS pro Fachsemester, andererseits die Unterscheidung von Abbrechern und Nicht-Abbrechern. Wir sind bereits im Ergebnis zu Tabelle 5.6 darauf eingegangen, dass die Berechnung der Anzahl an ECTS-Punkten pro Fachsemester verfälscht sein kann, wenn das aktuelle, unvollständige Semester einbezogen wird. Bei Abbrechern ist dies jedoch nicht der Fall, da sie bis zu ihrem Abbruch am Ende eines Semesters ECTS-Punkte erreicht haben können. Abbrecher und Nicht-Abbrecher sollten daher getrennt betrachtet werden. Des Weiteren sind wir bereits in Abschnitt 5.3.2.1 darauf eingegangen, dass wir in Berichten, in denen Abbrecher analysiert werden, bisher davon ausgehen, dass das untersuchte Studienfach keine Absolventen aufweist.

5.3.2.4 Ergebnis: Vergleich von Studiengängen

In Abbildung 5.8 wird ein exemplarischer Auszug des Berichts gezeigt, in dem ein Vergleich der Studiengänge möglich ist. Darin wurden die Studienfächer anonymisiert. Das erste Studienfach

haben bisher 222 Studenten studiert, die Studenten haben eine um 8,36% verminderte, also bessere Note als der Durchschnitt über alle Studienfächer. Die Abbrecherquote ist um 16,28% höher, die durchschnittliche ECTS-Punkteanzahl um 0,60% niedriger und die Wiederholungsanzahl um 1,55% niedriger als beim Durchschnitt.

	Kennzahlen				
Fach	Studenten Anzahl	Note Durchschnitt normiert	Abbrecher Prozentsatz normiert	ECTS-Summe pro Student und Studiensemester normiert	Wiederholungen Durchschnitt normiert
[redacted]	222	-8,36%	16,28%	-,60%	-1,55%
[redacted]	120	-13,79%	-5,71%	-8,85%	-1,95%
[redacted]	120	16,00%	32,60%	4,45%	7,67%
[redacted]	56	6,15%	-43,17%	5,00%	-4,17%

Abb. 5.8: Beispiel Vergleich Studiengänge

Eine Einschränkung dieses Berichts stellt die Berechnung der Wiederholungen dar. Zu jeder Leistung ist gespeichert, den wievielten Versuch des Studierenden zum Bestehen einer Prüfung sie darstellt. Über diese Zahl wird der Durchschnitt berechnet, der somit die Anzahl an Wiederholungen zum Bestehen einer Prüfung nur abschätzt. Wir empfehlen zum produktiven Gebrauch dieses Berichts, die Kennzahl unmissverständlich zu berechnen, was jedoch einen aufwändigeren Bericht erfordert. Des Weiteren wurden die Noten hier bisher ungewichtet berechnet. Für die durchschnittliche Anzahl an ECTS-Punkten pro Studiensemester bestehen noch ähnliche Einschränkungen, wie die im vorherigen Abschnitt 5.3.2.3 zum Ergebnis der Tabelle 5.8. Die Normierung bezieht sich auf die Gesamtheit an Leistungen aller Studierenden und nicht zusammengefasst für einzelne Studiengänge, wodurch Studienfächer mit vielen Studierenden stärker ins Gewicht fallen und die Kennzahlen missverstanden werden können.

Abbildung 5.9 zeigt, wie der Bericht aus Tabelle 5.10 verwendet werden kann, um Studiengänge anhand interdisziplinärer Veranstaltungen zu vergleichen. Hier wird ein bestimmtes Modul betrachtet, das von Studierenden aus vier Studienfächern (anonymisiert) besucht wird. Die Studierenden des ersten Studienfachs haben demnach einen Notendurchschnitt von 2,47 erhalten, der 4,25% schlechter als der Durchschnitt aller Studierenden ist. Die Anzahl an nötigen Wiederholungen ist um 4,54% niedriger als beim Durchschnitt der Leistungen über die vier Studienfächer.

	Kennzahlen			
Fach	Note Durchschnitt	Note Durchschnitt normiert	Wiederholungen Durchschnitt	Wiederholungen Durchschnitt normiert
[redacted]	247,345	4,25%	1,139	-4,54%
[redacted]	202,422	-14,68%	1,181	-,98%
[redacted]	299,259	26,13%	1,23	3,14%
[redacted]	200	-15,70%	1,221	2,38%

Abb. 5.9: Beispiel Interdisziplinäre Module

Dennoch darf die Aussage einer solchen Untersuchung nicht überbewertet werden: Ein überdurchschnittlich schlechtes bzw. gutes Abschneiden der Studierenden eines Studienfachs kann auch durch nicht betrachtete Faktoren herbeigerufen sein, z.B. einen regelmäßigen Wechsel des Dozenten oder die teils pflichtmäßige, teils wahlpflichtmäßige Integration der Veranstaltung

in den Verlaufsplan der Studierenden. Des Weiteren gelten die selben Einschränkungen wie auch für Bericht aus Tabelle 5.9, die zuvor in diesem Abschnitt genannt wurden.

5.4 Data-Assay

Im Folgenden wird beschrieben, wie Informationen zu den Daten erhalten wurden, um die Anforderungen aus dem Business-Case zu behandeln. Dieses Prüfen, Beschreiben und Vorbereiten der Daten im Data-Assay hat über das gesamte Projekt hinweg insgesamt ca. 60 Stunden unserer Zeit in Anspruch genommen.

5.4.1 Rohdatenbeschreibung ohne Vorverarbeitung

Zum Beginn des Projekts hat uns ein Administrator der Universität verschiedene Tabellen aus der Datenbank des Hochschulinformationssystems exportiert und jeweils in einer CSV-Datei gespeichert. Die Datenquellen liegen demnach bereits in Tabellarischer Form vor. Sie sollten Daten zu allen Bachelorstudiengängen der Universität enthalten. Von jeder der Dateien haben wir eine Kopie erstellt. Die CSV-Dateien trennen einzelne Werte mit Strichpunkt ohne sie durch Anführungszeichen zu umschließen und sind im verbreiteten ASCII-Format kodiert. Eine Kopfzeile nennt die Namen der Attribute. Tabelle 5.12 enthält weitere Informationen, die wir mit dem Texteditor *Notepad++* herausgefunden haben.

Name	**Attribute**	**Instanzen**	**Bemerkungen**
stg	9	12631	Stammblätter
sos	10	5025	Studentendaten
lab	18	108737	Prüfungsleistungen
labzuord	5	99051	Prüfungsleistungszuordnungstabelle
pord	15	23826	Prüfungsdaten
pnrzuord	4	29215	Prüfungszuordnungstabelle
k_abint	4	70	Schlüsseltabelle für Studienabschlüsse
k_stg	4	147	Schlüsseltabelle für Studienfächer
k_vert	4	358	Schlüsseltabelle für Vertiefungsfächer
k_akfz	6	256	Schlüsseltabelle für Staatszugehörigkeit
k_hzbart	3	56	Schlüsseltabelle für Hochschulzugangsberechtigung

Tab. 5.12: Beschreibung Bachelordaten

In eine sog. „Schlüsseltabelle“ sind Informationen aus einer anderen Tabelle ausgelagert bzw.

normalisiert. Ein Fremdschlüssel dieser anderen Tabelle korrespondiert zum Primärschlüssel der Schlüsseltabelle.

Daneben ließen sich der Name, der Datentyp sowie weitere Informationen zu den Attributen der Dateien feststellen, ohne Änderungen an den Rohdaten vorzunehmen.

5.4.2 Rohdatenbeschreibung mit Vorverarbeitung

Damit die Dateien mit weiteren Werkzeugen geöffnet werden konnten, haben wir sie um die Endung „.csv“, als Zeichen für CSV-Dateien, ergänzt. Wir konnten die Datei „pord“ zunächst mit keinem Data-Mining-Werkzeug einlesen, da sie von diesen nicht als korrekt strukturierte CSV-Datei erkannt wurde. Anscheinend waren beim Export Fehler aufgetreten, manuell hätten wir sie unter den 23826 Instanzen nicht finden können. Das Werkzeug *RapidMiner* hat beim Einlesen der Datei mit dem Operator „ExampleSource“ bei der Fehlersuche geholfen, indem es die Zeilen angab, in denen ein Problem mit der Struktur gefunden wurde; so wurde entdeckt, dass einzelne Strichpunkte und Zeilenumbrüche in den Prüfungsnamen der Grund waren. Diese haben wir manuell durch andere Zeichen ersetzt.

Das weitere Vorverarbeiten der Rohdaten hat sich darauf beschränkt, jede Datei in eine Datenbank „bachelor“ auf dem *Data-Warehouse-Server* einzulesen. Dazu haben wir einen *Job* mit *Pentaho Data Integration* erstellt. Dieser leert zunächst die zu füllenden Datenbanktabellen, damit sich neue Daten mit alten Daten nicht vermischen können und startet eine *Transformation*. Jede Datei wird darin mit einem „Text file input“-Knoten eingelesen. Leere Schlüsselattribute werden durch den Ersatzwert „mynull“ ersetzt, um nicht mit Missing-Values verwechselt zu werden. Einige kleinere Änderungen, wurden vorgenommen, z.B. das Trimmen derjenigen Attribute, die von Leerzeichen umschlossen werden. Jede Eingabe wird schließlich mit einem „Table output“-Knoten verbunden, der die Daten in die jeweilige Datenbanktabelle einliest.

Um Fragen an die Rohdaten effizient bearbeiten zu können, war es nötig, Primär- und Fremdschlüssel als solche in der Datenbank zu kennzeichnen, z.B. durch Indices. Dies haben wir mit *HeidiSQL* und *MySQL Workbench* getan. Anschließend waren auch komplexe SQL-Abfragen möglich, um Fragen an die Daten zu beantworten, die bei der Umsetzung der Anforderungen aus dem Business-Case aufgetreten sind.

5.4.3 Ergebnis der Rohdatenbeschreibung

Insgesamt haben wir folgende relevante Informationen zu den einzelnen Datenquellen festgestellt:

5.4.3.1 stg

Die Datei „stg“, siehe Tabelle 5.13, enthält die Daten der Stammblätter.

Name	Beschreibung	Datentyp	Bemerkungen
matnr	Matrikelnummer	String	Der Name des Attributs war in den Rohdaten mit verschlüsselt worden und wurde in matnr umbenannt.
abschl	Abschluss-ID	String	Fremdschlüssel zur Abschluss-ID aus k_abint; nur die Werte „82“ (Bachelor 1-Hauptfach), „B1“ (Zwei-Fach-Bachelor) und „B2“ (Zwei-Hauptfach-Bachelor).
stg	Studienfach-ID	String	Fremdschlüssel zu k_stg
vert	Vertiefung-ID	String	Fremdschlüssel zu k_vert; Immer leerer Wert also „keine Vertiefung“
kzfa	Haupt oder Nebenfach	String (1)	H (Hauptfach), N (Nebenfach)
pversion	Prüfungsordnungsversion	String	2006, 2007, 2008, 2009, Datentyp String, weil Fremdschlüssel zu Attribut „pversion“ in „pord“, welches auch nicht-numerische Werte enthält
semester	Studiensemester	Integer	von 20022 (Wintersemester 2002) bis 20092 (Wintersemester 2009)
stgsem	Fachsemester	Integer	von 0 bis 9

Tab. 5.13: Beschreibung stg

Mit SQL-Abfragen wurde weiterhin festgestellt:

- In der Datenquelle der Stammblätter ist kein einzelnes Attribut über alle Instanzen eindeutig. Vielmehr besitzt jeder Student pro Studiensemester maximal ein Stammblatt in einem Studiengang. Ein Studiengang wird durch die Attribute „abschl“, „stg“, „vert“, „kzfa“ und „pversion“ festgelegt, weshalb diese Attribute gemeinsam mit „matnr“ und „semester“ den Primärschlüssel darstellen.
- Es gibt Studierende, die ihr Studienfach gewechselt haben. Solche Studenten weisen zunächst Stammblätter mit einem Studienfach, in darauffolgenden Studiensemestern Stammblätter mit einem anderen Studienfach auf. Solche Studenten sollen als Abbrecher ihres vorherigen Fachs gelten.

- Es gibt Studierende, die während eines Studiensemesters mehrere Stammblätter mit verschiedenen Studienfächern besitzen, also mehrere Studienfächer gleichzeitig studieren.
- Es gibt Studierende, die über mehrere Studiensemester im gleichen Fachsemester eines Studiengangs verbleiben, was bei Urlaubssemestern der Fall ist.

5.4.3.2 sos

Die Datei „sos“, siehe Tabelle 5.14, enthält die Studentendaten, darunter Informationen zur Hochschulzugangsberechtigung (Hzb).

Name	**Beschreibung**	**Datentyp**	**Bemerkungen**
matnr	Matrikelnummer	String	Der Name des Attributs war in den Rohdaten mit verschlüsselt worden und wurde in matnr umbenannt; Primärschlüssel
gebdat	Geburtsdatum	Date	Format: yyyy-mm-dd
geschl	Geschlecht	String(1)	„M“ oder „W“
staat	ID der Staatsangehörigkeit	String	Fremdschlüssel für k_akfz
hzbart	ID der Hzb-Art	String	Fremdschlüssel für k_hzbart
hzbjahr	Jahr der Hzb	Integer	von 1968 bis 2009
hznote	Note	Integer	von 100 bis 990; mehrmals „None“; Integer, wenn „None“ in *null* umgewandelt wird
hssem	Aktuelles Hochschulsemester	Integer	von 1 bis 67

Tab. 5.14: Beschreibung sos

Für jeden Studenten, der ein Stammblatt besitzt, haben wir auch entsprechende Studentendaten, was sich durch eine Betrachtung der Fremdschlüssel mit SQL feststellen ließ.

5.4.3.3 lab

Die Datei „lab“, siehe Tabelle 5.15, enthält die Daten der Prüfungsleistungen.

Bei „pnote“ der Datenquelle lab ist uns aufgefallen, dass einige Notenwerte unverständlich waren. Deshalb wurde dieses Attribute mit „pstatus“ verglichen. Ergebnis: „000“ bei „PV“ und „BE“; „000“ bis „400“ bei „BE“; „500“ bei „NB“; „800“ bei „BE“; „900“ bei „EN“. „None“ kam bei mehreren Attributen vor. Später haben wir missverständliche Werte zu Missing-Values transformiert.

Name	Beschreibung	Datentyp	Bemerkungen
matnr	Matrikelnummer	String	Der Name des Attributs war in den Rohdaten mit verschlüsselt worden und wurde in matnr umbenannt.
labnr	Leistung-ID	String	Primärschlüssel
abschl	Abschluss-ID	String	Fremdschlüssel zu k_abint; meist Attributwert leer
stg	Studienfach-ID	String	Fremdschlüssel zu k_stg; meist Attributwert leer
vert	Vertiefung-ID	String	Fremdschlüssel zu k_vert; immer leerer Wert
kzfa	Haupt-/Nebenfach	String (1)	„H“ oder „N“
pversion	Prüfungsordnungsversion	String	„-1“, „0“ oder von 2006 bis 2009; Fremdschlüssel zu pversion von stg oder pord
pversuch	Nummer des Versuchs	Integer	von 1 bis 7
pstatus	Prüfungsstatus	String	„BE“, „NB“, „EN“, „PV“, „AN“
pnote	Prüfungsnote	Integer	Integer, wenn aus „None“ *null*; 3-stellig
pdatum	Datum der Prüfung	Date	Mehrmals „None“; Format: YYYY-MM-DD; Date, wenn aus „None“ *null* gemacht wird
psws	Semesterwochenstunden	Integer	von 0 bis 94
pordnr	Pordnummer	String	Fremdschlüssel zu pord
bonus	ECTS-Punkte	Integer	von 0 bis 90; teilweise „none“ in *null* umzuwandeln
psem	Prüfungssemester	String	von 20062 bis 20092

Tab. 5.15: Beschreibung lab

Wir haben festgestellt, dass die Versuchsanzahl bei Prüfungen mit Prüfungsart „MO“ immer dem Wert „1“ entspricht, also keine Information enthält. Über die Tabelle „labzuord“ sind die meisten Leistungen in einem Modul mit Leistungen aus Teilmodulen mit Prüfungsart „TM“ verbunden. Bei diesen entspricht „pversuch“ tatsächlich der Versuchsanzahl. Daher haben wir entschieden, anstelle der Leistung des Moduls die Leistungen in den Teilmodulen für die Anforderungen an den Tabellen 5.9 und 5.10 herzunehmen. Bei Modulleistungen, für die es keine Teilmodulleistung gibt, wurde die Information aus der Modulleistung hergenommen, was jedoch nur in Ausnahmefällen auftrat.

Es gibt Studierende, die während eines Studiensemesters, in dem sie mit einem Stammblatt in einen Studiengang eingeschrieben sind, keine einzige Leistung erbringen. Andersherum gibt es auch Leistungen, die von einem Studierenden während eines Studiensemesters absolviert werden, ohne mit einem Stammblatt immatrikuliert zu sein. Später haben wir für jede Leistung dennoch ein zugehöriges Stammblatt bestimmt.

Bei „pversion“ von „lab“ ist aufgefallen, dass die Werte „-1“ und „0“ keine Schlüsselattributswerte, sondern Ersatzwerte darstellen. Solche Leistungen können z.B. übergreifend für mehrere Prüfungsordnungsversionen gelten.

5.4.3.4 labzuord

Die Datei „labzuord“, siehe Tabelle 5.16, beschreibt eine Hierarchie der Leistungen. In einer sog. „Parent-Child-Hierarchie“ werden darin Leistungen anderen Leistungen zugeordnet, um Beziehungen zwischen Teilmodulen, Modulen und Prüfungsbereichen zu beschreiben. Um die Informationen dieser Datei zu extrahieren haben wir mit dem Pentaho-Data-Integration-Step „Closure Generator“ eine sog. „Closure Table“ erstellt. In dieser ist jedes Paar an Leistungen enthalten, zwischen denen es in der Hierarchie einen Weg gibt. In der Closure Table haben wir herausgefunden, dass jedes Modul ein oder mehrere zugehörige Teilmodule besitzt, sowie in Prüfungsbereiche eingeteilt ist.

Name	**Beschreibung**	**Datentyp**	**Bemerkungen**
matnr	Matrikelnummer	String	Der Name des Attributs war in den Rohdaten mit verschlüsselt worden und wurde in matnr umbenannt
labnr	Leistung-ID (Kind)	String	Fremdschlüssel zu lab
artzuordnung	Art der Zuordnung	String	z.B. „F00A“, „FK“
pordnrzu	Prüfung-ID(Vater)	String	Fremdschlüssel zu pord
labnrzu	Leistung-ID (Vater)	String	Fremdschlüssel zu lab

Tab. 5.16: Beschreibung labzuord

5.4.3.5 pord

Die Datei „pord“, siehe Tabelle 5.17, enthält die Daten der Prüfungsdaten.

Die Tabelle „pnrzuord“ enthält für uns redundante Informationen aus „labzuord“, war für die Analyse nicht notwendig und wird daher nicht weiter beschrieben.

Name	Beschreibung	Datentyp	Bemerkungen
abschl	Abschluss-ID	String	Fremdschlüssel zu k_abint
stg	Studienfach-ID	String	Fremdschlüssel zu k_stg
vert	Vertiefung-ID	String	Fremdschlüssel zu k_vert
kzfa	Haupt/Nebenfach	String	z.B. Fremdschlüssel zu stg
pversion	Prüfungsordnungsversion	String	Fremdschlüssel zu stg
pdtxt	Drucktext der Pruefung	String	-
pltxt1	Langtext der Pruefung	String	-
pltxt3	Langtext der Pruefung	String	meist „None“
part	Prüfungsart	String	z.B. „YB“, „MO“, „TM“
bonus	ECTS-Punkte	Real	meist 0.0, „None“; Dezimalzeichen ist der Punkt.
pordnr	Prüfungsordnungs-ID	String	Primärschlüssel
sws	Semesterwochenstunden	Real	0.0, „None“; Dezimalzeichen ist der Punkt

Tab. 5.17: Beschreibung pord

5.4.3.6 k_abint

Die Datei „k_abint“, siehe Tabelle 5.18, ist eine Schlüsseltabelle, in der Daten zu Abschlüssen abgefragt werden.

Name	Beschreibung	Datentyp	Bemerkungen
abint	AbschlussID	String	Primärschlüssel, einmal leerer Wert
ktxt	Kurztext	String	-
dtxt	Durchschnittstext	String	-
ltxt	Langtext	String	-

Tab. 5.18: Beschreibung k_abint

5.4.3.7 k_stg

Die Datei „k_stg“, siehe Tabelle 5.19, ist eine Schlüsseltabelle, in der Daten zu Studienfächern abgefragt werden.

Name	Beschreibung	Datentyp	Bemerkungen
stg	Studienfach-ID	String	Primärschlüssel
ktxt	Kurztext	String	-
dtxt	Durchschnittstext	String	-
ltxt	Langtext	String	-

Tab. 5.19: Beschreibung k_stg

5.4.3.8 k_vert

Die Datei „k_vert“, siehe Tabelle 5.20, ist eine Schlüsseltabelle, in der Daten zu Vertiefungsfächern abgefragt werden.

Name	Beschreibung	Datentyp	Bemerkungen
vert	VertiefungsID	String	Primärschlüssel
ktxt	Kurztext	String	-
dtxt	Durchschnittstext	String	-
ltxt	Langtext	String	-

Tab. 5.20: Beschreibung k_vert

5.4.3.9 k_akfz

Die Datei „k_akfz“, siehe Tabelle 5.21, ist eine Schlüsseltabelle, in der Daten zur Herkunft abgefragt werden.

Name	Beschreibung	Datentyp	Bemerkungen
akfz	Vertiefung-ID	String	Primärschlüssel
ltxt	Herkunftsangabe	String	-
erdteil	Erdteil	String	z.B. „- - -“, „ASI“, „EUR“
iso_code	ISO-Code des Landes	String(2)	30 Einträge mit leerem Wert

Tab. 5.21: Beschreibung k_akfz

5.4.3.10 k_hzbart

Die Datei „k_hzbart“, siehe Tabelle 5.22, ist eine Schlüsseltabelle, in der Daten zur Hochschulzugangsberechtigung (Hzb) abgefragt werden.

Name	Beschreibung	Datentyp	Bemerkungen
hzbart	Hzb-Art-ID	String	Primärschlüssel
ltxt	Beschreibung	String	-

Tab. 5.22: Beschreibung k_hzbart

5.5 Data-Warehouse

Nun sind die Daten also in einer Datenbank vorhanden und zur Weiterverarbeitung vorbereitet. Ein weiteres Ziel bestand darin, mit Unterstützung der Informationen aus dem Data-Assay, die Daten in fest-definierte Strukturen zu bringen, um eine Erstellung der Berichte über den Rechner des *Data-Warehouse* zu ermöglichen. Die Konzeption und Umsetzung solcher Strukturen in einem Data-Warehouse hat insgesamt ca. 50 Stunden benötigt.

Wir haben in diesem Projekt die Werkzeuge *MySQL Server*, *Mondrian OLAP Server* und *BI-Server* sowie einige Verwaltungswerkzeuge, z.B. *MySQL Workbench* und *Pentaho Schema Workbench* verwendet.

5.5.1 ER-Modell

Aus den bisher erhaltenen Informationen haben wir ein *Entity-Relationship-Modell* entwickelt, das die Objekte und Attribute enthält, die in den Anforderungen aus dem Business-Case verlangt sind.

In Abbildung 5.10 wird das ER-Modells als *UML-Klassendiagramm* dargestellt.

Im Folgenden werden die enthaltenen Objekte beschrieben. Es werden insbesondere abgeleitete Attribute erklärt; nicht-abgeleitete Attribute haben wir direkt aus bestehenden Attributen übernommen und wurden bereits zu einem früheren Zeitpunkt beschrieben.

5.5.1.1 Stammblatt

Dieses Objekt beschreibt alle vorhandenen Stammblätter und ihre Eigenschaften.

Die Abgeleiteten Attribute erklären sich wie folgt:

First_Studiensemester entspricht dem Studiensemester, in dem sich der Student des Stammblatts als erstes im Studiengang des Stammblatts befand. Dies entspricht auch der *Kohorte*, der der Student zugeordnet werden kann. Dieses Abgeleitete Attribut, genauso wie die nächsten drei Attribute, wurde durch eine SQL-Abfrage mit „Group-By“-Operator auf die Stammblätter des Studenten erstellt.

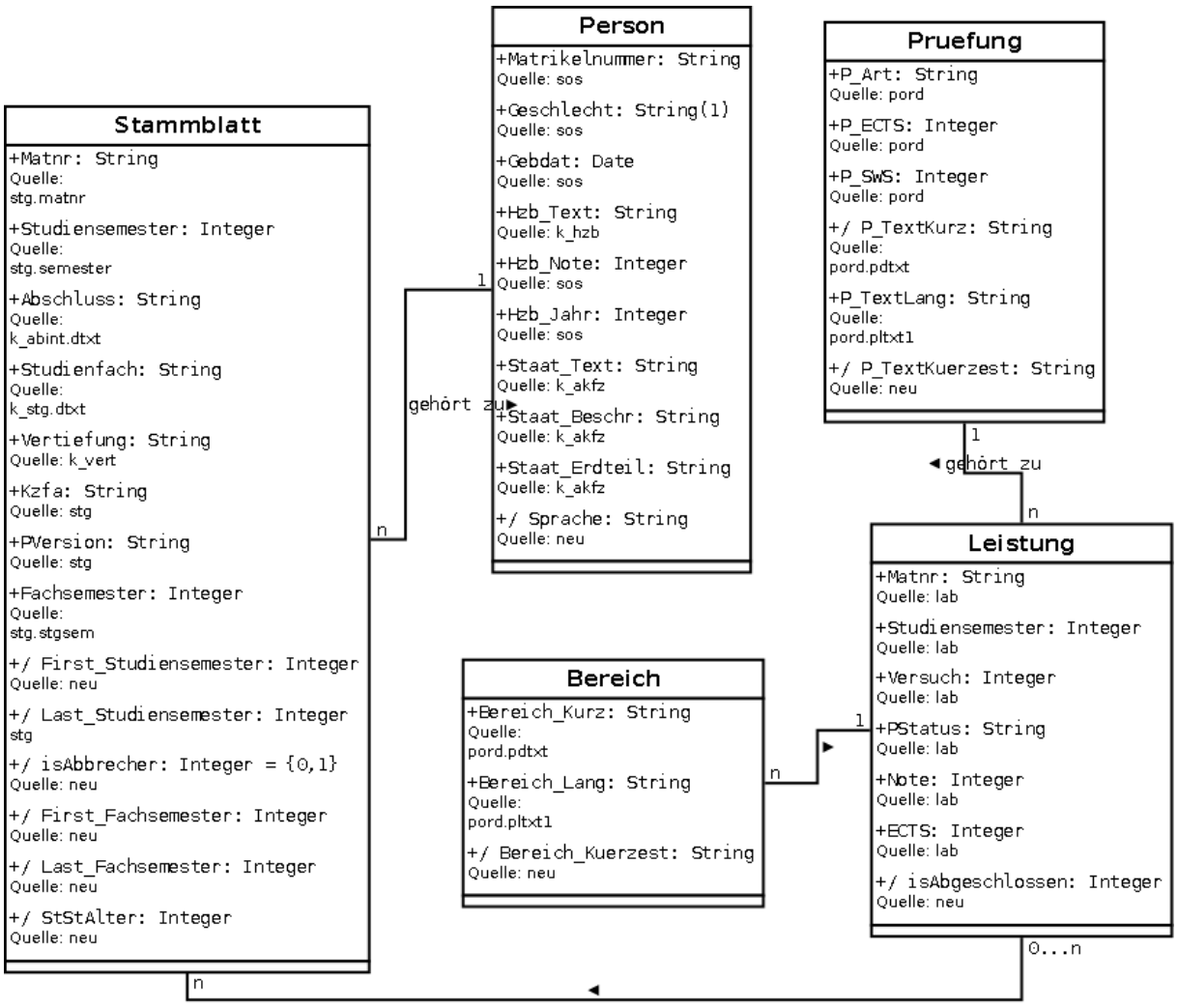


Abb. 5.10: Bachelor ER-Modell

Last_Studiensemester entspricht dem Studiensemester, in dem sich der Student bisher als letztes in dem Studiengang befand.

First_Fachsemester entspricht dem Fachsemester, in dem sich der Student des Stammblatts als erstes im Studiengang des Stammblatts befand.

Last_Fachsemester entspricht dem Fachsemester, in dem sich der Student bisher als letztes in dem Studiengang befand. Diese Angabe zum aktuellen Fachsemester eines Studenten werden wir für Anforderung aus Tabelle 5.4 benötigen.

isAbbrecher entspricht der Information darüber, ob dieser Student im aktuell erfassten Studiensemester noch mit einem Stammblatt vertreten ist. Ein Abbrecher besitzt den Wert „1“, ein Nicht-Abbrecher den Wert „0“. Solange es noch keine Bachelorabsolventen gibt, unterscheidet dieses Attribut korrekt zwischen Abbrechern und Nicht-Abbrechern. Diese

Eigenschaft benötigen wir für die Tabellen 5.2, 5.5, 5.8 und 5.9. Die aktuell Studierenden werden durch dieses Attribut korrekt gekennzeichnet, was für die Tabellen 5.1, 5.3, 5.4, 5.6 und 5.7 nötig ist.

StStAlter entspricht dem Alter, das der Student in seinem ersten Studiensemester des Studienganges hatte. Es wird aus „Gebdat“ und „First_Studiensemester“ berechnet. Als Stichtag wird für jedes Studiensemester das Datum des Vorlesungsbeginns verwendet, welches manuell für die vergangenen Studiensemester in Pentaho Data Integration eingegeben wurde. Dieses Attribut benötigen wir für die Tabelle 5.2.

Studiensemester und Fachsemester, aber auch die von ihnen abgeleiteten Attribute wie First_Studiensemester, werden als Integer gespeichert.

Jedes Stammblatt kann über die Matrikelnummer einer bestimmten Person zugeordnet werden.

5.5.1.2 Person

Als Personen werden alle vorhandenen Studenten mit ihren Stammdaten bezeichnet.

Der Wert „None“ wird für „Hzb_Note“ und „Hzb_Jahr“ in *null* umgewandelt, damit diese Attribute als Integer gespeichert werden können.

Eine Person besitzt das abgeleitete Attribut „Sprache“. Dieses besitzt den Wert „DE“, wenn diejenige Person aus einem deutschsprachigen Land – also „Staat_Text“ z.B. dem Wert „D“ entspricht – stammt, den Wert „Non-DE“, falls nicht. So werden die Sprachkenntnisse abgeschätzt, wie es laut Tabelle 5.2 verlangt ist. Jede Person besitzt mindestens ein Stammblatt.

5.5.1.3 Leistung

In diesem Objekt werden die Leistungen aus Modulen und Teilmodulen zusammengefasst.

Es besitzt das abgeleitete Attribut „isAbgeschlossen“. Dieses Attribut beschreibt, ob für eine Leistung in einem Teilmodul bereits eine Leistung im zugehörigen Modul bestanden worden ist. Diese Leistungen dürfen bei den ECTS- und Notenangaben der Anforderungen aus Tabellen 5.1, 5.6 und 5.7 nicht einbezogen werden. In der Datei „labzuord“ wird jeder Leistung in einem Teilmodul eine Leistung in einem Modul zugeordnet. Daraus lässt sich die zugehörige Modulleistung eines Teilmoduls feststellen und überprüfen, ob „pstatus“ dem Wert „BE“, also bestanden, entspricht.

Jeder Leistung sind ein oder mehrere Bereiche und Stammblätter sowie eine Prüfung zugeordnet.

Die zugehörigen Stammblätter ergeben sich durch die Bereiche: So endet in der Datei „labzuord“ die Hierarchie der Leistungen von jeder Leistung aus einem Modul oder Teilmodul in Leistungen

aus einem oder mehreren Prüfungsbereichen. Ein solcher Bereich ist mit dem Wert „YB“ des Attributs „part“ gekennzeichnet und speichert Informationen zum Studiengang. Mittels dieser Information lassen sich für jede Leistung ein oder mehrere Stammblätter finden – und zwar über den Fremdschlüssel aus „Matnr“ „Studiensemster“, „Abschluss“, „Studienfach“, „Vertiefung“, „Kzfa“ und „Pversion“. Für den Fall, dass eine Leistung in einem Semester erbracht wurde, in dem der Student nicht eingeschrieben war, wird die Leistung dem oder den passenden Stammblättern zugeordnet, bei denen er sich in seinem ersten Studiensemester befand.

Diese komplexe Behandlung der Leistungen, Bereiche und Stammblätter ist insbesondere bei Studenten relevant, die zeitgleich mehrere Studienfächer studieren und den Abschluss „Zwei-Fach-Bachelor“ oder „Zwei-Hauptfach-Bachelor“ aufweisen. Deren Leistungen in fachübergreifenden Veranstaltungen können so mehreren Stammblättern und Studiengängen zugeordnet werden.

Für das Attribut „Note“ haben wir Noten, die nicht zwischen „000“ und „400“ liegen, in *null* umgewandelt, damit Summen und Durchschnitte später richtig berechnet werden. Und auch den Wert „None“ haben wir für „ECTS“ in *null* umgewandelt. Des Weiteren haben wir „Studiensemester“ und „Versuch“ als Integer gespeichert.

5.5.1.4 Bereich

Wie im vorherigen Abschnitt erklärt, können jeder Leistung in einem Modul oder Teilmodul Leistungen in Prüfungsbereichen zugeordnet werden. Diese Prüfungsbereiche werden im Objekt „Bereich“ zusammengefasst. Die Attribute „Bereich_Kurz“ und „Bereich_Lang“ geben jedem Bereich Kurz- und Langnamen und werden aus den Prüfungsdaten gelesen.

Wir haben den Prüfungsbereich für die Tabellen 5.6, 5.7, 5.3 und 5.4 benötigt.

Jeder Bereich benötigt laut dieser Anforderungen auch ein Kennzeichen. Dieses Kennzeichen besteht aus einer 26-Buchstaben-Kodierung von Zahlen mit allen Kleinbuchstaben des Alphabets, die wir in JavaScript programmiert haben, siehe Quellcode 5.1, und die durch den Kettle-Step „Modified Java Script Value“ ausgeführt wird. Die Kennzeichen ließen sich später aus der Datenbank in eine CSV-Datei exportieren und den Berichten als Legende für die Entscheidungsträger beilegen.

```
1 var letterArray = new Array('a','b','c','d','e','f','g','h','i','j','k','l','m','n',
2 'o','p','q','r','s','t','u','v','w','x','y','z');
3 var letterNumber = 26;
4
5 var zahl = number;
6 var rest = 0;
7 var ist = zahl;
8 var arrayNumber = 0;
9 var idArray = new Array();
```

```
10
11 while (ist != 0) {
12   rest = zahl % letterNumber;
13   ist = Math.floor(zahl/letterNumber);
14   idArray[arrayNumber] = rest;
15   arrayNumber++;
16   zahl = ist;
17 }
18
19 var id = "";
20 var idArray_Reverse = idArray.reverse();
21
22 for(var i = 0; i < arrayNumber; i++) {
23   id = id + letterArray[idArray_Reverse[i]];
24 }
```

Quellcode 5.1: Progammierung Kennzeichen

5.5.1.5 Prüfung

Auch die Namen von Prüfungen sollten, z.B. laut Tabelle 5.3, durch ein Kennzeichen abgekürzt werden, was wir gleichzeitig mit der Abkürzung der Bereiche getan haben. Auch haben wir hier die Abkürzung der Kurznamen vorgenommen, wie sie in dieser Anforderung beschrieben wird.

Wir haben eine Transformation mit *Pentaho Data Integration* erstellt, die für jedes Objekt im Entity-Relationship-Modell und seinen Attributen eine Tabelle erstellt. Mittels *HeidiSQL* und *MySQL Workbench* haben wir die Primär- und Fremdschlüssel eingestellt, um eine performante Abfrage und Weiterverarbeitung des ER-Modells zu ermöglichen.

Des Weiteren haben wir die Attributwerte „None“ für „P_ECTS“ und „P_SWS“ in *null* umgewandelt, damit diese Werte als Integer in der Datenbank gespeichert und später interpretiert werden können.

5.5.2 MD-Modell

Keinen der Berichte aus dem Business-Case haben wir direkt über das ER-Modell erstellt, zu aufwändig wären die SQL-Abfragen gewesen; stattdessen haben wir es hauptsächlich als Informationsquelle zum Erstellen eines Multidimensionalen Modells verwendet. Mit diesem werden letztendlich die Inhalte für die Berichte abgefragt. Wir haben zwei *Data-Cubes*, einen mit Sicht auf Stammblätter, einen mit Sicht auf Leistungen, modelliert. Die Data-Cubes werden in einem Klassendiagramm, siehe Abbildung 5.11, dargestellt.

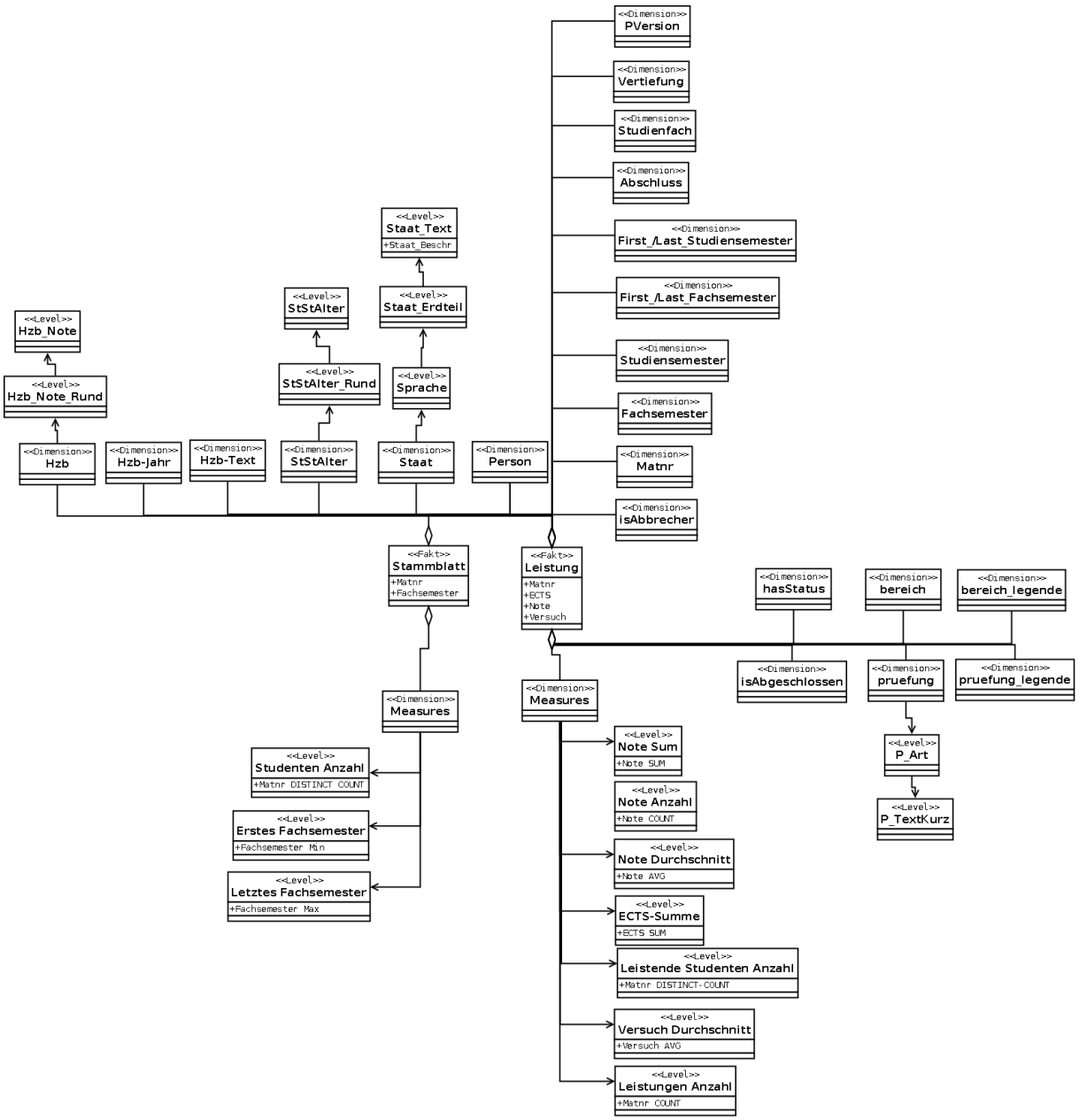


Abb. 5.11: Bachelor MD-Modell

5.5.2.1 Stammblatt

Jedes Stammblatt besitzt als Kennzahlattribute die Matrikelnummer und das Fachsemester des Stammblatts. Von Ersterem wird in Kennzahl „Studenten Anzahl“ die Anzahl unterschiedlicher Matrikelnummern, also Studenten, berechnet. Von „Fachsemester“ wird das minimale Fachsemester, „Erstes Fachsemester“, und das maximale Fachsemester, „Letztes Fachsemester“, berechnet.

Das Fakt Stammblatt besitzt ausschließlich gemeinsame Dimensionen, die in einem späteren Abschnitt beschrieben werden.

5.5.2.2 Leistung

Jede Leistung besitzt folgende Kennzahlen:

Note Sum summiert die Noten der Leistungen auf.

Note Anzahl zählt die vergebenen Noten. Noten mit Wert *null* werden nicht mitgezählt.

Note Durchschnitt berechnet den Durchschnitt der Noten. Noten mit Wert *null* fließen nicht in den Durchschnitt ein.

ECTS-Summe summiert die ECTS-Punkte auf.

Leistende Studenten Anzahl zählt die Anzahl unterschiedlicher Matrikelnummern und damit die Anzahl an Studenten, die eine Leistung erbracht haben.

Versuch Durchschnitt errechnet den Durchschnitt der Versuchszahl, also den durchschnittlichen Versuch eines Studierenden, um eine Prüfung zu bestehen. Diese Kennzahl werden wir verwenden, um die durchschnittliche Anzahl an Wiederholungen von Leistungen abzuschätzen, wie sie in den Tabellen 5.9 und 5.10 benötigt wird.

Leistungen Anzahl zählt die Anzahl an Leistungen.

Aus den Attributen „pstatus“ und „isAbgeschlossen“ werden Dimensionen direkt, ohne Hierarchie, erstellt.

Eine Leistung, die mehreren Bereichen zugeordnet werden kann, wird für den Data-Cube verdoppelt, so dass jede Leistung einen eigenen Bereich besitzt. Dieser wird durch zwei Dimensionen ausgedrückt, einmal durch „bereich“ mit dem Attribut „Bereich_Kurz“, und einmal durch „bereich_legende“ mit dem Attribut „Bereich_Kuerzest“.

Auch die Prüfung, in der die Leistung absolviert wird, wird in zwei Dimensionen ausgedrückt, einmal ausgeschrieben und einmal mit Kennzeichen. Ersteres wird noch durch eine vorherige Hierarchiestufe „P_Art“ erweitert, das die Prüfungen in Module und Teilmodule trennt.

5.5.2.3 Gemeinsame Dimensionen

Die Data-Cubes teilen sich folgende *Gemeinsame Dimensionen*, über die sich Kennzahlen zu mehreren Data-Cubes gleichzeitig abfragen lassen.

Jedem Stammblatt und jeder Leistung können die Attribute „isAbbrecher“, „Matnr“, „Fachsemester“, „Studiensemester“, „First_Fachsemester“, „Last_Fachsemester“,

„First_Studiensemester“, „Last_Studiensemester“, „Abschluss“, „Studienfach“, „Vertiefung“, „Pversion“, „Hzb_Text“ und „Hzb_Jahr“ nach dem ER-Modell zugeordnet und als direkte Dimensionen verwendet werden. Die Dimension „Person“ teilt die Stammblätter oder Leistungen nach dem Geschlecht der Studierenden auf. Die Attribute „Hzb_Note“ und „StStAlter“ erhalten noch eine Hierarchiestufe hinzu, in der ihre Werte diskretisiert werden. Die Diskretisierungen hätte man auch bereits für das ER-Modell einführen können, dies wurde hier nicht gemacht, da sie in den Anforderungen erst später hinzugekommen sind. Die Dimension „Staat“ besteht aus einer Hierarchie mit „Sprache“, „Staat_Erdteil“ und „Staat_Text“.

Auch für das MD-Modell haben wir mehrere Transformationen mit *Pentaho Data Integration* erstellt, die es in ein Datenbankmodell umsetzen. Das Vorgehen besteht aus der Abfrage aller Eigenschaften der Stammblätter und Leistungen aus dem ER-Modell, wie es in Quellcode 5.2 am Beispiel der Leistung gezeigt wird. Dort werden zunächst alle Leistungen ausgegeben, die von Studenten während ihrer Immatrikulation geleistet wurden, anschließend Leistungen, die außerhalb der Immatrikulation geleistet wurden; für diese werden Stammblätter aus dem ersten Studiensemester verwendet.

```
1  select ER_Stammblatt.*, ER_Person.*, ER_Leistung.*, ER_Pruefung.*
2  , ER_Bereich.*, "Student" as Matnr_Ersatz
3  from ER_Leistung
4  left join ER_Person on ER_Leistung.Matnr = ER_Person.Matnr
5  left join ER_Pruefung on ER_Leistung.Pordnr = ER_Pruefung.Pordnr
6  left join ER_Bereich on ER_Leistung.Labnr = ER_Bereich.labnr
7  left join ER_Stammblatt on ER_Bereich.matnr_key = ER_Stammblatt.matnr_key
8  and ER_Bereich.psem_key = ER_Stammblatt.semester_key
9  and ER_Bereich.abschl_key = ER_Stammblatt.abschl_key
10 and ER_Bereich.stg_key = ER_Stammblatt.stg_key
11 and ER_Bereich.vert_key = ER_Stammblatt.vert_key
12 and ER_Bereich.kzfa_key = ER_Stammblatt.kzfa_key
13 and (ER_Bereich.pversion_key = ER_Stammblatt.pversion_key
14 or ER_Leistung.pversion_key = 0 or ER_Leistung.pversion_key = -1)
15 where ER_Stammblatt.Matnr is not null
16
17 union
18
19 select Stamm2.*, ER_Person.*, ER_Leistung.*, ER_Pruefung.*
20 , ER_Bereich.*, "Student" as Matnr_Ersatz
21 from ER_Leistung
22 left join ER_Person on ER_Leistung.Matnr = ER_Person.Matnr
23 left join ER_Pruefung on ER_Leistung.Pordnr = ER_Pruefung.Pordnr
24 left join ER_Bereich on ER_Leistung.Labnr = ER_Bereich.labnr
25 left join ER_Stammblatt on ER_Bereich.matnr_key = ER_Stammblatt.matnr_key
26 and ER_Bereich.psem_key = ER_Stammblatt.semester_key
27 and ER_Bereich.abschl_key = ER_Stammblatt.abschl_key
28 and ER_Bereich.stg_key = ER_Stammblatt.stg_key
29 and ER_Bereich.vert_key = ER_Stammblatt.vert_key
30 and ER_Bereich.kzfa_key = ER_Stammblatt.kzfa_key
31 and (ER_Bereich.pversion_key = ER_Stammblatt.pversion_key or ER_Leistung.pversion_key = 0
32 or ER_Leistung.pversion_key = -1)
```

```
33 left join ER_Stammblatt as Stamm2 on ER_Bereich.matnr_key = Stamm2.matnr_key
34 and ER_Bereich.abschl_key = Stamm2.abschl_key and ER_Bereich.stg_key = Stamm2.stg_key
35 and ER_Bereich.vert_key = Stamm2.vert_key and ER_Bereich.kzfa_key = Stamm2.kzfa_key
36 and (ER_Bereich.pversion_key = Stamm2.pversion_key or ER_Leistung.pversion_key = 0
37 or ER_Leistung.pversion_key = -1) and Stamm2.Studiensemester =
       Stamm2.First_Studiensemester
38 where ER_Stammblatt.Matnr is null and Stamm2.Matnr is not null
```

Quellcode 5.2: Erstellung Data-Cube Leistung

Die Dimensionstabellen werden dann mit dem Step „Dimension lookup/update“ erstellt und gefüllt, die Faktentabellen mit einfachen „Table output“-Steps. Zur leichteren Verwaltung hat nicht jede Dimension eine eigene Dimensionstabelle erhalten, z.B. wurden „Person“, „Hzb_Jahr“, „Hzb_Text“, „Hzb_Note“ und „Staat“ in einer Tabelle gespeichert. Mittels *HeidiSQL* und *MySQL Workbench* haben wir die Primär- und Fremdschlüssel eingestellt, um eine performante Abfrage des MD-Modells zu ermöglichen.

5.6 Reporting

Die Berichte wurden entweder mit *Pentaho Report Designer* und *Pentaho Design Studio* oder direkt über den *Pentaho BI-Server* erstellt. Dies hat uns insgesamt 40 Stunden in Anspruch genommen.

5.6.1 Übersicht - Eingeschriebene Studenten mit Note und ECTS-Punkten

Die Schwierigkeit, diesen Bericht herzustellen, bestand darin, den kumulierten ECTS- und Notendurchschnitt mittels abgeleiteter Kennzahlen zu berechnen.

Den kumulierten ECTS-Durchschnitt in der Kennzahl „ects“ haben wir dabei zunächst durch einen Ausdruck in Prädikatenlogik dargestellt. Studiensemester und Fachsemester werden hier verkürzt „ss“ und „fs“ geschrieben. Levels werden nicht ausdrücklich genannt, wenn ihre Dimension keine Hierarchie besitzt. Die Data-Cubes enthalten durch den „slice“-Operator nur Daten eines bestimmten Studienfachs. Zunächst wird die Kennzahl „ECTS Student“ definiert; sie gibt für jede Zelle im Data-Cube „Leistung“, die durch das Fachsemester, das Studiensemester und die Matrikelnummer bestimmt ist, die ECTS-Summe eines Studierenden an, die er bis dahin erreicht hat. Daraus wird dann „ECTS“ berechnet, das den Durchschnitt der vorherigen Kennzahl über alle Studierende berechnet, die in diesem Studiensemester und Fachsemester eingeschrieben sind:

$$\forall c \in [fs, ss, matnr] \in Leistung, c.ECTSStudent =$$
$$sum\{d.ECTS - Summe | d \in [fs, ss, matnr] \in Leistung,$$
$$d.fs <= c.fs \wedge d.matnr = c.matnr \wedge d.status = BE\}$$

$$\forall f \in [fs, ss] \in Stammblatt,$$
$$f.ECTS = avg\{g.ECTSStudent | g \in [fs, ss, matnr] \in Leistung,$$
$$f.fs = g.fs \wedge f.ss = g.ss \wedge \exists e | e \in [fs, ss, matnr] \in Stammblatt,$$
$$e.fs = g.fs \wedge e.ss = g.ss \wedge e.Matnr = g.Matnr \wedge e.StudentenAnzahl = 1\}$$

Diesen haben wir anschließend in AW-RA dargestellt. Hier werden zunächst die abgeleiteten Kennzahlen „ECTS Student“ (ecStud) und „ECTS Student gesamt“ (ecStGe) berechnet. Ersteres gibt für einen Studenten in einem Fach- und Studiensemester die Anzahl an ECTS-Punkten aus bestandenen Leistungen an. Die zweite Kennzahl errechnet für einen Studierenden die gesamten ECTS-Punkte. Diese werden in Kennzahl „ECTS“ für eingeschriebene Studenten (AnzStud = 1) aufsummiert und durch die Anzahl an Studenten geteilt (AnzStudGe). Die Kennzahlen „AnzStud“ und „AnzStudGe“ zählen lediglich die Anzahl unterschiedlicher Matrikelnummern auf zwei Granularitätsstufen und werden daher nicht beschrieben. Man beachte die Hilfstabellen „Base1“ und „Base2“, die die Zellen enthalten, für die die Aggregationen durchgeführt werden. Man beachte, dass hier Kennzahlen aus zwei Data-Cubes verwendet werden, Stammblatt und Leistung. Diese Möglichkeit wurde im Werk von Chen et al. [19] nicht beschrieben, ließ sich jedoch in AW-RA realisieren:

$$ecStud =$$
$$g_{(fs,ss,matnr),sum(ECTS-Summe)}\sigma_{Status=BE}(Leistung)$$

$$Base1 =$$
$$g_{(fs,ss,matnr),0}(Stammblatt)$$

$$Base2 =$$
$$g_{(fs,ss),0}(Stammblatt)$$

$$ecStGe =$$

$$Base1 \bowtie_{ecStud.fs<=Base1.fs \wedge ecStud.matnr=Base1.matnr, sum(ecStud.M)} ecStud$$

$$ECTS = Base2$$

$$\bowtie_{ecStGe.fs=Base2.fs \wedge ecStGe.ss=Base2.ss \wedge ecStGe.AnzStud=1, sum(ecStGe.M)/Base2.AnzStudGe.M} ecStGe$$

Sowohl die Erstellung des Ausdrucks in Prädikatenlogik, als auch in AW-RA haben zum Verständnis beigetragen. Weitaus intuitiver und schneller war jedoch die Erstellung von *Aggregation Workflows*, um die Abfrage zu konzeptionieren, siehe Abbildung 5.12. Das Diagramm ist von unten nach oben aufgebaut, errechnet wird zuerst „ECTS Student“, daraus dann „ECTS Student gesamt“ und letztendlich „ECTS“.

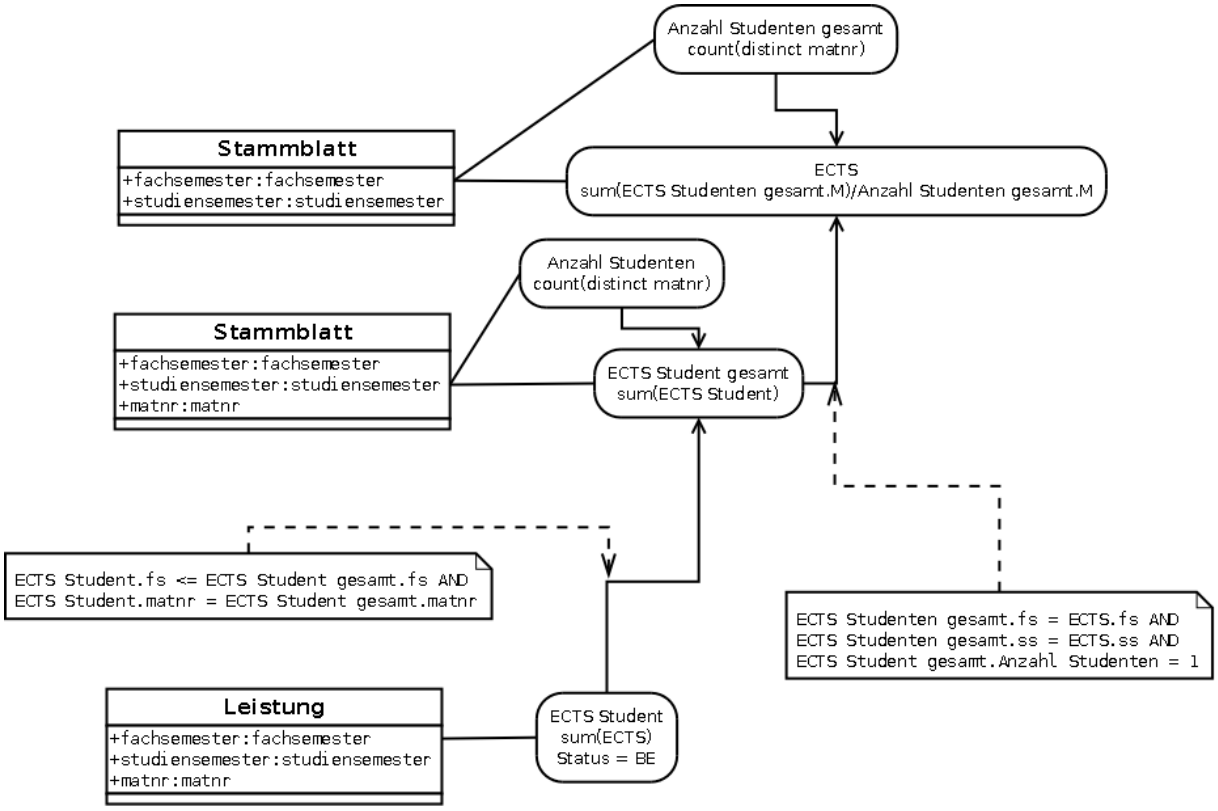


Abb. 5.12: Aggregation Workflows kumulierte ECTS

Auch die kumulierten Durchschnittsnoten, die zudem nach ECTS-Punkten gewichtet werden sollten, wurden durch Aggregation Workflows konzeptioniert, siehe Abbildung 5.13. Auch dieses Diagramm ist von unten nach oben aufgebaut. „Note gewichtet“ ist die mittels ECTS-Punkte

gewichtete Notensumme aus bestandenen Leistungen; diese wird in Kennzahl „Note Student gewichtet“ durch die Gesamtsumme an ECTS-Punkten aus bestandenen Leistungen mit einer Note geteilt, um die gewichtete Note zu erhalten. Für eingeschriebene Studenten wird anschließend in „Note Student gewichtet“ der Durchschnitt dieser gewichteten Noten berechnet.

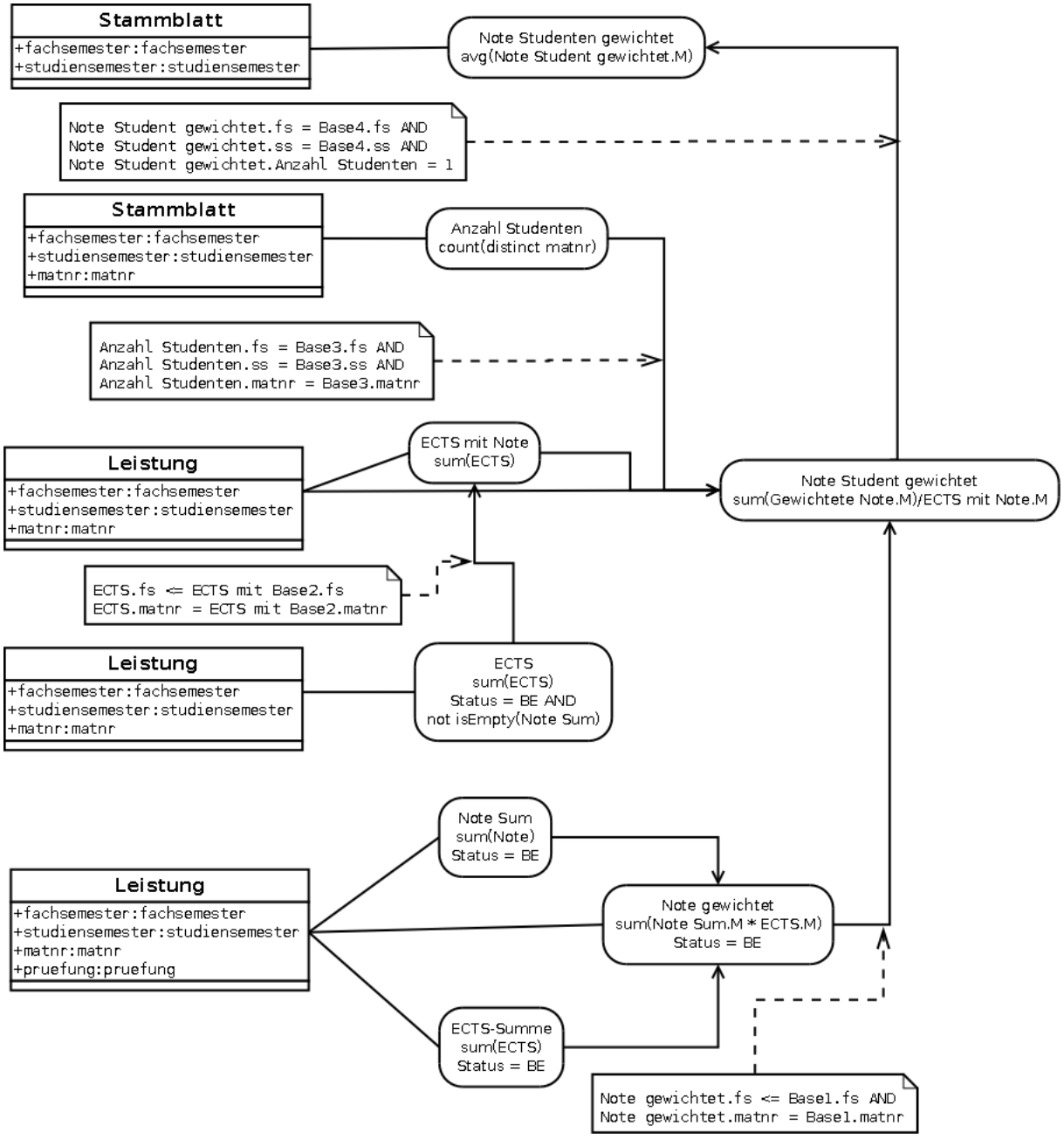


Abb. 5.13: Aggregation Workflows gewichtete Note

Quellcode 5.3 zeigt die MDX-Abfrage. Man beachte, dass in dieser Abfrage die Kennzahlen zweier Data-Cubes verwendet werden: „Studenten Anzahl“ des Data-Cube Stammblatt und die Kenn-

zahlen „Note Sum“ und „ECTS-Summe“ aus Data-Cube Leistung. Für diese Art von Abfragen haben wir beide Data-Cubes mit *Mondrian Schema Workbench* in einem *Virtuellen Data-Cube* „uebersichtstudent“ zusammengefasst. Auch spätere Abfragen nehmen von diesem Gebrauch. Man beachte auch, dass in dieser MDX-Abfrage kurze Namen für Kennzahlen verwendet werden, damit jede Spalte eine möglichst geringe Breite benötigt. Außerdem werden Kennzahlen in ein geeignetes Format gebracht, z.B. werden die Noten als einstellige Zahl mit einer Stelle hinter dem Komma angezeigt. Es werden auch nicht die vollständigen Data-Cubes betrachtet, sondern z.B. nur Leistungen, die nicht abgeschlossen sind, „isAbgeschlossen“ also den Wert „0“ besitzt.

Der Ausdruck „{studienfach}“ ist nicht MDX-spezifisch, sondern ein Parameter, der in *Pentaho Design Studio* verwendet wird. Vor der Ausführung dieses Berichts wird nach dem Studienfach gefragt, zu dem der Bericht ausgeführt werden soll. Über eine Programmablaufschleife in Pentaho Design Studio konnten wir unter Verwendung solcher Parameter automatisch für jedes Studienfach einen Auszug des Berichts erstellen lassen. Dazu haben wir eine Berichtsvorlage erstellt, in das die Parameter sowie die Ausgabe einer MDX-Abfrage eingegeben werden. In Abhängigkeit dieser Eingaben hat sich die Berichtsvorlage mit einer dynamischen Anzahl an Zeilen und Spalten gefüllt. Für diese Lösung war es nötig, Standardfunktionen von Pentaho Design Studio durch eigene Programmierung in JavaScript zu erweitern (für den Quellcode, siehe Dokumentation im Anhang). Auch die Berichte aus den Tabellen 5.3, 5.4, 5.6 und 5.7 haben wir derart umgesetzt, von denen sich genauso, ohne manuellem Zusatzaufwand, Auszüge für alle Studienfächer erstellen lassen.

```
1  with member [Measures].[#] as
2  'Int(([isAbgeschlossen].[All isAbgeschlossens]
3  , [hasStatus].[All hasStatuss], [Measures].[Studenten Anzahl]))'
4
5  member [Measures].[Note] as
6  'IIf(IsEmpty([Measures].[#]), NULL, Round(([Measures].[Note Studenten gewichtet] /
       100.0), 1.0))'
7
8  member [Measures].[Note Studenten gewichtet] as
9  'Avg(Filter([matnr].[Matnr].Members, (NOT IsEmpty([Measures].[#]))))
10 , [Measures].[Note Student gewichtet])'
11
12 member [Measures].[Note Student gewichtet] as
13 '(Sum(LastPeriods(100.0, [fachsemester].CurrentMember)
14 , ([studiensemester].[All studiensemesters], [Measures].[Note Sum gewichtet]))
15 / Sum(LastPeriods(100.0, [fachsemester].CurrentMember)
16 , ([studiensemester].[All studiensemesters], [Measures].[ECTS mit Note])))'
17
18 member [Measures].[ECTS mit Note] as
19 'Sum(Filter([pruefung].[Pordnr].Members
20 , (NOT IsEmpty([Measures].[Note Sum]))), [Measures].[ECTS-Summe])'
21
22 member [Measures].[Note Sum gewichtet] as
23 'Sum([pruefung].[Pordnr].Members, [Measures].[Note Leistung gewichtet])'
24
```

```
25 member [Measures].[Note Leistung gewichtet] as
26 '([Measures].[Note Sum].Value * [Measures].[ECTS-Summe].Value)'
27
28 member [Measures].[ects richtig kumuliert] as
29 'Sum(Crossjoin(Crossjoin(Crossjoin({[studiensemester].[All studiensemesters]}
30 , LastPeriods(100.0, [fachsemester].CurrentMember))
31 , Filter([matnr].[Matnr].Members, (NOT IsEmpty([Measures].[#]))))
32 , {[Measures].[ECTS-Summe]}), [Measures].[ECTS-Summe])'
33
34 member [Measures].[ects] as
35 'IIf(IsEmpty([Measures].[ects richtig kumuliert])
36 , NULL, Int(Round(([Measures].[ects richtig kumuliert] / [Measures].[#]), 0.0)))'
37
38 select NON EMPTY
39 Crossjoin([studiensemester].[All studiensemesters].Children
40 , {[Measures].[#], [Measures].[ects], [Measures].[Note]}) ON COLUMNS,
41
42 NON EMPTY [fachsemester].[All fachsemesters].Children ON ROWS
43 from [uebersichtstudent]
44 where ([fach].[All fachs].[{studienfach}], [hasStatus].[All hasStatuss].[BE]
45 , [isAbgeschlossen].[All isAbgeschlossens].[0])
```

Quellcode 5.3: MDX-Ausdruck Übersicht - Eingeschriebene Studenten

Dieser Bericht konnte direkt in der Business-Story verwendet werden und wird dort exemplarisch in Abbildung 5.1 gezeigt.

5.6.2 Abbrecher Eigenschaften

Für diesen Bericht wurden die Dimensionen „Matnr“, „First_Studiensemester“, „StStAlter“ (auf der diskretisierten Hierarchiestufe), „Hzb_Text“, „Hzb_Note“ (auf der diskretisierten Hierarchiestufe), „Sprache“, „isAbbrecher“ und „Person“ zur Beschreibung der Zeilen verwendet. Die Spaltendimensionen der Abfrage bestehen aus „pruefung“ (auf der Hierarchiestufe „P_Text_Kurz“) sowie „studiensemester“; für die erste Gruppe an Spalten wird die Kennzahl „Bisher Bestanden“, für die zweite Gruppe an Spalten die Kennzahl „ECTS-Summe pro SS Diskretisiert“ verwendet.

Die erste Kennzahl gibt den Wert „1“ aus, wenn unter einer Menge an Leistungen eine bestandene Leistung enthalten ist. Diese Kennzahl wird verwendet, um für einen Studenten und ein Modul anzuzeigen, ob er es bereits bestanden hat und gibt ansonsten den Wert *null* aus. Die Definition der Kennzahl zeigt Quellcode 5.4.

```
1 with member [Measures].[Bisher Bestanden] as
2 'IIf((NOT IsEmpty(([hasStatus].[All hasStatuss].[BE], [Measures].[Studenten Anzahl]))),
      1.0, NULL)'
```

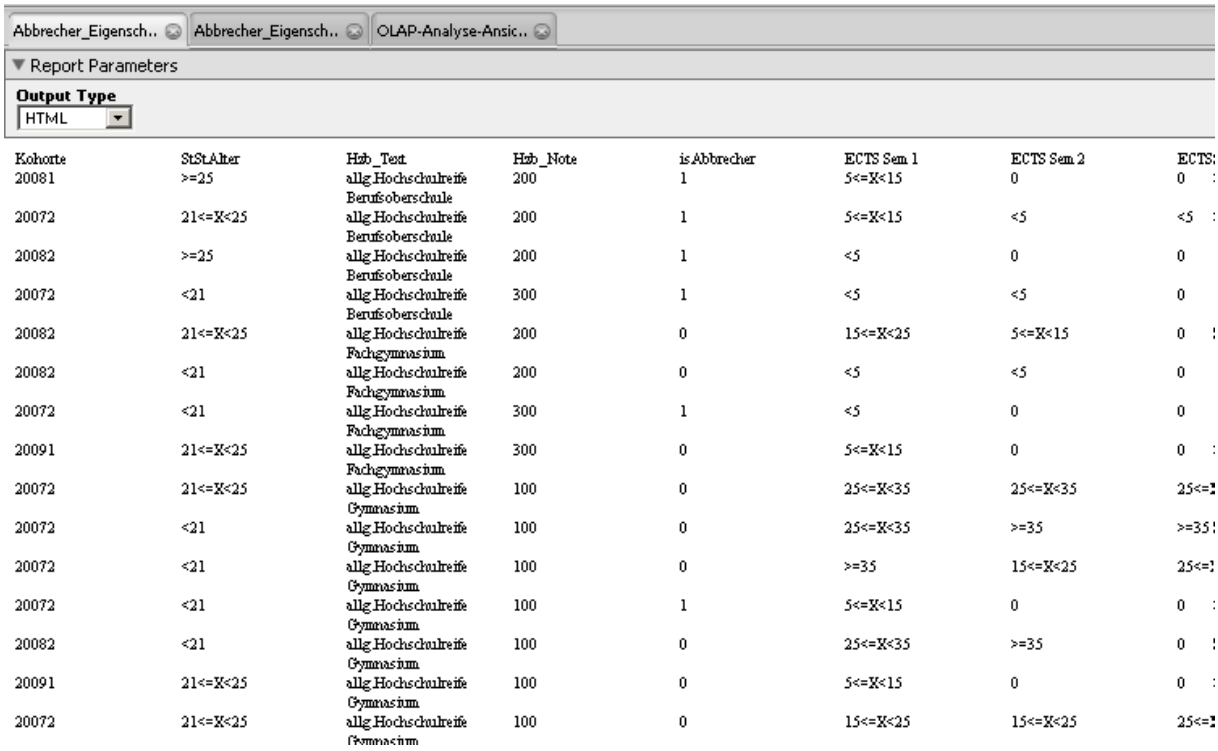
Quellcode 5.4: Bisher Bestanden

Die zweite Kennzahl diskretisiert die Anzahl ECTS-Punkte pro Studiensemester, wie im Business-Case unter 5.2.2.1 verlangt, siehe Quellcode 5.5.

```
member [Measures].[ECTS-Summe pro SS Diskretisiert] as
'CASE WHEN ([Measures].[ECTS-Summe pro SS] < 5.0) THEN "<5"
WHEN ([Measures].[ECTS-Summe pro SS] < 15.0) THEN "5<=X<15"
WHEN ([Measures].[ECTS-Summe pro SS] < 25.0) THEN "15<=X<25"
WHEN ([Measures].[ECTS-Summe pro SS] < 35.0) THEN "25<=X<35"
WHEN ([Measures].[ECTS-Summe pro SS] > 34.0) THEN ">=35" END'
```

Quellcode 5.5: ECTS-Summe pro SS Diskretisiert

Aus der MDX-Abfrage haben wir mittels *Pentaho Report Designer* einen Bericht erstellt und auf dem *BI-Server* veröffentlicht. Einen exemplarischen Ausschnitt zeigt Abbildung 5.14.



Kohorte	StStAlter	Hzb_Text	Hzb_Note	isAbbrecher	ECTS Sem 1	ECTS Sem 2	ECTS
20081	>=25	allg.Hochschulreife Berufsoberschule	200	1	5<=X<15	0	0
20072	21<=X<25	allg.Hochschulreife Berufsoberschule	200	1	5<=X<15	<5	<5
20082	>=25	allg.Hochschulreife Berufsoberschule	200	1	<5	0	0
20072	<21	allg.Hochschulreife Berufsoberschule	300	1	<5	<5	0
20082	21<=X<25	allg.Hochschulreife Fachgymnasium	200	0	15<=X<25	5<=X<15	0
20082	<21	allg.Hochschulreife Fachgymnasium	200	0	<5	<5	0
20072	<21	allg.Hochschulreife Fachgymnasium	300	1	<5	0	0
20091	21<=X<25	allg.Hochschulreife Fachgymnasium	300	0	5<=X<15	0	0
20072	21<=X<25	allg.Hochschulreife Gymnasium	100	0	25<=X<35	25<=X<35	25<=
20072	<21	allg.Hochschulreife Gymnasium	100	0	25<=X<35	>=35	>=35
20072	<21	allg.Hochschulreife Gymnasium	100	0	>=35	15<=X<25	25<=
20072	<21	allg.Hochschulreife Gymnasium	100	1	5<=X<15	0	0
20082	<21	allg.Hochschulreife Gymnasium	100	0	25<=X<35	>=35	0
20091	21<=X<25	allg.Hochschulreife Gymnasium	100	0	5<=X<15	0	0
20072	21<=X<25	allg.Hochschulreife Gymnasium	100	0	15<=X<25	15<=X<25	25<=

Abb. 5.14: Bericht Eigenschaften Abbrecher

5.6.3 Lehrveranstaltungen nach kumulierten Fachsemestern

Mit drei abgeleiteten Kennzahlen werden hier die nötigen Inhalte erzeugt, siehe Quellcode 5.6.

gibt ohne Kommastellen die Anzahl an leistenden Studenten an; da diese Abfrage auf bestandene Leistungen beschränkt ist, gibt sie damit die Anzahl an Studenten an, die eine Prüfung bestanden haben.

 # alle gibt, wie die Tabelle aus 5.3 verlangt, die Anzahl der eingeschriebenen Studenten an; in diesem Teilwürfel sind das diejenigen eines Studienfachs, die noch nicht abgebrochen haben. Man beachte, dass dazu die Dimensionen „hasStatus“, „isAbgeschlossen“, „pruefung“, „pruefung_legende“ auf ihrer höchsten Hierarchiestufe betrachtet werden müssen. Denn „Studenten Anzahl“ stammt aus dem Data-Cube der Stammblätter, der diese Dimensionen nicht mit dem Data-Cube der Leistungen teilt.

#1.FS gibt die Anzahl der aktuell Studierenden im ersten Fachsemester an. Besonders wichtig war uns hier, dass die Abfrage auch nach Aktualisierung der Bachelordaten funktioniert, also ohne konkrete Nennung des aktuellen Fachsemesters. Dazu wurde „Studenten Anzahl“ aus denjenigen Stammblättern ausgelesen, deren Studenten im bisher letzten aufgezeichneten Studiensemester im ersten Fachsemester sind.

Man beachte außerdem, dass zunächst die Prüfungen angezeigt werden, die mit „Pflicht“ beginnen, was durch einen *Mondrian OLAP Server*-spezifischen Regulären Ausdruck erreicht wird. Die Abfrage betrifft weiterhin nur Nicht-Abbrecher.

```
1  with member [Measures].[#] as 'Int([Measures].[Leistende Studenten Anzahl])'
2
3  member [Measures].[# alle] as
4  'IIf(IsEmpty([Measures].[#]), NULL, Int(([hasStatus].[All hasStatuss]
5  , [isAbgeschlossen].[All isAbgeschlossens], [pruefung].[All pruefungs]
6  , [pruefung_legende].[All pruefung_legendes], [Measures].[Studenten Anzahl])))'
7
8  member [Measures].[#1.FS] as 'IIf(IsEmpty([Measures].[#])
9  , NULL, Int(([hasStatus].[All hasStatuss], [isAbgeschlossen].[All isAbgeschlossens]
10 , [pruefung].[All pruefungs], [pruefung_legende].[All pruefung_legendes]
11 , [fachsemester].[All fachsemesters].[1], Tail(Filter([studiensemester].Members
12 , (NOT IsEmpty([Measures].[Studenten Anzahl])))).Item(0.0), [Measures].[Studenten
       Anzahl])))'
13
14 select NON EMPTY {
15 ([fachsemester].[All fachsemesters], [Measures].[# alle])
16 , ([fachsemester].[All fachsemesters], [Measures].[#1.FS])
17 , Crossjoin([fachsemester].[All fachsemesters].Children, {[Measures].[#]})}
18 ON COLUMNS,
19 NON EMPTY {Order(Crossjoin(Filter([pruefung].[P_TextKurz].Members
20 , (Filter([bereich].[Bereich_Kurz].Members
21 , (NOT IsEmpty([Measures].[Leistende Studenten Anzahl]))).Item(0.0).Name MATCHES
       "(?i)Pflicht.*"))
22 , [pruefung_legende].[P_TextKuerzest].Members),
       [pruefung].[P_TextKurz].CurrentMember.Name, BASC)
23 , Order(Crossjoin(Filter([pruefung].[P_TextKurz].Members
24 , NOT (Filter([bereich].[Bereich_Kurz].Members
25 , (NOT IsEmpty([Measures].[Leistende Studenten Anzahl]))).Item(0.0).Name MATCHES
       "(?i)Pflicht.*"))
26 , [pruefung_legende].[P_TextKuerzest].Members),
       [pruefung].[P_TextKurz].CurrentMember.Name, BASC)}
27 ON ROWS
28 from [uebersichtstudent]
```

```
29 where ([fach].[All fachs].[{studienfach}]
30 , [isAbbrecher].[All isAbbrechers].[0]
31 , [hasStatus].[All hasStatuss].[BE], [isAbgeschlossen].[0])
```

Quellcode 5.6: MDX-Ausdruck Lehrveranstaltungen nach kumulierten Fachsemestern

Eine exemplarische Version dieses Berichts ist in die Business-Story eingefügt worden, siehe Abbildung 5.2.

5.6.4 Lehrveranstaltungen nach aktuellen Fachsemestern

In der Abfrage wurde als Spaltendimension „last_fachsemester“ betrachtet. Für Nicht-Abbrecher, entspricht dies den aktuellen Fachsemestern, wie sie in Tabelle 5.4 gefordert sind. Die Anzahl der Studenten, die eine Prüfung bestanden haben, soll unter anderem prozentual zur Anzahl an eingeschriebenen Studenten angegeben werden; dazu wird vorher die Hilfskennzahl „Eingeschriebene Studenten“ berechnet. Quellcode 5.7 zeigt die Abfrage.

```
1  with member [Measures].[#] as 'Int([Measures].[Leistende Studenten Anzahl])'
2
3  member [Measures].[Eingeschriebene Studenten] as
4  '([pruefung].[All pruefungs], [pruefung_legende].[All pruefung_legendes]
5  , [hasStatus].[All hasStatuss], [isAbgeschlossen].[All isAbgeschlossens]
6  , [Measures].[Studenten Anzahl])'
7
8  member [Measures].[%] as 'Int((([Measures].[Leistende Studenten Anzahl]
9  / [Measures].[Eingeschriebene Studenten]) * 100.0))'
10
11 member [Measures].[Note] as 'Round([Measures].[Note Durchschnitt], 1.0)'
12
13 select NON EMPTY {Crossjoin([last_fachsemester].Children, {[Measures].[#]})
14 , Crossjoin([last_fachsemester].Children, {[Measures].[%]})
15 , Crossjoin([last_fachsemester].Children, {[Measures].[Note]})} ON COLUMNS,
16
17 NON EMPTY {Order(Crossjoin(Filter([pruefung].[P_TextKurz].members
18 , (Filter([bereich].[Bereich_Kurz].members
19 , not isEmpty([Measures].[Leistende Studenten Anzahl]))).item(0).name MATCHES
       "(?i)Pflicht.*"))
20 , [pruefung_legende].[P_TextKuerzest].members),
       [pruefung].[P_TextKurz].CurrentMember.name, BASC)
21 ,Order(Crossjoin(Filter([pruefung].[P_TextKurz].members,
       (Filter([bereich].[Bereich_Kurz].members
22 , not isEmpty([Measures].[Leistende Studenten Anzahl]))).item(0).name NOT MATCHES
       "(?i)Pflicht.*"))
23 , [pruefung_legende].[P_TextKuerzest].members),
       [pruefung].[P_TextKurz].CurrentMember.name, BASC)
24 } ON ROWS
25 from [uebersichtstudent]
26 where ([fach].[All fachs].[{studienfach}], [isAbbrecher].[All isAbbrechers].[0]
27 , [hasStatus].[All hasStatuss].[BE], [isAbgeschlossen].[All isAbgeschlossens].[0])
```

Quellcode 5.7: MDX-Ausdruck Lehrveranstaltungen nach aktuellen Fachsemestern

Auch dieser Bericht konnte direkt in der Business-Story verwendet werden und wird dort exemplarisch in Abbildung 5.3 gezeigt.

5.6.5 Veranstaltungen Trennschärfe

Um diesen Bericht zu erstellen, haben wir zwei abgeleitete Kennzahlen gebildet, „Abbrecher von Bestanden“ und „Abbrecher von Nicht-Bestanden“. Erstere gibt den prozentualen Anteil von Abbrechern unter denjenigen an, die eine Prüfung bestanden haben. Die zweite Kennzahl gibt den prozentualen Anteil von Abbrechern unter denjenigen an, die eine Prüfung nicht bestanden haben. Beide Kennzahlen werden ähnlich erstellt, Quellcode 5.8 zeigt den MDX-Ausdruck über „Abbrecher von Nicht-Bestanden“. Man beachte die Angabe eines „FORMAT_STRING“, um eine Prozentzahl zu erhalten. Auch die angeforderte Sortierung wird direkt in der MDX-Abfrage umgesetzt.

```
1 member [Measures].[Abbrecher von Nicht-Bestanden] as
2 '((([isAbbrecher].[All isAbbrechers].[1], [hasStatus].[All hasStatuss])
3 - ([isAbbrecher].[All isAbbrechers].[1], [hasStatus].[All hasStatuss].[BE]))
4 / (([isAbbrecher].[All isAbbrechers], [hasStatus].[All hasStatuss])
5 - ([isAbbrecher].[All isAbbrechers], [hasStatus].[All hasStatuss].[BE])))'
6 , FORMAT_STRING = "#,###.00%"
```

Quellcode 5.8: Abbrecher von Nicht-Bestanden

Ein Beispiel für die Ausgabe dieser Abfrage ist in der Business-Story in Abbildung 5.4 enthalten.

5.6.6 Studenten Komprimiert

Dieser Bericht enthält eine große Anzahl an Abgeleiteten Kennzahlen, die einzelne Spalten aus dem Bericht umsetzen und größtenteils selbsterklärend sind. Die Kennzahlen erhalten unter anderem Kürzel und werden gerundet.

Die Anzahl an Hochschulsemestern wird über die Anzahl an Studiensemestern, in denen ein Student eingeschrieben war, festgestellt. Diese Kennzahl wurde erst später in die Anforderungen aufgenommen. Stattdessen hätte man auch das Attribut „hssem“ aus der Datenquelle „sos“ in das Data-Warehouse aufnehmen können, für das in diesem Stadium jedoch mehr Aufwand nötig gewesen wäre.

Die Kennzahl „e/fs-1“ würde für Studenten, die im ersten Fachsemester studieren, die Teilung durch Null, eine unerlaubte mathematische Berechnung, verlangen. In diesem Fall wird stattdessen ein leerer Wert ausgegeben.

Neben dem Parameter {studienfach} wird auch die Dimension „isAbbrecher“ parametrisiert. So kann dieser Bericht auch für Abbrecher verwendet werden.

Die in der Anforderung genannte Sortierung wird direkt hier übernommen, durch zwei verschachtelte Sortierfunktionen, einmal auf den Namen der Dimension „last_fachsemester“, einmal auf die Kennzahl „ects“.

```
1  with member [Measures].[HS] as
2  'Int(Sum([studiensemester].[All studiensemesters].Children,
       [Measures].[isImmatrikuliert]))'
3
4  member [Measures].[isImmatrikuliert] as
5  'IIf((([pruefung].[All pruefungs], [bereich].[All bereichs], [hasStatus].[All hasStatuss]
6  , [isAbgeschlossen].[All isAbgeschlossens], [Measures].[Studenten Anzahl]) > 0.0), 1.0,
       NULL)'
7
8  member [Measures].[FS] as
9  'Int(([pruefung].[All pruefungs], [bereich].[All bereichs], [hasStatus].[All hasStatuss]
10 , [isAbgeschlossen].[All isAbgeschlossens], [Measures].[Letztes Fachsemester]))'
11
12 member [Measures].[ES] as
13 'Int(([pruefung].[All pruefungs], [bereich].[All bereichs], [hasStatus].[All hasStatuss]
14 , [isAbgeschlossen].[All isAbgeschlossens], [Measures].[Erstes Fachsemester]))'
15
16 member [Measures].[ects] as 'Int([Measures].[ECTS-Summe])'
17
18 member [Measures].[e/fs] as
19 'IIf(IsEmpty([Measures].[ECTS-Summe]), NULL, Int(Round(([Measures].[ECTS-Summe]
20 / [Measures].[FS]), 0.0)))'
21
22 member [Measures].[e/fs-1] as
23 'IIf(([Measures].[FS] > 1.0), Int(Round(([Measures].[ECTS-Summe]
24 / ([Measures].[FS] - 1.0)), 0.0)), NULL)'
25
26 member [Measures].[Note] as 'Round([Measures].[Note Durchschnitt], 1.0)'
27
28 select NON EMPTY {
29 Crossjoin(Crossjoin({[bereich].[All bereichs]}, {[bereich_legende].[All
       bereich_legendes]})
30 , {[Measures].[ES], [Measures].[HS]}), Crossjoin(Crossjoin({[bereich].[All bereichs]}
31 , {[bereich_legende].[All bereich_legendes]}), {[Measures].[ects], [Measures].[e/fs]
32 , [Measures].[e/fs-1], [Measures].[Note]}),
       Crossjoin(Crossjoin({[bereich].[Bereich_Kurz].Members}
33 , {[bereich_legende].[Bereich_Kuerzest].Members}), {[Measures].[ects],
       [Measures].[Note]})}
34 ON COLUMNS,
35
36 NON EMPTY Order(Order(Crossjoin([matnr].[All matnrs].Children
37 , [last_fachsemester].[All last_fachsemesters].Children), [Measures].[ects], DESC)
38 , [last_fachsemester].CurrentMember.Name, BDESC)
39 ON ROWS
40 from [uebersichtstudent]
41 where ([fach].[All fachs].[{studienfach}], [isAbbrecher].[All
       isAbbrechers].[{isAbbrecher}]
```

```
42 , [hasStatus].[All hasStatuss].[BE], [isAbgeschlossen].[All isAbgeschlossens].[0])
```

Quellcode 5.9: MDX-Ausdruck Studenten Komprimiert

Auch dieser Bericht konnte direkt in der Business-Story verwendet werden und wird dort exemplarisch in Abbildung 5.5 gezeigt.

5.6.7 Studenten – Detailliert

Dieser Bericht ist in den Kennzahlen und Zeilen identisch zum vorherigen aufgebaut, weshalb in Quellcode 5.10 nur die Spaltendimension beschrieben wird. Die Anforderung verlangt, dass Module und Teilmodule aus einem Bericht, der mit „Pflicht“ beginnt, aufgeschlüsselt werden; durch einen MDX-spezifischen Regulären Ausdruck haben wir dies erreicht.

```
1  select NON EMPTY {Crossjoin(Crossjoin(Crossjoin(Crossjoin({[bereich].[All bereichs]}
2  , {[bereich_legende].[All bereich_legendes]}), {[pruefung].[All pruefungs]})
3  , {[pruefung_legende].[All pruefung_legendes]}), {[Measures].[ES], [Measures].[HS]})
4  , Crossjoin(Crossjoin(Crossjoin(Crossjoin({[bereich].[All bereichs]}
5  , {[bereich_legende].[All bereich_legendes]}), {[pruefung].[All pruefungs]})
6  , {[pruefung_legende].[All pruefung_legendes]}), {[Measures].[ects], [Measures].[e/fs]
7  , [Measures].[e/fs -1], [Measures].[Note]})
8  , Crossjoin(Crossjoin(Crossjoin(Crossjoin([bereich].[Bereich_Kurz].Members
9  , {[bereich_legende].[Bereich_Kuerzest].Members}), {[pruefung].[All pruefungs]})
10 , {[pruefung_legende].[All pruefung_legendes]}), {[Measures].[ects], [Measures].[Note]})
11 , Crossjoin(Crossjoin(Crossjoin(Crossjoin(Filter([bereich].[Bereich_Kurz].Members
12 , ([bereich].[Bereich_Kurz].CurrentMember.Name MATCHES "(?i)Pflicht.*"))
13 , {[bereich_legende].[Bereich_Kuerzest].Members}), [pruefung].[P_TextKurz].Members)
14 , {[pruefung_legende].[P_TextKuerzest].Members}), {[Measures].[Note]})}
15 ON COLUMNS,
```

Quellcode 5.10: MDX-Ausdruck Studenten – Detailliert

Abbildung 5.6 zeigt in der Business-Story ein Beispiel für diesen Bericht.

5.6.8 ECTS-Verteilung

Hier haben wir die Kennzahl „e/fs“ verwendet, wie sie in Quellcode 5.9 für den Bericht aus Tabelle 5.6 erstellt wurde. Diese Kennzahl wird für alle Studierenden eines Studiengangs berechnet. Die Dimension „isAbbrecher“ wird verwendet, um in einer zweiten Spalte anzugeben, ob es sich um einen Abbrecher oder Nicht-Abbrecher handelt.

5.6.9 Vergleich Studiengänge

Für einen Vergleich der Studienfächer haben wir Kennzahlen erstellt, die die prozentuale Abweichung vom Durchschnitt, also den normierten Durchschnittswert der Note, Wiederholungs-

anzahl, ECTS-Punkte und Abbrecherquote angeben. Quellcode 5.11 zeigt die Berechnung am Beispiel der Note, die anderen Kennzahlen werden vergleichbar erstellt. Die Kennzahl der Wiederholungen haben wir mittels „Versuch Durchschnitt“ lediglich abgeschätzt. Diese Kennzahl errechnet – zumindest für Leistungen aus Teilmodulen, wie wir im Data-Assay festgestellt haben – den Durchschnitt der Versuchsanzahl und nicht die durchschnittliche Anzahl Wiederholungen, die zum Bestehen einer Prüfung nötig ist. Man kann mit Sicherheit eine MDX-Abfrage konzeptionieren, die die Wiederholungen berechnet. Des Weiteren wird die Normierung nach der Gesamtheit der Leistungen der Studierenden vorgenommen und nicht nach der Durchschnittsleistung eines Studienfachs. Diese Einschränkungen des Berichts werden auch in der Business-Story behandelt.

```
1 with member [Measures].[Note Durchschnitt normiert] as
2 '(([Measures].[Note Durchschnitt] - ([fach].[All fachs], [Measures].[Note Durchschnitt]))
3 / ([fach].[All fachs], [Measures].[Note Durchschnitt]))', FORMAT_STRING = "#,###.00%"
```

Quellcode 5.11: Note Durchschnitt normiert

Trotz der Einschränkungen kann dieser Bericht nützliche Informationen enthalten, weshalb ein exemplarischer Auszug in der Business-Story als Abbildung 5.8 enthalten ist.

5.6.10 Interdisziplinäre Module

Um ein interdisziplinäres Modul wie in Tabelle 5.10 beschrieben zu betrachten, sind zwei Kennzahlen nötig, ihre Berechnung wird in Quellcode 5.12 gezeigt. Für die Berechnung der Wiederholungen mussten die Teilmodule des Moduls betrachtet werden, da ausschließlich in diesen die Anzahl an Wiederholungen erfasst wird. Für diesen Bericht bestehen ähnliche Einschränkungen wie für den Bericht aus Tabelle 5.9; sie werden in der Business-Story erwähnt.

```
1 with member [Measures].[Note Durchschnitt Normiert] as
2 '((([Measures].[Note Durchschnitt] - ([fach].[All fachs], [Measures].[Note
      Durchschnitt]))
3 / ([fach].[All fachs], [Measures].[Note Durchschnitt])))', FORMAT_STRING = "#,###.00%"
4
5 member [Measures].[Wiederholungen Durchschnitt Normiert] as
6 '((([Measures].[Wiederholungen Durchschnitt]
7 - ([fach].[All fachs], [Measures].[Wiederholungen Durchschnitt]))
8 / ([fach].[All fachs], [Measures].[Wiederholungen Durchschnitt])))', FORMAT_STRING =
      "#,###.00%"
```

Quellcode 5.12: MDX-Ausdruck Interdisziplinäre Module

Auch dieser Bericht wird in der Business-Story demonstriert, in Abbildung 5.9.

5.7 Data-Mining

Im folgenden Abschnitt wird das Vorgehen zum Anwenden von Data-Mining-Techniken auf die Berichte beschrieben. Hierfür waren 5 Stunden notwendig.

Für die Tabelle 5.1 waren keine Data-Mining-Techniken notwendig. Die Tabellen 5.5, 5.9 und 5.10 haben lediglich eine Sortierung der Instanzen verlangt. Auch für die Anforderungen 5.3, 5.4, 5.6 und 5.7 hat sich das Data-Mining auf eine bestimmte Sortierung der Instanzen beschränkt, die bereits während des Reporting durchgeführt werden konnte. Die Berichte werden an die Dekane der Universität Würzburg weitergeleitet; diese können sie mit ihrem Hintergrundwissen zum jeweiligen Studienfach direkt interpretieren.

5.7.1 Abbrecher Eigenschaften

Im Bericht der Abbrecher aus Tabelle 5.2 haben wir Attributwerte darauf untersucht, ob sie das Attribut „isAbbrecher“ erklären können, bzw. inwiefern Unterschiede zwischen Abbrechern oder Nicht-Abbrechern bestehen. Dazu haben wir zwei Techniken angewandt: das Lernen eines Entscheidungsbaums und die Subgruppenentdeckung.

Für beides haben wir den Bericht jeweils für ein Studienfach ausgeführt und als CSV-Datei exportiert. Diese CSV-Datei haben wir mit *Weka* eingelesen, als ARFF-Datei gespeichert und manuell Einstellungen an der ARFF-Datei vorgenommen, z.B. die Attribute „Bisher Bestanden“ jedes Moduls von Numerischen Attributen zu binären Kategorischen Attributen mit den Werten „0“ und „1“ umgewandelt.

Quellcode 5.13 zeigt einen anonymisierten Auszug der ARFF-Kopfzeilen.

```
1  @relation BI-Server-Export
2  @attribute Kohorte {20072,20081,20082,20091,20092}
3  @attribute StStAlter {21<=X<25,<21,>=25}
4  @attribute Hzb_Text {'allg.Hochschulreife  Berufsoberschule','allg.Hochschulreife
       Fachgymnasium'
5  ,'allg.Hochschulreife  Gymnasium','allg.Hochschulreife  Kolleg'... }
6  @attribute Hzb_Note {100,200,300,400}
7  @attribute isAbbrecher {0,1}
8  @attribute ECTS_Sem_1 {0,15<=X<25,25<=X<35,5<=X<15,<5,>=35}
9  @attribute ECTS_Sem_2 {0,15<=X<25,25<=X<35,5<=X<15,<5,>=35}
10 @attribute ECTS_Sem_3 {0,15<=X<25,25<=X<35,5<=X<15,<5,>=35}
11 @attribute ECTS_Sem_4 {0,15<=X<25,25<=X<35,5<=X<15,<5,>=35}
12 @attribute Modul 1 {0,1}
13 @attribute Modul 2 {0,1}
14 @attribute Modul 3 {0,1}
15 @attribute Modul 4 {0,1}
16 ...
```

Quellcode 5.13: ARFF-Header Abbrecher Eigenschaften

Mit Weka haben wir anschließend für ausgewählte Studienfächer einen Entscheidungsbaum mit dem Algorithmus J48 [56] erstellt. Diesen haben wir jedoch nicht als hilfreich empfunden, um die Frage nach Gründen für Abbrecher zu beantworten. Abbildung 5.15 zeigt zur Veranschaulichung einen anonymisierten Entscheidungsbaum eines Studienfachs.

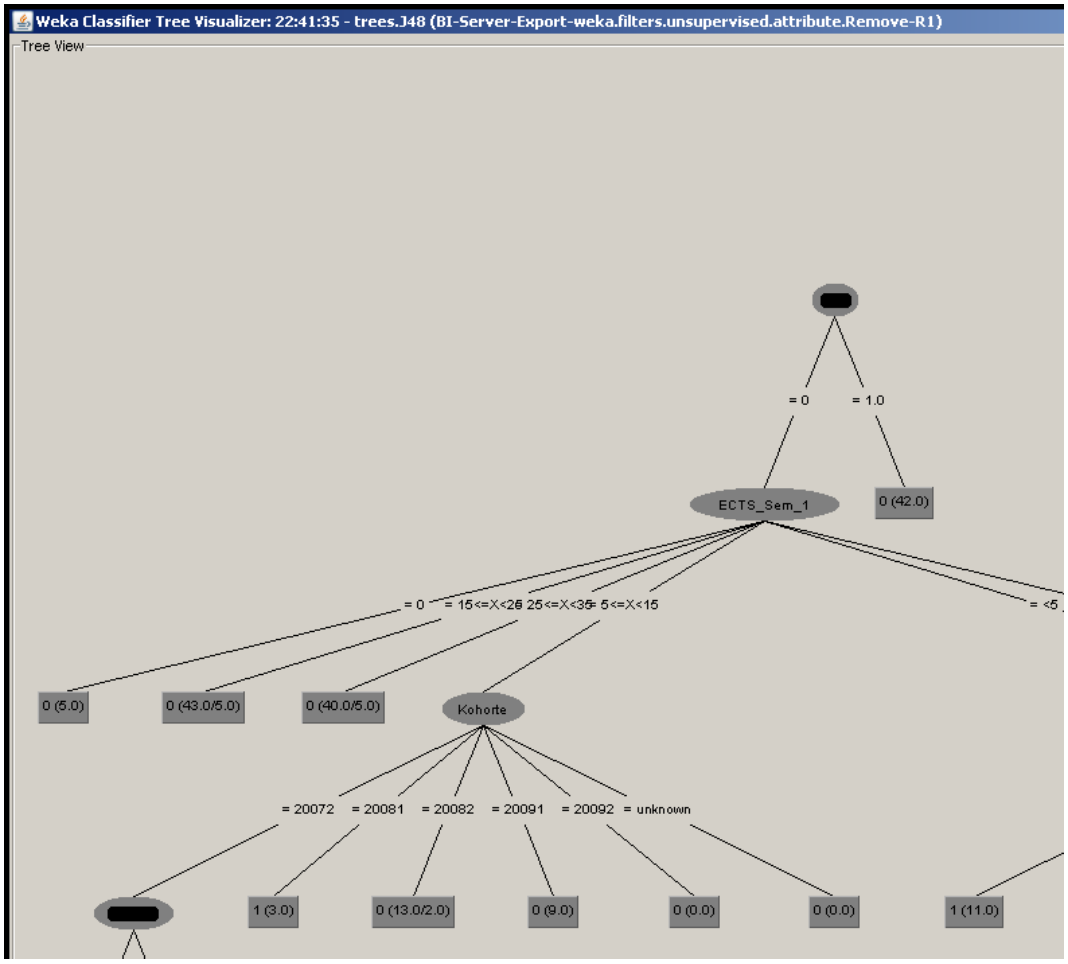


Abb. 5.15: Blätter im Entscheidungsbaum: „0“ (Nicht-Abbrecher), „1“ (Abbrecher)

Vielversprechender ist in diesem Fall die Subgruppenentdeckung. Dazu haben wir die ARFF-Datei in *VIKAMINE* eingelesen und den Algorithmus SD-Map ausgewählt, um eine „vollständige, aber effiziente“ [5, S.47] Suche nach Subgruppen vorzunehmen.

Wir suchen in dem Bericht nach Subgruppen, die sich im Bezug auf das Attribut „isAbbrecher“ von der Gesamtpopulation deutlich unterscheiden; d.h. wir suchen nach Subgruppen, deren Prozentsatz an Abbrechern deutlich größer als in der Gesamtpopulation ist. Der Algorithmus ließ sich durch Parameter konfigurieren: Als Qualitätsfunktion haben wir die „weighted relative accuracy“ (WRACC) verwendet, die als vielversprechende Kennzahl für interessante Subgruppen gilt (vgl. [42]; [5, S. 28]). Frei nach *Ockham's Razor* haben wir Subgruppen bevorzugt, die sich

durch die Kombination möglichst weniger Attributwerte (maximal 5) ausdrücken lassen. Gleichzeitig sollten nur Subgruppen gefunden werden, die mindestens 20 Instanzen, also Studenten, aufweisen. Es hat sich gezeigt, dass die Subgruppenentdeckung ein großes Potenzial besitzt, um Gründe für Studienabbrecher zu entdecken bzw. Vermutungen zu bestätigen. Daher wurde die Tabelle 5.11 mit exemplarischen Ergebnissen der Business-Story hinzugefügt.

5.7.2 ECTS-Verteilung

Wir haben den Bericht zur Anforderung 5.8 als CSV-Datei exportiert, und ähnlich wie für Tabelle 5.2 in eine ARFF-Datei umgewandelt. Diese Datei haben wir mit *Rattle* eingelesen und die Verteilung der ECTS-Punkte anzeigen lassen. Ein Beispiel für eine solche Verteilung ist im Business-Case in Abbildung 5.7 zu sehen.

6 Ergebnisse der Einzelfallstudie: CaseTrain

Dieses Kapitel beschreibt die Einzelfallstudie zum Projekt CaseTrain, in dem das Ziel verfolgt wird, ein fallbasiertes Lehrsystem der Universität Würzburg anhand von Benutzungsdaten zu beurteilen. Das Kapitel ist wie folgt strukturiert:

Ein Management-Summary fasst die wichtigsten Ergebnisse des Projekts zusammen. Ein Business-Case beschreibt den Hintergrund und die Ziele des Projekts. In einer Business-Story werden dann die Ergebnisse beschrieben. Die darauffolgenden Abschnitte beschreiben das Vorgehen im Projekt zum Data-Assay, Data-Warehouse, Reporting und Data-Mining.

6.1 Management-Summary

Ziel des Projekts ist es, den Nutzen des fallbasierten Lehrsystems CaseTrain, welches den Studierenden der Universität Würzburg über das Internet angeboten wird, mittels Benutzungsdaten zu beurteilen. Jedes Jahr muss ein Antrag gestellt werden, um dieses System mittels Studiengebühren finanzieren zu können, weshalb eine solche Leistungskontrolle wichtig ist.

Über CaseTrain bieten Dozenten den Studierenden praxisnahe Fälle an, um den Lernstoff effektiver lernen zu können. Während einer Fallbearbeitung beantworten die Lernenden eine Reihe von Fragen, werden dabei durch Hilfsfunktionen des Systems unterstützt und erhalten am Ende, sofern sie nicht vorher die Bearbeitung abbrechen, eine Fallzusammenfassung sowie die Prozentzahl der korrekt beantworteten Fragen.

Im Folgenden einige Fragestellungen, die wir im Projekt behandelt haben:

1. Wie zufrieden sind die Lernenden mit CaseTrain?
2. Welchen Einfluss hat das Lernen mit CaseTrain auf Prüfungsergebnisse?
3. Werden die Hilfsfunktionen des Lehrsystems wie beabsichtigt genutzt?
4. Wie können die Autoren dabei unterstützt werden, effektivere Fälle zu erstellen?

Fragestellung 1 beantworten wir durch detaillierte Auflistungen von Umfrageergebnissen, in denen die Lernenden zu CaseTrain Schulnoten vergeben und Feedbacktexte schreiben können. CaseTrain hat eine durchschnittliche Note von 1,9 erhalten. Die Abbruchquote in CaseTrain

beträgt 32,7%, unterscheidet sich jedoch deutlich zwischen den Fällen, weshalb wir auch aufgeschlüsselte Berichte erstellt haben. In den letzten zwei Jahren ist die Nutzungshäufigkeit von CaseTrain deutlich gestiegen (siehe Abbildung 6.1 in der Business-Story). Die Lernenden haben durchschnittlich 67,5% der Fragen richtig beantwortet und sind grundsätzlich interessiert an der Beantwortung der Fragen und nicht nur an der Fallzusammenfassung, insbesondere, wenn die Fälle von den Dozenten als prüfungsrelevant angeboten werden. Dies hat sich in Form einer überdurchschnittlichen Beschäftigungsdauer mit CaseTrain gezeigt, als wir Fragestellung 2 behandelt haben. Allerdings hat sich die Vermutung nicht bestätigen lassen, dass das Lernen mit CaseTrain zu besseren Prüfungsergebnissen führt (siehe Abbildung 6.4); dies vermutlich auch deshalb, weil noch zu wenige Daten vorhanden sind. Mit mehr Prüfungsergebnissen besitzt diese Untersuchung vielversprechende Möglichkeiten, um den Nutzen von CaseTrain zu belegen.

Für Fragestellung 3 haben wir detaillierte Statistiken über die Benutzung von Hilfsfunktionen erstellt. Demnach wurden diese „Features“ von CaseTrain meist erwartungsgemäß benutzt. Diese Untersuchung der Hilfsfunktionen ergab zudem einen Einblick in das Lernverhalten der Studierenden, auf regelmäßiger Basis kann sie aufschlussreiche Kennzahlen zur Verwendung von CaseTrain bieten.

Autoren scheinen sich häufig schwer zu tun, den Aufwand für das Bearbeiten eines Falls richtig einzuschätzen. Dabei hat die Zeit, die ein Lernender an einem Fall arbeiten muss, sicherlich einen Einfluss auf seine Motivation. Um Fragestellung 4 zu behandeln und die Autoren dabei zu unterstützen, gute Fälle zu erstellen, können wir ihnen die durchschnittlich nötige Fallbearbeitungsdauer sowie die durchschnittliche Zeit, die ein Lernender ohne Unterbrechung mit einem Fall beschäftigt sein möchte, mitteilen. Aber auch andere Kennzahlen zu den Fällen, z.B. Bewertungen der Studierenden, können hilfreiches Feedback für die Autoren bieten – unter der Voraussetzung, dass es in komprimierter und verständlicher Form übergeben wird.

Ein wichtiges Ergebnis des Projekts liegt im erstellten Data-Warehouse. Die genannten, aber auch jederzeit weitere Untersuchungen lassen sich darüber vereinfacht erstellen, zudem mit aktuellen Daten. Es bietet sich an, aussagekräftige Kennzahlen zu definieren, die die Qualität von CaseTrain objektiv bewerten. Das Projekt hat dazu erste Möglichkeiten identifiziert und umgesetzt. Wenn das Data-Warehouse entsprechend weiterentwickelt wird, könnten seine Ausgaben die Argumentationsgrundlage des jährlichen Finanzierungsantrags von CaseTrain darstellen; dieser ließe sich damit im Idealfall sogar automatisieren.

6.2 Business-Case

Die Inhalte des Business-Case haben sich in mehreren Gesprächen zwischen den Projekt-Beteiligten sowie während der Projektdurchführung ergeben. Hervorgehobene Begriffe wurden

im Business-Case in einem Glossar beschrieben, der auch hier unter Abschnitt 6.2.6 zu finden ist, aber auch über den Index dieser Arbeit erreicht werden kann.

6.2.1 Hintergrund und Motivation

Fallbasiertes Lernen wird in zunehmend vielen Lehreinrichtungen genutzt, um Lernenden die Möglichkeit zu geben, theoretische Kenntnisse anhand realistischer Problemfälle praxisnah anzuwenden. An der Universität Würzburg wird im Rahmen des seit 2007 bestehenden Projekts „Blended-Learning“ das Autoren- und Ablaufsystem „CaseTrain“ über das Internet angeboten, um Fallbasiertes Lernen einer möglichst großen Anzahl an Studierenden zugänglich zu machen.

Das CaseTrain-Projekt wird aus Studienbeiträgen finanziert, jedes Jahr muss eine Verlängerung des Projekts beantragt werden; daher ist eine kontinuierliche Beurteilung und Verbesserung wichtig. So gilt es u.a., die Lehrmethode des Fallbasierten Lernens an sich zu bewerten. Dazu muss eine ausreichende Qualität der Technik und der Fälle sichergestellt sein, weshalb auch die Technik des Lehrsystems und die angebotenen Fälle zu beurteilen sind.

Das langfristige Ziel ist eine häufigere und weiter verbreitete Anwendung des Fallbasierten Lernens durch das CaseTrain-Projekt.

6.2.2 Problemstellung und Möglichkeiten

Im Folgenden werden Fragestellungen beschrieben, die im Rahmen des hier beschriebenen Data-Mining-Projekts behandelt werden sollen.

6.2.2.1 Methode Fallbasiertes Lernen

Um die Methode des Fallbasierten Lernens allgemein zu beurteilen, wurden drei relevante Fragestellungen identifiziert:

1. Wie zufrieden sind die Lernenden mit der Methode des Fallbasierten Lernens (Spaß, Motivation, Lernerfolg)?
2. Wird die Methode des Fallbasierten Lernens anders angewendet als beabsichtigt?
3. Welche Auswirkungen hat das Lernen mit CaseTrain auf Prüfungsergebnisse?

Maßnahmen sind dann z.B. die anderweitige Integration der Methode in den Studiengang bzw. die Veränderung des Systems oder der Fälle.

6.2.2.2 Technik Fallbasiertes Lernen

Um Möglichkeiten zur Verbesserung der Technik festzustellen, sind zwei weitere Fragestellungen interessant:

1. Wie zufrieden sind die Lernenden mit der Technik des Lehrsystems?
2. Wie nützlich sind Zusatzfunktionen (z.B. Nutzerhilfen) des Systems?

Mögliche Maßnahmen sind abgeänderte oder neue Zusatzfunktionen für die Lernenden.

6.2.2.3 Qualität der Fälle

Um die Qualität der Fälle zu beurteilen, sind zwei Fragestellungen von Bedeutung:

1. Wie zufrieden sind die Lernenden mit einzelnen Fällen?
2. Welche Unterschiede finden sich zwischen einzelnen Fällen?

Maßnahmen dazu sind die Beratung der Dozenten bei der Erstellung der Fälle.

6.2.3 Aktuelle Situation und Datenlage

Voraussetzungen für eine Analyse sind Aufzeichnungen über das Verhalten der Lernenden, während sie Fälle bearbeiten – also das System nutzen. Solche Aufzeichnungen wurden als *Logdaten* automatisch gespeichert und stehen für das Projekt über einen Zeitraum von zwei Jahren, seit Wintersemester 2007, zur Verfügung.

6.2.3.1 Logdaten

Die Logdaten protokollieren jede Aktion der Lernenden während sie einen Fall bearbeiten und können im Normalfall, wenn der Zugriff auf CaseTrain über „WueCampus“, der universitätsweiten eLearning-Plattform, geschieht, einem Studenten eindeutig zugeordnet werden. Jede Fallbearbeitung besteht den Logdaten nach aus dem Beantworten von Fragen. Für jede Antwort erhält der Lernende einen *Score*; am Ende der Fallbearbeitung werden die Scores zu einem *Gesamtscore* zusammenfügt.

Daneben wird beispielsweise die Ausführung von Zusatzfunktionen, sog. *Features*, des Ablaufsystems protokolliert. Zu unterscheiden sind dabei zwei verschiedene Formen. Einerseits gibt es Features, bei denen nur eine Häufigkeit der Nutzung angegeben werden kann:

Pause Die Pausierung und Wiederaufnahme einer Fallbearbeitung.

Hintergrundinfo Die Weiterleitung zu allgemeinen Hintergrundinformationen zum Fall über einen Web-Link des Autors, typischerweise zu einer Webseite mit Vorlesungsfolien.

Link Die Weiterleitung zu speziellen Hintergrundinformationen zum Fall über einen Web-Link des Autors.

Außerdem gibt es Features, die aktiviert bzw. geöffnet und nach einiger Zeit wieder deaktiviert bzw. geschlossen werden. Bei ihnen kann auch eine Nutzungsdauer angegeben werden:

Introinfo Die Anzeige eines kurzen Einführungstexts zu CaseTrain.

Fallverlauf Anzeige einer textuellen Zusammenfassung des bisherigen Fallverlaufs – am Ende eine vollständige Zusammenfassung des Falls.

Fragehinweis Anzeige von Hinweisen des Autors zu einer bestimmten Frage.

Lösungskommentar Anzeige des Lösungskommentars des Autors zur Antwort einer bestimmten Frage.

Bild Die Vergrößerung eines Bildes.

Zum Ende der Bearbeitung eines Falls werden den Lernenden bei über 90% der Fälle Standard-Fragen zur Evaluation gestellt:

Fallnote Zufriedenheit mit dem Fall als Schulnote.

Bediennote Zufriedenheit mit der Bedienung als Schulnote.

Feedbacktext Freitext-Verbesserungsvorschläge zum Fall oder CaseTrain.

Auf Grund einer automatischen und höchst-redundanten Logdatenerfassung kann von einer hohen Datenqualität ausgegangen werden.

6.2.3.2 Meta-Informationen

Zusätzliche Informationen zu den Fällen beschreiben Einstellungen, z.B. zur Evaluation, sowie Hinweise der Autoren, z.B. die erwartete Fallbearbeitungsdauer.

6.2.3.3 Prüfungsergebnisse

Zu einigen Fällen können die Ergebnisse der Prüfungen, in denen sie prüfungsrelevant waren, angegeben werden. So kann festgestellt werden, wie erfolgreich der Bearbeiter eines prüfungsrelevanten Falls in der Prüfung war.

Für drei Prüfungen sind die Ergebnisse sowie relevante Fälle aus CaseTrain bekannt: Für die Übungsprüfung und Hauptprüfung der Veranstaltung „Quantitative Methoden A“ sowie für die

Prüfung der Veranstaltung „Anwendungsorientierte Informatik“.

Die Logdaten werden bereits auf einige Fragestellungen analysiert. Ein Programm liest sie dazu in eigene Datenstrukturen ein und erlaubt die Betrachtung der Auswertungen in einem Webinterface. Neue Fragestellungen lassen sich nur durch eine Änderung am Programmcode behandeln. Auch lassen sich zusätzliche Datenquellen, beispielsweise Prüfungsergebnisse, nur mit hohem Aufwand berücksichtigen.

6.2.4 Alternative und empfohlene Lösungen

Im Folgenden werden Ansätze genannt, um die Fragestellungen mittels verfügbarer Datenquellen zu behandeln. Außerdem wird darauf eingegangen, inwiefern alternative Lösungen, ggf. mit Zusatzdaten, möglich sind.

6.2.4.1 Anforderungen: Zufriedenheit mit Methode, Technik und Fällen

Die subjektive Meinung zur Fallbasierten Lernmethode, zur Technik und zu einzelnen Fällen kann durch Umfragen der Studenten am besten abgeschätzt werden (Fragestellungen 1 aus 6.2.2.1, 6.2.2.2 und 6.2.2.3). Mit ihnen können Studenten explizit nach ihrer Zufriedenheit befragt werden. Wie in Abschnitt 6.2.3.1 beschrieben, erhalten die Studenten in den meisten Fällen Standard-Fragen zur Evaluation. Diese Standard-Evaluationen mit Fallnote, Bediennote und Feedbacktext bieten eine unmittelbare Möglichkeit zur Einschätzung der Zufriedenheit mit Methode, Technik und den einzelnen Fällen.

Um weitere Informationen zur Zufriedenheit der Lernenden zu erhalten, müssten diese Befragungen erweitert, beispielsweise neue Fragen hinzugefügt werden. Um eine größere Anzahl an Studierenden zu befragen, könnte eine neue Umfrage auch außerhalb des Ablaufsystems erfolgen. Allerdings versprechen Befragungen grundsätzlich nur einen mäßigen Rücklauf, insbesondere kurzfristig. Des Weiteren wären diese Informationen nicht weniger subjektiv. Aus diesen Gründen lohnt es sich vorerst nicht, diese Möglichkeit der individuellen Umfragen zu berücksichtigen.

Mittels der Meta-Informationen (siehe Abschnitt 6.2.3.2) kann man Fallbearbeitungen identifizieren, in denen Standard-Evaluationen verwendet werden. Ihre Informationen lassen sich eindeutig interpretieren, dabei wollen wir die Fallnote, Bediennote und den Feedbacktext jeweils im Bezug zu einer konkreten *Fallversion* betrachten.

In einer ersten Anforderung (siehe Tabelle 6.1) soll herausgefunden werden, welche Fälle besonders schlecht bewertet werden. Sie gibt für jede Fallversion die Anzahl an gegebenen Fall- und Bediennoten sowie ihren Durchschnitt an und sortiert den Bericht absteigend nach der Durchschnittsfallnote.

Name:	**Evaluationsnoten**
Eingabe:	Logdaten, Meta-Informationen
Bericht:	Zeilen: Fallversion mit Standard-Evaluation
	Spalten: Fallnote Anzahl; Fallnote Durchschnitt; Bediennote Anzahl; Bediennote Durchschnitt
Muster:	Absteigend sortiert nach Fallnote

Tab. 6.1: Anforderung Evaluationsnoten

In der zweiten Anforderung (siehe Tabelle 6.2) soll herausgefunden werden, welche Feedbacktexte relevante Informationen beinhalten.

Name:	**Feedbacktexte**
Eingabe:	Logdaten, Meta-Informationen
Bericht:	Zeilen: Version eines Falls; Nicht-leerer Feedbacktext
	Spalten: Datum; Fallnote; Bediennote
Muster:	Für jede Fallversion nach Datum sortiert, so dass jüngere Feedbacktexte weiter oben angezeigt werden.

Tab. 6.2: Anforderung Feedbacktexte

Eine objektivere Einschätzung zur Zufriedenheit mit der Fallbasierten Methode, der Technik und der Fälle gibt das Verhältnis zwischen abgebrochenen und beendeten Fallbearbeitungen; Tabelle 6.3 zeigt diese Anforderung, in der Fälle, die besonders häufig abgebrochen werden, herausgefunden werden.

Name:	**Abbruchquoten**
Eingabe:	Logdaten
Bericht:	Zeilen: Version eines Falls
	Spalten: Fallbearbeitungen Anzahl; Abbrüche Prozent
Muster:	Nach Abbruchquote absteigend sortiert

Tab. 6.3: Anforderung Abbruchquoten

In diesem Zusammenhang ist interessant, ob die Nutzung von CaseTrain in einer Veranstaltung verpflichtend ist. Solche Informationen können nur durch Befragungen der Studierenden oder Dozenten mit Sicherheit festgestellt werden; es ist zweifelhaft, ob der Aufwand gerechtfertigt ist, weshalb diese Möglichkeit bisher nicht in Betracht gezogen wird.

6.2.4.2 Anforderungen: Methode Fallbasiertes Lernen

Eine hohe kumulierte Dauer der Anzeige des Fallverlaufs am Ende einer Fallbearbeitung, vor allem in Relation zur *Bearbeitungsdauer* und zum erreichten *Gesamtscore*, deutet darauf hin, dass solche Lernende nur die Fallzusammenfassung zum Lernen des Prüfungsstoffes nutzen. Hiermit wird Fragestellung 2 aus Abschnitt 6.2.2.1 behandelt. Tabelle 6.4 beschreibt diese Anforderung, in der folgende Fragen beantwortet werden: Finden sich Lernende, die den Fallverlauf besonders lange anschauen? Haben diese Lernenden auch einen geringen Gesamtscore? Die Beantwortung dieser Fragen wird Hinweise dazu geben, ob manche Lernende mehr an der Zusammenfassung der Fälle als an der Beantwortung der Fragen interessiert sind. Bei abgebrochenen Bearbeitungen kann der Fallverlauf nicht am Ende betrachtet worden sein, weshalb wir in dieser Anforderung nur beendete Fallbearbeitungen berücksichtigen.

Name:	**Fallverlauf Ende**
Eingabe:	Logdaten
Bericht:	Zeilen: Nicht-anonyme Lernende
	Spalten: beendete Fallbearbeitungen Anzahl; Gesamtscore Durchschnitt; Anteil Durchschnitt der Summe der Dauer des Fallverlaufs am Ende pro beendeter Fallbearbeitung von Durchschnitt der Gesamtbearbeitungsdauer
Muster:	Diagramm, das den Anteil, den Gesamtscore und die Anzahl an Fallbearbeitungen gemeinsam darstellt.

Tab. 6.4: Anforderung Fallverlauf Ende

Eine vielversprechende Möglichkeit zur Bewertung der Methode des Fallbasierten Lernens bietet der Vergleich zwischen dem Lerneinsatz mit CaseTrain und den Ergebnissen in der Prüfung, für die gelernt wurde. Der Lerneinsatz kann durch die Anzahl an Fallbearbeitungen, die durchschnittliche bzw. aufsummierte *Bearbeitungsdauer* und Gesamtbearbeitungsdauer sowie den durchschnittlichen, maximal erreichten oder aufsummierten Gesamtscore ausgedrückt werden. Da wir von relativ wenig Studenten Prüfungsergebnisse und relevante Übungsfälle kennen (siehe Abschnitt 6.2.3.3), soll nicht zwischen einzelnen Prüfungen unterschieden werden. Hier ist interessant, ob ein Zusammenhang zwischen einer der Kennzahlen des Lerneinsatzes und der Prüfungsleistung vorliegt.

Des Weiteren kann die durchschnittliche Zeit, die ein Lernender ohne Unterbrechung an einem Fall arbeitet – die Dauer der *Kontinuierlichen Fallbearbeitung* – etwas über die grundsätzliche Nutzung der Methode des Fallbasierten Lernens aussagen, siehe Anforderung aus Tabelle 6.6. Damit soll herausgefunden werden, wie lange die Lernenden im Durchschnitt ohne Unterbrechung arbeiten und inwiefern die Tatsache über einen Abbruch und die *Bearbeitungsnummer*

Name:	**Prüfungsergebnisse**
Eingabe:	Logdaten, Prüfungsergebnisse
Bericht:	Zeilen: Prüfungsleistung eines nicht-anonymen Lernenden
	Spalten: Anzahl Fallbearbeitungen; Bearbeitungsdauer Durchschnitt/-Summe; Gesamtdauer Durchschnitt/Summe; Gesamtscore Durchschnitt/Max/Summe; Prüfungsergebnis; Prüfungsnote
Muster:	Berechnung der Abhängigkeit zwischen einzelnen Attributen; Bei hohen Abhängigkeiten zwischen zwei Attributen werden die Attribute gegeneinander in einem Diagramm dargestellt.

Tab. 6.5: Anforderung Pruefungsergebnisse

eine Rolle spielen – am Beispiel für den Zeitraum nach dem 1. Juni 2009.

Name:	**Kontinuierliche Fallbearbeitung**
Eingabe:	Logdaten
Bericht:	Zeilen: Kontinuierliche Bearbeitung in einer Fallbearbeitung nach dem 1. Juni 2009
	Spalten: abgebrochene oder nicht-abgebrochene Fallbearbeitung; Bearbeitungsnummer; Dauer
Muster:	Darstellung von Kennzahlen der Dauer in Abhängigkeit der Abbruchinformation bzw. der Bearbeitungsnummer.

Tab. 6.6: Anforderung Kontinuierliche Fallbearbeitung

6.2.4.3 Anforderungen: Technik Fallbasiertes Lernen

Um Fragestellung 2 aus Abschnitt 6.2.2.2 zu behandeln, soll die Nutzungshäufigkeit und -dauer der *Features* betrachtet werden. Es werden unterschieden: Features, die allgemein in einer Fallbearbeitung genutzt werden, ohne Bezug zu einer Frage (sog. „Fallbearbeitungsaktionen“). Außerdem Features, die im Kontext einer Frage verwendet werden (sog. „Scoreaktionen“).

In der Anforderung aus Tabelle 6.7 werden Nutzungsinformationen zu den Features „Pause“, „Bild“, „Fallverlauf“, „Hintergrundinfo“, „Introinfo“ und „Link“ (vgl. Abschnitt 6.2.3.1) angegeben. Außerdem führen wir ein zusätzliches Feature, „Bearbeitung“, ein, das die *Kontinuierliche Fallbearbeitung* beschreibt. Die Nutzung dieser Features haben wir im Begriff Fallbearbeitungsaktionen zusammengefasst.

Für jedes Feature wird der Zeitpunkt der Betätigung – der *Bearbeitungsstatus* – unterschieden (Anfang, Mitte, Ende). Die Zahlen geben Hinweise darauf, ob Features anders als erwartet

verwendet werden. Da wir davon ausgehen, dass das Verhalten der Lernenden in abgebrochenen Fallbearbeitungen grundsätzlich vom typischen Verhalten abweicht, werden ausschließlich beendete Fallbearbeitungen betrachtet.

Name:	**Fallbearbeitungsaktionen Frequenztabelle**
Eingabe:	Logdaten
Bericht:	Zeilen: Bearbeitungsstatus eines Features
	Spalten: Anzahl beendeter Fallbearbeitungen; Fallbearbeitungsaktionen Anzahl insgesamt; Fallbearbeitungsaktionen Anzahl Durchschnitt pro Fallbearbeitung; Fallbearbeitungsaktionen Dauer insgesamt; Fallbearbeitungsaktionen Dauer Durchschnitt pro Fallbearbeitung
Muster:	-

Tab. 6.7: Anforderung Fallbearbeitungsaktionen Frequenztabelle

An Features, die im Zusammenhang mit einer Frage verwendet werden, werden „Fragehinweis“ und „Lösungskommentar“ (vgl. 6.2.3.1) unterschieden. Wir haben die Nutzung dieser beiden Features durch den Begriff *Scoreaktionen* benannt. Sie werden getrennt für abgebrochene und nicht-abgebrochene Fallbearbeitungen sowie unterschiedliche *Bearbeitungsnummern* betrachtet. Tabelle 6.8 beschreibt diese Anforderung. Auch hier geben die Zahlen Hinweise darauf, ob Features anders als erwartet verwendet werden, z.B. wie häufig das Feature für einen Score verwendet wird; außerdem wie lange es verwendet wird im Durchschnitt unter allen Scores bzw. im Durchschnitt unter den Scores, in denen es überhaupt verwendet wird.

Name:	**Scoreaktionen Frequenztabelle**
Eingabe:	Logdaten
Bericht:	Zeilen: abgebrochene Fälle insgesamt; abgebrochene Fälle je Bearbeitungsnummer; nicht-abgebrochene Fälle insgesamt; nicht-abgebrochene Fälle je Bearbeitungsnummer
	Spalten: Scores Anzahl insgesamt; Score Durchschnitt; Fragehinweis Dauer Durchschnitt unter Scores, bei denen Fragehinweis verwendet; Fragehinweis Dauer Durchschnitt unter allen Scores; Lösungskommentar Dauer Durchschnitt unter Scores, bei denen Lösungskommentar verwendet; Lösungskommentar Dauer Durchschnitt unter allen Scores; Fragehinweis Häufigkeit Durchschnitt unter Scores; Lösungskommentar Häufigkeit Durchschnitt unter Scores
Muster:	-

Tab. 6.8: Anforderung Scoreaktionen Frequenztabelle

Auch um zu analysieren, wie sich die Dauer der Anzeige des Lösungskommentars zum *Score* einer Frage verhält, wird eine entsprechende Anforderung erstellt, siehe Tabelle 6.9. Da die Anzahl an aufgezeichneten Scores sehr groß ist, soll nur eine repräsentative Teilmenge betrachtet werden. Daher werden nur Scores von Erstbearbeitungen herangezogen, da diese vermutlich am ernsthaftesten bearbeitet worden sind. Außerdem werden nur Fallbearbeitungen der letzten Prüfungsphase (hier: nach dem 1. Juni 2009) einbezogen. Es wird vermutet, dass Lösungskommentare, die sehr lange (über zwei Stunden) angezeigt wurden, Ausnahmen darstellen, die in diesem Zusammenhang unbedeutend sind; auch sie werden nicht berücksichtigt. Es soll herausgefunden werden, ob der Score einer Antwort Auswirkungen hat auf die Dauer der Anzeige des Lösungskommentars. So wird vermutet, dass die Lernenden bei niedrigeren Scores den Lösungskommentar deutlich länger betrachten.

Name:	**Lösungskommentar und Score**
Eingabe:	Logdaten
Bericht:	Zeilen: Score einer beendeten Fallbearbeitung, einer Erstbearbeitung, einer Bearbeitung nach dem 1. Juni 2009, mit Nutzungsdauer des Lösungskommentars unter zwei Stunden
	Spalten: Score, Lösungskommentar Dauer
Muster:	Abhängigkeit der beiden Attribute; Diagramm, das beide Attribute gegeneinander zeichnet.

Tab. 6.9: Anforderung Lösungskommentar und Score

6.2.4.4 Anforderungen: Qualität Fälle

Zur Behandlung der Fragestellung 2 aus Abschnitt 6.2.2.3, werden laut Anforderung aus Tabelle 6.10 für jede Version eines Falls, der Unterschied zwischen der *Bearbeitungsdauer* und der vom Autor vermuteten Dauer (*Autordauer*) angegeben, außerdem Fälle hervorgehoben, bei denen ein großer Unterschied zwischen den beiden Werten besteht. Dies nur für beendete Fallbearbeitungen.

Zusätzlich haben wir uns entschlossen, eine Übersicht über die Anzahl an Fallbearbeitungen über den in den Logdaten beschriebenen Zeitraum anzuzeigen. Dies einmal mit Unterscheidung von Abbrüchen, siehe Tabelle 6.11 und einmal mit Unterscheidung der *Bearbeitungsnummer*, siehe Tabelle 6.12. Denn es ist interessant, wo die Bearbeitungsspitzen liegen und ob es Auffälligkeiten in der Verteilung der Abbrüche oder Bearbeitungsnummern über die Monate hinweg gibt.

Name:	**Autordauer**
Eingabe:	Logdaten, Meta-Informationen
Bericht:	Zeilen: Version eines Falls
	Spalten: beendete Fallbearbeitungen Anzahl; Bearbeitungsdauer Durchschnitt; Autordauer; absolute Differenz zwischen Bearbeitungsdauer Durchschnitt und Autordauer
Muster:	Absteigend nach absoluter Differenz sortiert.

Tab. 6.10: Anforderung Autordauer

Name:	**Bearbeitungsspitzen Abbrüche**
Eingabe:	Logdaten
Bericht:	Zeilen: Monat
	Spalten: Anzahl nicht-abgebrochene Fallbearbeitungen; Anzahl abgebrochene Fallbearbeitungen
Muster:	Säulendiagramm der Anzahl Fallbearbeitungen über Monate

Tab. 6.11: Anforderung Bearbeitungsspitzen Abbrüche

Name:	**Bearbeitungsspitzen Bearbeitungsnummer**
Eingabe:	Logdaten
Bericht:	Zeilen: Monat
	Spalten: je Bearbeitungsnummer, Anzahl Fallbearbeitungen
Muster:	Säulendiagramm der Anzahl Fallbearbeitungen über Monate

Tab. 6.12: Anforderung Bearbeitungsspitzen Bearbeitungsnummer

6.2.5 Projektplanung

6.2.5.1 Beteiligte

Entscheidungsträger und Auftraggeber ist Professor Frank Puppe, der Leiter des Projekts CaseTrain. Die CaseTrain-Entwickler Marianus Ifland und Alexander Hörnlein bieten Daten- und Domänenexpertise. Der Autor dieser Arbeit dient als Data-Mining-Experte.

6.2.5.2 Ressourcen

Es werden ausschließlich kostenfrei nutzbare Open-Source-Werkzeuge eingesetzt. Als Hardware werden drei herkömmliche Arbeitsplatzrechner benötigt, einer für ein Data-Warehouse, ein

zweiter für die Analysen, ein dritter zur Verwaltung der Dokumentation.

6.2.5.3 Projektplan

Der Projektplan teilt sich in folgende Meilensteine auf:

Data-Assay Es werden die Daten beschrieben und aufbereitet, so dass sie für die Anforderungen geeignet sind.

Data-Warehouse Es wird ein Data-Warehouse erstellt, das Abfragen ermöglicht, um die angeforderten Berichte zu erstellen.

Reporting Es werden die angeforderten Berichte erstellt.

Data-Mining Es werden in den Berichten die angeforderten Muster gesucht.

Änderungen des Business-Case werden dem Entscheidungsträger vorgelegt. Dieser kann während des Projekts stets den Status Quo in Form von Beschreibungen des Data-Assay, Demonstrationen des Data-Warehouse, Auszüge der Reports und Demonstrationen des Data-Mining erhalten.

Am Ende des Projekts erhält der Entscheidungsträger die Ergebnisse des Projekts in Form einer Business-Story.

6.2.6 Glossar

6.2.6.1 Autordauer

Dabei handelt es sich um eine Einschätzung des Fallautors, wie lange die Bearbeitung im Durchschnitt dauern sollte.

6.2.6.2 Bearbeitungsnummer

Hier wird durch eine Zahl beschrieben, wieviele Bearbeitungen einer Fallversion ein Lernender bereits durchgeführt hat. Es werden die erste, zweite, und letzte sowie alle mittleren Fallbearbeitungen unterschieden.

6.2.6.3 Bearbeitungsstatus

Der Status einer Fallbearbeitungsaktion beschreibt, in welchem Stadium der Fallbearbeitung eine Aktion durchgeführt worden ist; es werden hierbei Anfang, bevor der eigentliche Fall startet, Mitte, während der Beantwortung der Fragen, und Ende, nachdem die Beantwortung der Fragen abgeschlossen worden ist, unterschieden.

6.2.6.4 Bearbeitungsdauer

Die Bearbeitungsdauer beschreibt die Zeit, die der Lernende mit der Beantwortung der Fragen beschäftigt war. Sie beginnt mit der ersten und endet bei der Beantwortung der letzten Frage. Die Gesamtbearbeitungsdauer dagegen beschreibt die gesamte Zeit, die der Lernende mit der Fallbearbeitung verbringt.

6.2.6.5 Fallversion

Jeder Fall besitzt ein oder mehrere Versionen, die sich teilweise stark unterscheiden können und daher getrennt voneinander betrachtet werden.

6.2.6.6 Feature

Ein Feature stellt eine Zusatzfunktion des Ablaufsystems dar, die von den Lernenden bei Bedarf genutzt werden kann, beispielsweise das Betätigen einer Pause oder die Anzeige einer Beschreibung des bisherigen Fallverlaufs.

6.2.6.7 Gesamtscore

Aus den einzelnen *Scores* ergibt sich der Gesamtscore, eine Zahl zwischen 0 und 1 und eine zusammenfassende Bewertung der Fallbearbeitung.

6.2.6.8 Kontinuierliche Fallbearbeitung

Eine Kontinuierliche Fallbearbeitung beschreibt die Zeit, die ein Lernender ohne Unterbrechung an einer Fallbearbeitung verbringt. Sie benennt den Zeitraum zwischen dem Beginn bzw. der Wiederaufnahme einer Fallbearbeitung nach einer Pause und dem Abschließen, dem Abbruch oder der Pausierung der Fallbearbeitung.

6.2.6.9 Score

Der Score beschreibt mit einer Zahl zwischen 0 und 1 die Bewertung einer Frageantwort.

6.3 Business-Story

Die Business-Story wird dem Auftraggeber als Ergebnis des Projekts übergeben. Informationen darüber, wie die Ergebnisse erzeugt wurden, sind in den darauffolgenden Kapiteln enthalten.

Zur Business-Story gehört ein Management-Summary, das dieser Einzelfallstudie als Einleitung vorangesetzt wurde, siehe Abschnitt 6.1.

6.3.1 Ziel des Projekts CaseTrain

Ziel dieses Projekts ist es, Maßnahmen zu empfehlen, die das Fallbasierte Lernen mit CaseTrain an der Universität Würzburg an Bedeutung gewinnen lässt. Dazu sollen die Methode des Fallbasierten Lernens, die Technik von CaseTrain sowie die Fälle der Autoren bewertet werden.

Im Folgenden wird auf jedes dieser Teilziele anhand der Anforderungen aus dem Business-Case eingegangen.

6.3.2 Entdeckungen und Begründungen

Der Datensatz, den wir analysiert haben, besteht größtenteils aus Aufzeichnungen von Aktionen der Lernenden in CaseTrain. Sie beschreiben 169.244 Fallbearbeitungen von mindestens 2.608 verschiedenen Studierenden der Universität, die zwischen dem 15.11.2007 und dem 06.08.2009 durchgeführt worden sind.

6.3.2.1 Bearbeitungsspitzen

Wir haben keine Auffälligkeiten bezüglich der Anzahl an Fallbearbeitungen innerhalb dieser Zeit entdeckt. Mit Sicherheit kann gesagt werden, dass die Anzahl an Fällen und/oder Lernenden stetig wächst und mit ihr die Anzahl an Fallbearbeitungen. Die Bearbeitungsspitzen sind markant. In einem fortführenden Data-Mining-Projekt kann man jede Spitze separat untersuchen und normiert in Abhängigkeit anderer betrachten. Auf diese Weise können auch direkt Änderungen an der Methode, der Technik oder der Fälle verfolgt und Auswirkungen gezeigt werden.

Abbildungen 6.1 und 6.2 geben einen Eindruck über den Verlauf der Fallbearbeitungen, einmal mit Unterscheidung von abgebrochen Fallbearbeitungen, und einmal mit Unterscheidung einzelner *Bearbeitungsnummern*. Im ersten Diagramm beschreibt die Kategorie „0“ die Anzahl an Nicht-Abbrüchen, „1“ die Anzahl an Abbrüchen. Im zweiten Diagramm sind mit der Kategorie „1“ die Erstbearbeitungen, mit „2“ die Zweitbearbeitungen, mit „55“ die mittleren Bearbeitungen und mit „99“ die Letztbearbeitungen gemeint.

Die Diagramme weisen keine relevanten Auffälligkeiten bezüglich der Anzahl an Fallbearbeitungen auf. Die eine Fallbearbeitung im Jahr 27 ist ein irrelevanter Aufzeichnungsfehler. Zu Beginn der Erfassungszeit, die gleichzeitig die Lancierung des Lehrsystems markiert, bestanden hauptsächlich mittlere Bearbeitungen, was auf wenige Lernende, möglicherweise zum Testen des Systems, während dieser Zeit hindeutet.

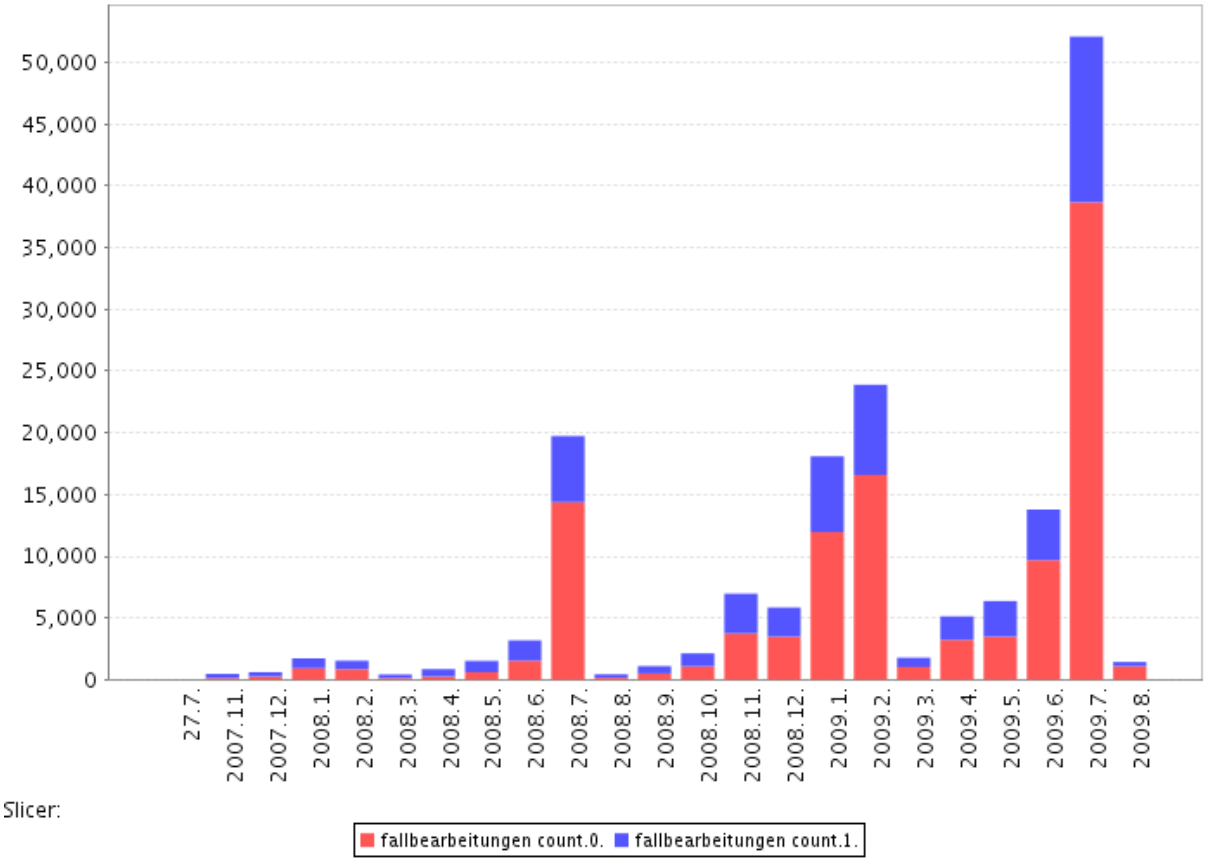


Abb. 6.1: Anzahl Fallbearbeitungen (y-Achse) pro Monat (x-Achse); rot: beendete Fallbearbeitungen; blau: abgebrochene Fallbearbeitungen

Dieses Ergebnis behandelt die Anforderungen aus Tabellen 6.11 und 6.12 des Business-Case.

6.3.2.2 Evaluationsnoten und Feedbacktexte

In diesem und dem darauffolgenden Abschnitt werden die Ergebnisse zur Zufriedenheit der Lernenden mit der Methode Fallbasiertes Lernen, der Technik von CaseTrain und den Fällen beschrieben, siehe Abschnitt 6.2.4.1 im Business-Case.

Die durchschnittliche Bediennote liegt bei 1,91; die durchschnittliche Fallnote liegt bei 1,99 – wenn eine Note vergeben wird, wird auch die andere Note vergeben, insgesamt 8.937 Mal.

Wir haben die Anforderung aus Tabellen 6.1 und 6.2 erfüllt. Für jeden einzelnen Fall verfügen wir über eine Zusammenfassung der Bedien- und Fallnoten sowie die Feedbacktexte.

Allerdings relativiert sich die Signifikanz der durchschnittlichen Fallnote jeweils anhand der Anzahl an Fallbearbeitungen. Möglich wäre es, eine weitere *Kennzahl* herzunehmen, die beide Informationen berücksichtigt und zur Sortierung verwendet wird. In weiteren Analysen kann man

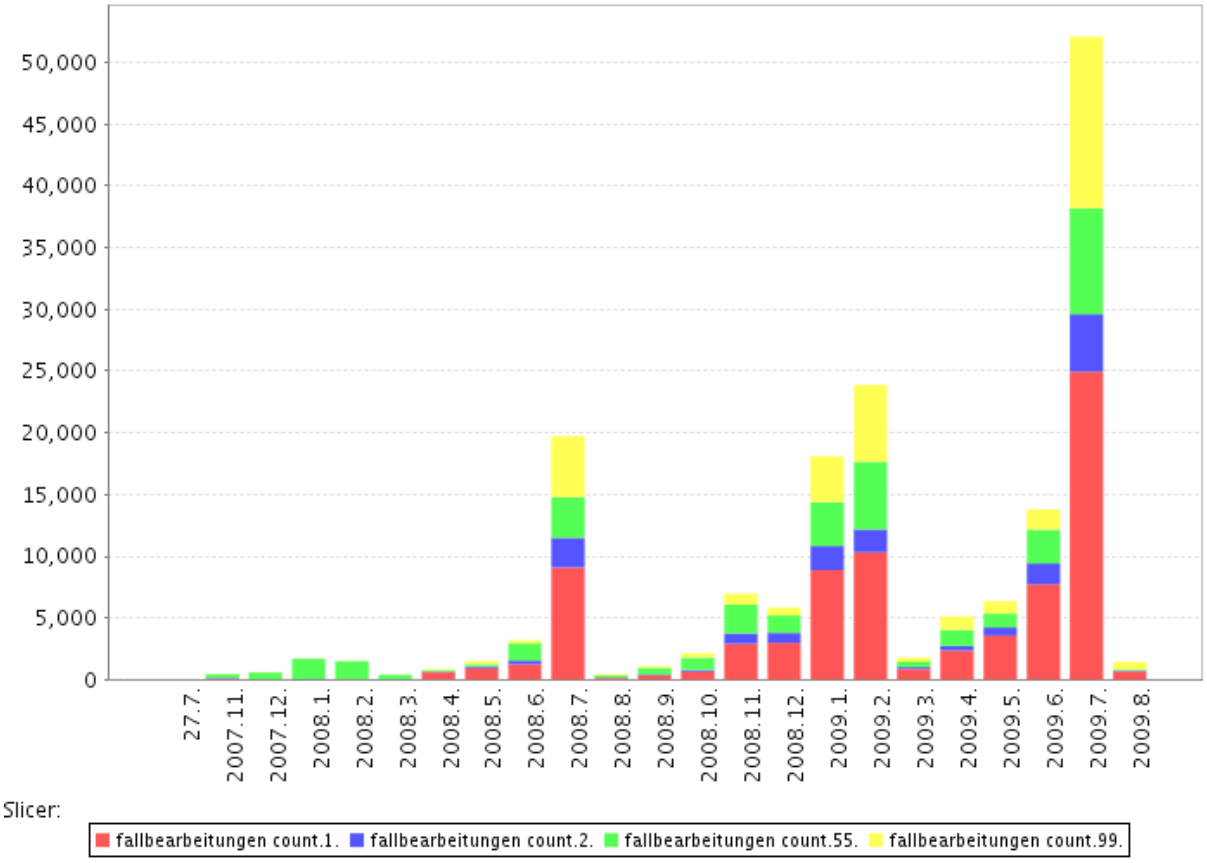


Abb. 6.2: Anzahl Fallbearbeitungen (y-Achse) pro Monat (x-Achse); fallbearbeitungen.count.1: Erstbearbeitungen; 2: Zweitbearbeitungen; 55: mittlere Fallbearbeitungen; 99: Letztbearbeitungen

die Objektivität der Evaluationsnoten anhand des *Gesamtscore* abschätzen – als zusätzlicher Indikator für ernstzunehmende Evaluationsnoten.

Ein Vorschlag wäre, Durchschnittsnoten, die sich aus mehr als einer bestimmten Anzahl an Evaluationen zusammensetzen, sowie Feedbacktexte, für einen relevanten Zeitraum, den Autoren per E-Mail zuzuschicken – möglicherweise in regelmäßigen Abständen. Das würde ihnen Feedback geben und sie motivieren, bessere Fälle zu erstellen.

In weiteren Analysen der Feedbacktexte, z.B. mittles Techniken zum *Text-Mining*, können auch weitere Fragestellungen behandelt werden; beispielsweise kann überprüft werden, ob eine Verbindung zwischen negativen oder positiven Bewertungen im Feedbacktext und den enthaltenen Evaluationsnoten vorhanden ist.

6.3.2.3 Abbruchquote

Auch die Abbruchquoten haben wir nach der Anforderung aus Tabelle 6.3 für alle Fallversionen erstellt. Die Quote beträgt insgesamt 32,68%; Autoren, deren Fälle häufig durchgeführt, aber auch überdurchschnittlich oft abgebrochen werden, können darüber in Kenntnis gesetzt werden.

Bezüglich der Abbruchquote empfehlen wir, in einem fortführenden Projekt eine Fragestellung genauer zu behandeln: Warum brechen die Lernenden Fallbearbeitungen ab?

6.3.2.4 Fallverlauf Ende

In diesem und den nächsten beiden Abschnitten wird nun genauer auf die Methode des Fallbasierten Lernens eingegangen, wie in Abschnitt 6.2.4.2 gefordert.

Es bestand die Vermutung, dass ein Teil der Lernenden ausschließlich über das Feature der Anzeige des Fallverlaufs am Ende lernt. Diese Vermutung ließ sich nicht bestätigen, zu gering war der Anteil an auffälligen Lernenden und zu hoch deren durchschnittlicher Gesamtscore, wie das Diagramm in Abbildung 6.3 zeigt.

Es zeigt auf einer horizontalen Achse den durchschnittlichen Anteil des Fallverlaufs am Ende zur Gesamtbearbeitungsdauer und auf der vertikalen Achse den durchschnittlichen Gesamtscore aller Lernenden. Die Größe jedes Punktes visualisiert die Anzahl an Fallbearbeitungen.

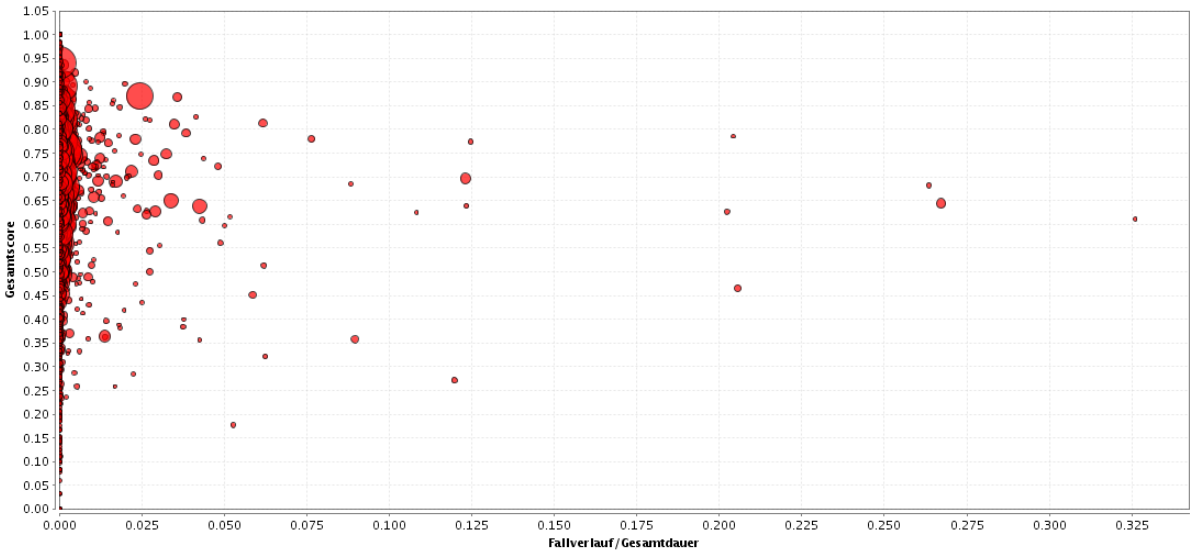


Abb. 6.3: Für jeden Lernenden: Anteil der Fallzusammenfassung an Gesamtdauer (x-Achse) zu durchschnittlichem Gesamtscore (y-Achse) sowie Anzahl an Fallbearbeitungen (Größe des Punktes)

Demnach verbringt der Großteil der Studierenden einen Bruchteil der Gesamtbearbeitungsdauer

mit dem Betrachten des Fallverlaufs am Ende. Auch diejenigen, die mehr Zeit damit verbringen, besitzen keinen übermäßig niedrigen Gesamtscore. Allerdings enthält diese Analyse auch Einschränkungen: So enthalten manche Fälle nur eine kurze Zusammenfassung, die zum Lernen nicht verwendet werden kann. In weiterführenden Untersuchungen könnten Fälle getrennt voneinander betrachtet werden. Durch die Anzahl an enthaltenen Wörtern sowie in Relation zur durchschnittlichen Bearbeitungsdauer des Falls könnte abgeschätzt werden, wie gut sich eine Fallzusammenfassung zum Lernen eignet und in Abhängigkeit dieser Indikatoren das Interesse der Lernenden an der Zusammenfassung festgestellt werden.

Was außerdem auffällt: Die Lernenden mit vielen Fallbearbeitungen besitzen einen motivierenden durchschnittlichen Gesamtscore zwischen 65% und 90%. Unter allen Fallbearbeitungen wurde ein durchschnittlicher Gesamtscore von 0,675 vergeben.

Für eine genauere Untersuchung der Effizienz der Fallbasierten Methode kann man in einem fortführenden Data-Mining-Projekt Lernende in Gruppen aufteilen, z.B. *schlechte* und *gute* Lernende. Anstatt solche Attribute selbst zu definieren, ließe sich auch für die Menge an Lernenden eine Reihe von *Kennzahlen* bestimmen und solche ausfindig machen, die verschiedene Lernende am besten charakterisieren – und sie darüber definieren. Innerhalb einzelner Gruppen an Lernenden werden weitere interessante Muster erwartet.

6.3.2.5 Prüfungsergebnisse

Langfristig interessant ist die Anforderung aus Tabelle 6.5. Ein Zusammenhang zwischen der Anzahl an Fallbearbeitungen und dem Prüfungsergebnis hat sich nicht bewahrheitet. Stattdessen zeigt sich eine Abhängigkeit zwischen dem durchschnittlichen Gesamtscore und der Leistung in der Prüfung. Diese ist jedoch nicht aussagekräftig genug, da sie auch durch einen nicht betrachteten Faktor beeinflusst worden sein kann: die Vorkenntnisse der Studierenden. Dieses Anfangsniveau ließe sich durch weitere Prüfungsergebnisse, auch solche für die nicht mit CaseTrain gelernt wurde, feststellen. In weiteren Analysen könnte dann untersucht werden, inwiefern ein Studierender sich durch das Lernen mit CaseTrain von seinem Durchschnitt entfernt hat. Um die Intensität des Lernens mit CaseTrain besser einzuschätzen, könnte neben einzelnen Faktoren wie der Anzahl Fallbearbeitungen, der Gesamtdauer sowie dem durchschnittlichen Score auch eine gewichtete Kombination als Indikator verwendet werden. Es werden weitere Prüfungsergebnisse benötigt, um das Potenzial dieser Untersuchung ausnutzen zu können.

Abbildung 6.4 stellt die Korrelationskoeffizienten der Attribute aus einer *Korrelationstabelle* dar. In dieser Visualisierung wird jede Korrelation zwischen zwei Attributen als Oval dargestellt. Je schmaler von links oben nach rechts unten verlaufend bzw. je stärker rot gefärbt, desto kleiner der Korrelationskoeffizient, desto größer also die negative Korrelation. Dagegen, je schmaler von links unten nach rechts oben verlaufend bzw. je stärker blau gefärbt, desto größer der

Korrelationskoeffizient und die positive Korrelation.

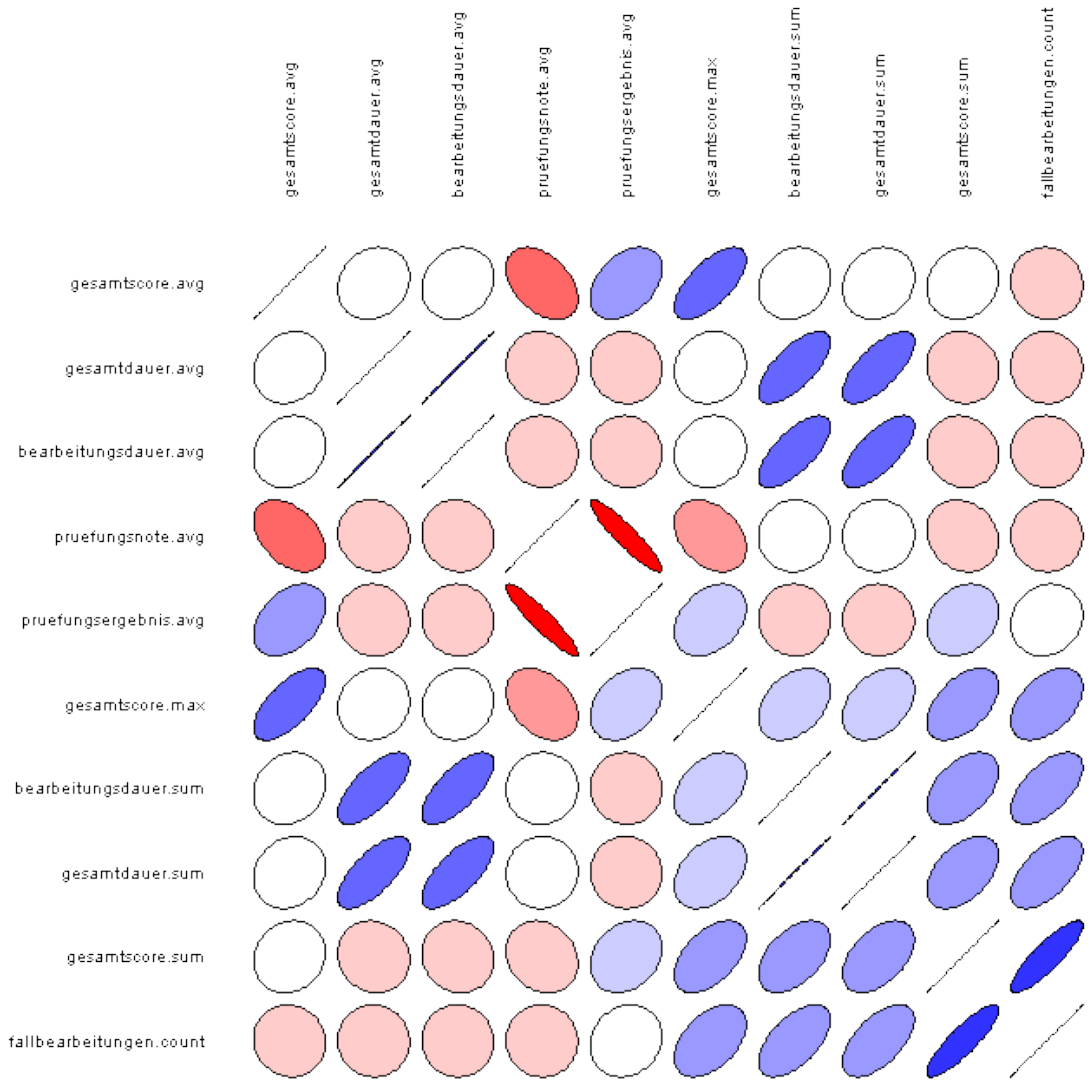


Abb. 6.4: Visualisierung der Korrelationskoeffizienten; rot: negativ korreliert; blau: positiv korreliert; Je stärker gefärbt bzw. schmaler geformt, desto stärker die Abhängigkeit

Im Diagramm wird deutlich:

- Wie zu erwarten, sind Gesamtdauer Durchschnitt/Summe und Bearbeitungsdauer Durchschnitt/Summe stark zueinander korrelierend, genauso wie Prüfungsergebnis und Prüfungsnote.
- Der maximale Gesamtscore unter allen Fallbearbeitungen korreliert positiv mit der Anzahl an Fallbearbeitungen – auch hier aus Wahrscheinlichkeitsgründen, wie erwartet.

- Die interessanteste Korrelation: Der durchschnittliche Gesamtscore korreliert positiv mit dem Prüfungsergebnis – je höher der durchschnittliche Gesamtscore, desto höher das Prüfungsergebnis bzw. niedriger die Note.

Die interessante Abhängigkeit zwischen Gesamtscore und Prüfungsergebnis weist einen Korrelationskoeffizienten von 0,4 auf, weder besonders niedrig, noch besonders hoch. Diese Korrelation haben wir weiter mit einem Diagramm untersucht, das für jeden Lernenden den durchschnittlichen Gesamtscore und das Prüfungsergebnis gegeneinander zeichnet, siehe Abbildung 6.5.

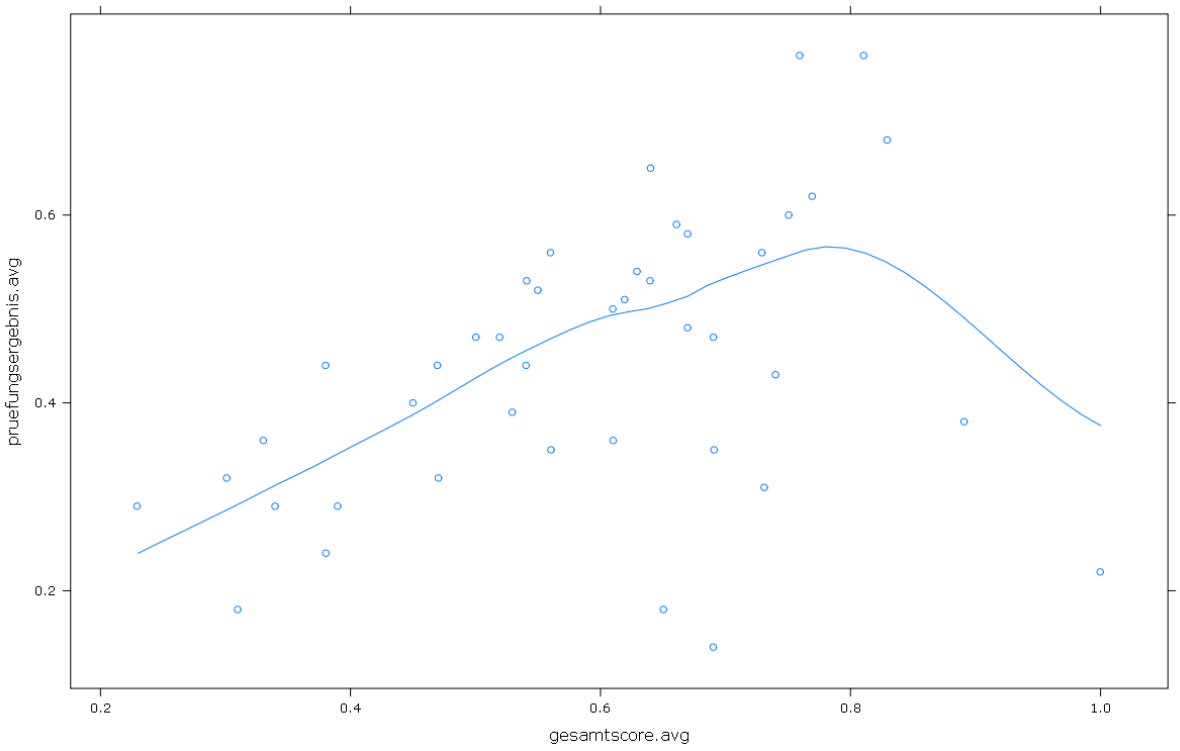


Abb. 6.5: Betrachtung der Lernenden; Durchschnitt Gesamtscore in Fallbearbeitungen (x-Achse) und durchschnittliches Ergebnis in den Prüfungen (y-Achse)

Darin wird deutlich, dass bereits ein Lernender Auswirkungen auf die Korrelation hat, da er, obwohl er lange mit CaseTrain gelernt, ein niedriges Ergebnis in der Prüfung erreicht hat. Ein Indiz dafür, dass noch zu wenige Prüfungsergebnisse vorhanden sind. Eine weitere Sammlung von Prüfungsleistungen, idealerweise automatisiert, kann deutlich signifikantere Ergebnisse liefern.

Ein weiteres Diagramm, ein sog. *Box-Plot*, siehe Abbildung 6.6, zeigt Kennzahlen zur Summe der Gesamtbearbeitungsdauer der Lernenden. Demnach haben die Studierenden im Durchschnitt 20.000 bis 30.000 Sekunden – ca. fünf bis sechs Stunden – mit CaseTrain verbracht. Es gibt allerdings auch Studierende, die kaum mit CaseTrain gelernt haben, was von Aussagen der Studierenden in einer Umfrage abweicht, nach der sie zumindest 50% der Lernzeit mit CaseTrain verbracht hatten. Ein anonymes Lernen ist nicht möglich, da die relevanten Fälle ausschließlich

über „WueCampus“ erreichbar waren. Wir empfehlen, in einer späteren Umfrage zu überprüfen, ob Lernende auch gemeinsam an einem Rechner mit CaseTrain arbeiten.

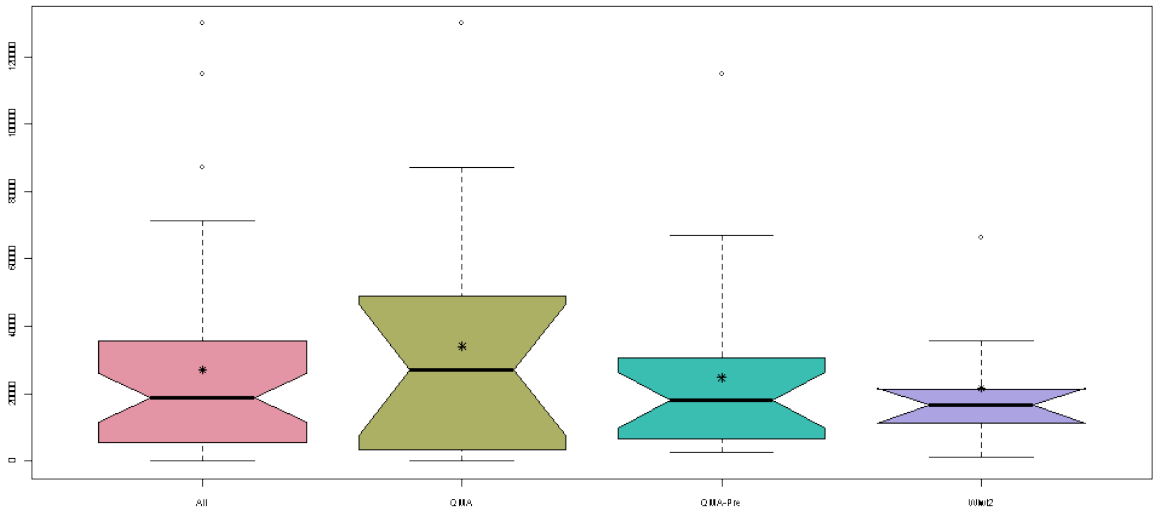


Abb. 6.6: Kennzahlen zur Gesamtdauer (s) in Fallbearbeitungen zu allen (All) sowie einzelnen Prüfungen in Box-Plots

Was daraus auch hervorgeht: Die Werte der Gesamtdauer in einzelnen Prüfungen haben sich stark unterschieden; dies erwarten wir auch für andere Kennzahlen. Wenn ausreichend Daten vorhanden sind, um einzelne Prüfungen zu betrachten, ist es daher sinnvoll, die Ergebnisse zu normieren, um besonders aufwändige oder überdurchschnittlich gut bzw. schlecht ausgefallene Prüfungen zu berücksichtigen.

Die Analyse der Prüfungsleistungen kann langfristig sehr ergiebig sein. Ein Problem sind manuelle Schritte, die bisher zur Eingabe der Prüfungsergebnisse sowie der relevanten Übungsfälle ins Data-Warehouse nötig sind. Wenn diese Informationen nicht in unstrukturierter Form (z.B. als Excel-Dateien), sondern direkt in tabellarischer Form (z.B. als CSV-Datei) vorliegen, können die Schritte weitgehend automatisiert werden.

6.3.2.6 Kontinuierliche Fallbearbeitung

Bezüglich *Kontinuierlicher Fallbearbeitungen* haben wir folgende Kennzahlen verglichen:

- Gesamtdauer Durchschnitt unter allen Fallbearbeitungen: 554s
- Bearbeitungsdauer Durchschnitt unter allen Fallbearbeitungen: 525s
- Gesamtdauer Durchschnitt unter allen Fallbearbeitungen bei Prüfungen: 1430s
- Kontinuierliche Bearbeitung Dauer Durchschnitt unter allen Fallbearbeitungen im betrachteten Zeitraum: 610s

- Kontinuierliche Bearbeitung Dauer Durchschnitt unter nicht-abgebrochenen Fallbearbeitungen im betrachteten Zeitraum: 709s
- Kontinuierliche Bearbeitung Dauer Durchschnitt unter abgebrochenen Fallbearbeitungen im betrachteten Zeitraum: 366s
- Kontinuierliche Bearbeitung Dauer Durchschnitt unter Erstbearbeitungen im betrachteten Zeitraum: 679s
- Kontinuierliche Bearbeitung Dauer Durchschnitt unter Letztbearbeitungen im betrachteten Zeitraum: 520s

Momentan führt die *Kontinuierliche Fallbearbeitung* meist vom Anfang bis zum Ende einer Fallbearbeitung – noch zu selten wurde die Pausefunktion aufgerufen (siehe auch Abbildung 6.9). Für den beschriebenen Zeitraum, eine der Bearbeitungsspitzen, war die Kontinuierliche Bearbeitung sogar länger als die durchschnittliche Gesamtdauer aller Fallbearbeitungen. Auf Dauer würde es sich lohnen, die Pausefunktion dennoch näher zu betrachten, und folgende Frage zu beantworten: Nach Pausierung, wie häufig wird der Fall „von vorne“ gestartet, die Möglichkeit einer Wiederaufnahme also nicht genutzt?

Des Weiteren ist auffallend: Die größte Motivation bestand dann, wenn die Fälle auch in Prüfungen relevant waren. Lernende, die für eine Prüfung lernen, arbeiten länger an Fällen als die Allgemeinheit. Eine stärkere Integration der Methode Fallbasiertes Lernen in eine Veranstaltung – beispielsweise durch explizit relevante Übungsfälle – würde folglich mit Sicherheit zu höheren Beteiligungen führen.

Bestätigt werden konnten folgende Vermutungen: Abgebrochene Fallbearbeitungen sind deutlich kürzer. Lernende bearbeiten einen Fall beinahe nur halb so lang ohne Unterbrechung, wenn sie ihn abbrechen als wenn sie ihn beenden werden. Auch bei mehrfach wiederholten Bearbeitungen beschäftigen sie sich kürzer mit einem Fall, möglicherweise aufgrund des Lerneffekts.

Abbildungen 6.7 und 6.8 stellen die Verteilung der Dauer der Kontinuierlichen Fallbearbeitungen in Abhängigkeit der Abbruchinformation und Bearbeitungsnummer dar, wie in Tabelle 6.6 gefordert.

6.3.2.7 Fallbearbeitungsaktionen Frequenztabelle

In diesem und den nächsten beiden Abschnitten nun etwas genauer zur Technik von CaseTrain, wie in Abschnitt 6.2.4.3 gefordert.

Eine Betrachtung des Berichts (siehe Abbildung 6.9) hat nützliche Informationen ergeben. Die Introhilfe wurde insgesamt nur 190 Mal angeschaut – eine äußerst geringe Zahl bei mindestens 2.608 Lernenden. Möglicherweise ist das Feature zu unauffällig. Die Pausefunktion dagegen ist

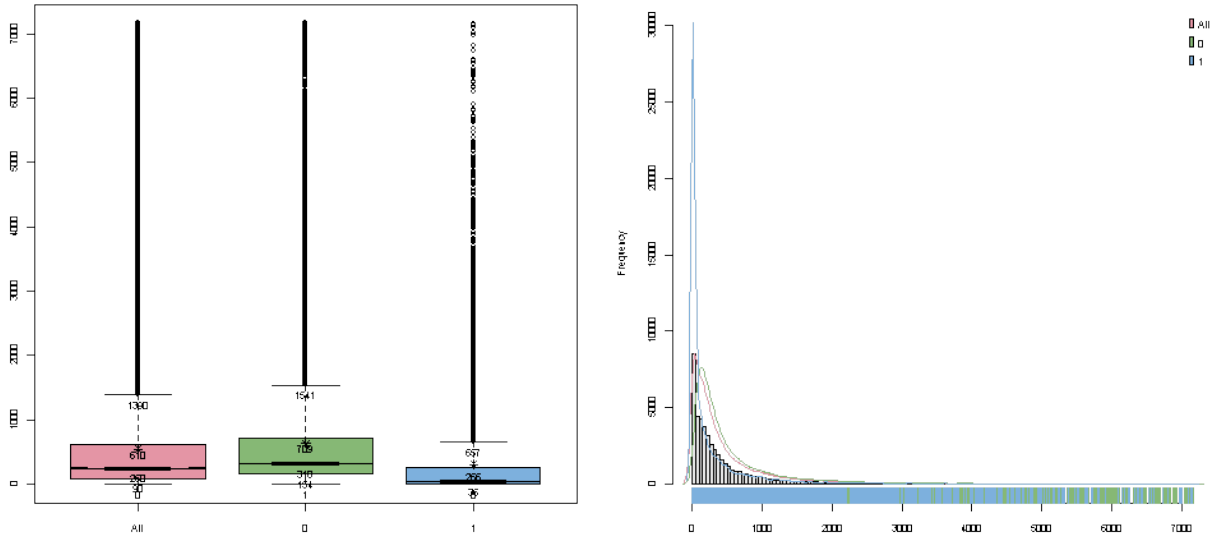


Abb. 6.7: Box-Plot und Verteilung der Bearbeitungsdauer (s) ohne Unterbrechung bei allen (All), beendeten (0) und abgebrochenen (1) Fallbearbeitungen

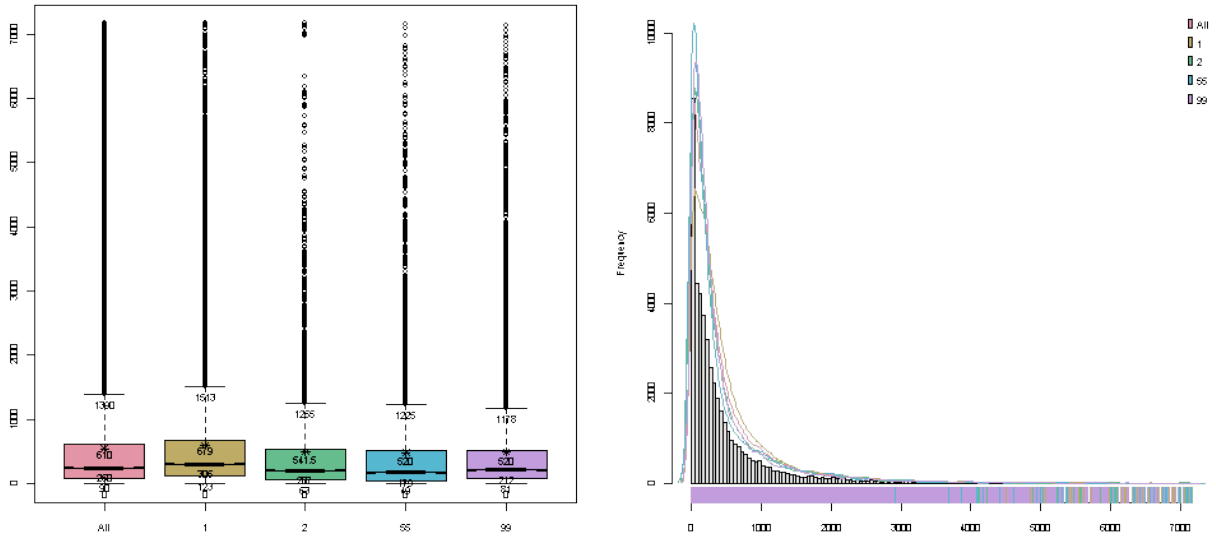


Abb. 6.8: Box-Plot und Verteilung der Bearbeitungsdauer (s) ohne Unterbrechung bei allen (All) Fallbearbeitungen, Erst- (1), Zweit-(2) und Letztbearbeitungen (99) sowie mittleren Fallbearbeitungen (55)

noch nicht lange als Feature vorhanden, was deren niedrige Zahl (2.405) erklärt. Darüber hinaus sind keine unerwarteten Häufigkeiten aufgetreten.

		Kennzahlen				
typ	status	Anzahl Fallbearbeitungen	fbak_dauer sum	aktionen count	Dauer Durchschnitt	Anzahl Durchschnitt
Bearbeitung	1	113.943	76.818,203	113.942	674,181	1
	2	113.943	2.291,570	2.406	20,112	0,021
	3	113.943				
Bild	1	113.943	675	155	0,006	0,001
	2	113.943	562,739	27.363	4,939	0,24
	3	113.943	5,384	128	0,047	0,001
Fallverlauf	1	113.943				
	2	113.943	1.130,513	34.473	9,922	0,303
	3	113.943	133,781	2.966	1,174	0,026
Hintergrundinfo	1	113.943				
	2	113.943		781		0,007
	3	113.943		15		0
Introinfo	1	113.943	3,965	187	0,035	0,002
	2	113.943	44	3	0	0
	3	113.943				
Link	1	113.943		28		0
	2	113.943		4.521		0,04
	3	113.943		43		0
Pause	1	113.943	0	7	0	0
	2	113.943	0	2.398	0	0,021
	3	113.943				

Abb. 6.9: Für jede Hilfsfunktion Gesamtdauer (s) und Häufigkeit der Nutzung, außerdem durchschnittliche Dauer und Häufigkeit pro Fallbearbeitung

6.3.2.8 Scoreaktionen Frequenztabelle

Auffällig ist, dass der Fragehinweis sehr lange gelesen wird. Selbst der Maximalwert von 12.497 Sekunden – über drei Stunden – ist nicht sicher ein Ausreißer. Der Lernende schreibt in seinem Feedbacktext "VIEL!!! zu lang... weniger tabellenwerte würden es fürs verständnis auch tun [sic]". Die Lernenden scheinen, nachdem sie den Fragehinweis erhalten haben, alles Nötige zu wissen und beginnen dann auf dem Papier mit ihrer Lösung, z.B. in Fällen zu Statistikveranstaltungen. Das Beantworten der Frage reduziert sich auf das reine Eingeben.

Abbildung 6.10 zeigt den relevanten Ausschnitt aus dem Bericht.

6.3.2.9 Lösungskommentar und Score

Der *Korrelationskoeffizient* zwischen den beiden Attributen beträgt -0.1032197 und ist daher unauffällig. Auch ein *Scatter-Plot* (siehe Abbildung 6.11) der beiden Attribute ergibt keinen sichtbaren Zusammenhang; übereinander liegende Werte werden darin durch einen *Jitter* leicht verschoben, um Anhäufungen sichtbar zu machen. Die Vermutung, dass Lernende, die einen niedrigen Score erhalten, den Lösungskommentar deutlich länger betrachten, lässt sich nicht bestätigen.

		Kennzahlen			
isAbbruch	nummerLernerFall	score count	score avg	fragehinweis avg	loesungskommentar avg
0	All nummerLernerFalls	812.643	0,67	61,676	40,329
	1	363.339	0,618	61,653	41,863
	2	72.025	0,684	93,305	45,374
	55	180.071	0,681	50,271	36,255
	99	197.208	0,751	59,001	38,338
1	All nummerLernerFalls	50.252	0,513	47,479	62,384
	1	22.380	0,465	56,319	72,866
	2	4.740	0,542	47,824	60,406
	55	12.855	0,549	30,6	51,481
	99	10.277	0,557	51,127	48,597

Abb. 6.10: Durchschnittliche Dauer (s) pro Anzeige des Fragehinweis und Lösungskommentars unter beendeten (0), abgebrochenen (1), sowie Erst-(1), Zweit-(2), Letzt-(99) und mittleren Fallbearbeitungen (55)

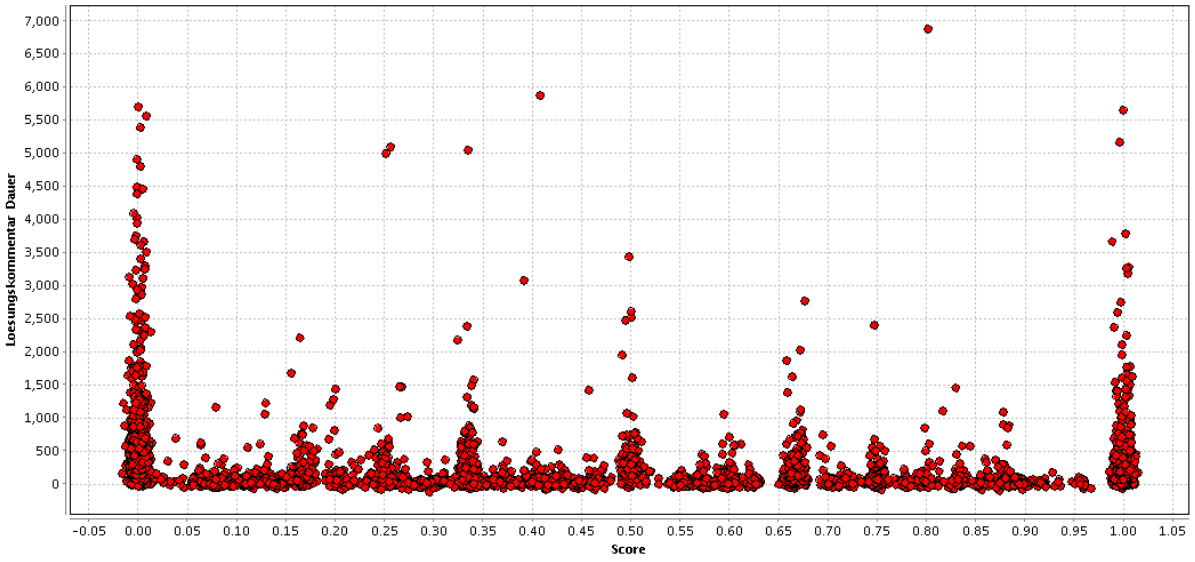


Abb. 6.11: Score (x-Achse) zu Dauer (s) der Anzeige des Lösungskommentars (y-Achse)

Ein Grund könnte die große Anzahl an Ausreißern sein. Obwohl wir Lösungskommentare, die länger als zwei Stunden angeschaut wurden, bereits bei der Erstellung des Berichts ausgeschlossen haben, zeigt ein *Box-Plot* (siehe Abbildung 6.12) die hohe Anzahl an Ausreißern. Diese Version eines Box-Plots kennzeichnet Werte als mögliche Ausreißer, die besonders stark vom Median abweichen – genauer, deren Abstand vom nächsten Quartil größer ist als 1,5 des Abstands vom Median zum Quartil. Das Diagramm zeigt demnach, dass eine neue Abfrage, die Anzeigen des Lösungskommentars bei einer Dauer über 623 Sekunden verwirft, sinnvoll wäre. Aber nachdem

ein weiterer Bericht mit dieser Schwelle erstellt worden ist, das gleiche Bild: Es gibt eine große Anzahl an Ausreißern. Möglicherweise wird der Lösungskommentar auch zu variabel verwendet, als dass der Score eine Aussage zur Dauer machen kann. Aussagekräftiger ist ggf. eine Betrachtung einzelner Fälle oder Fragen, wenn der Informationsgehalt von Lösungskommentaren zu stark schwankt.

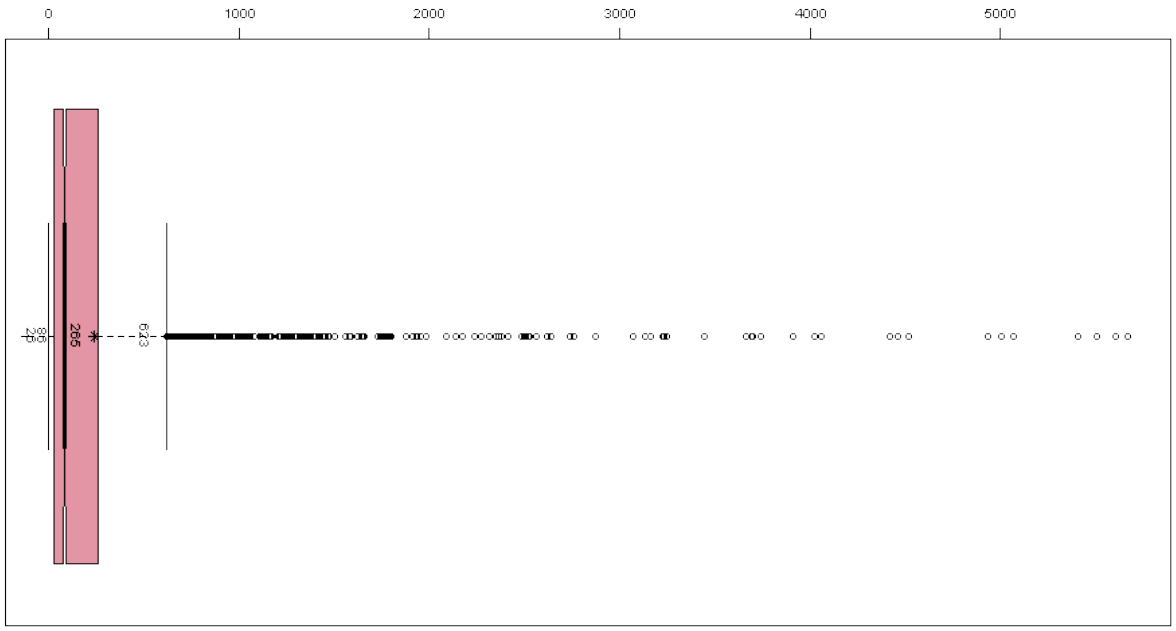


Abb. 6.12: Kennzahlen zur Dauer (s) der Anzeige des Lösungskommentars in einem Box-Plot

6.3.2.10 Autordauer

Nun werden die Anforderungen zur Bewertung der Qualität der Fälle beschrieben, wie laut Abschnitt 6.2.2.3 verlangt.

Für jede Fallversion besitzen wir die Information darüber, wie stark sich die durchschnittliche Bearbeitungsdauer von der vom Autor angegebenen Dauer unterscheidet.

Die durchschnittliche Bearbeitungsdauer von beendeten Fallbearbeitungen beträgt 11,57 Minuten, die durchschnittlich vom Autor angegebene Dauer beträgt 18,01 Minuten – eine Abweichung von insgesamt 6,44 Minuten, die jedoch vermutlich deshalb auftritt, da Autoren für kurze Fälle meist keine Einschätzung zur Dauer geben. Für einzelne Fälle kommen auch extreme Abweichungen vor, für einen Statistikfall haben 108 Fallbearbeitungen im Durchschnitt jeweils 104 Minuten gedauert, obwohl der Autor jeweils nur 30 Minuten veranschlagt hatte. Hier kann den Autoren geholfen werden, indem ihnen diese Information mitgeteilt wird – zumindest, wenn sie durch eine Mindestanzahl an Fallbearbeitungen gestützt wird.

6.3.3 Ausblick

Letztendlich lassen sich die bisherigen Ergebnisse als Potenzial erkennen. Sie geben häufig Hinweise auf Punkte, die es lohnt, in einem fortführenden Data-Mining-Projekt zu untersuchen.

Im Rahmen der Analysen wurden geeignete Werkzeuge ausgewählt und ein Data-Warehouse entwickelt. Die Ergebnisse und Erfahrungen auch langfristig zu nutzen, ist aus drei Gründen zu empfehlen:

Wir haben das Data-Warehouse so entworfen, dass ohne großen Aufwand aktuelle Daten eingelesen werden können. Für die Logdaten besteht über eine Datenbank eine fest-definierte Schnittstelle; deshalb ließe sich das Einlesen bereits zum jetzigen Zeitpunkt bis *auf Knopfdruck* automatisieren. Die Anforderungen aus diesem Projekt können somit für neue Daten in weitaus kürzerer Zeit umgesetzt werden. Die regelmäßige Evaluation von CaseTrain ist mit weniger Aufwand möglich.

Zudem kann das Data-Warehouse relativ einfach erweitert werden. Wir haben für die Daten ein verständliches Modell entwickelt, in das neue Elemente eingefügt werden können – beispielsweise externes Hintergrundwissen. Der Behandlung neuer Anforderungen werden keine Grenzen gesetzt. Das Modell stützt sich auf etablierte Konzepte, und kann auch von Außenstehenden nachvollzogen werden.

Außerdem stützt sich das Projekt ausschließlich auf Open-Source-Werkzeuge. Diese sind frei verwendbar und können bei Bedarf selbständig erweitert werden. Für jedes der verwendeten Werkzeuge sind auch alternative Open-Source-Systeme am Markt vorhanden.

6.4 Data-Assay

Im Folgenden wird beschrieben, wie Informationen zu den Daten erhalten wurden, um die Anforderungen aus dem Business-Case zu behandeln. Dieses Prüfen, Beschreiben und Vorbereiten der Daten im Data-Assay hat insgesamt ca. 100 Stunden in Anspruch genommen.

6.4.1 Logdaten - Beschreibung ohne Vorverarbeitung

Zu Beginn des Projekts haben wir die aktuellen Logdaten (vgl. Abschnitt 6.2.3.1) aus der operationalen Datenbank des Lehrsystems aus- und in eine Datei „statsdump.txt“ eingelesen. Die Größe dieser Datei hat 469MB betragen. Enthalten sein sollten Logdaten vom Wintersemester 2007 bis August 2009. Diesen *SQL-Dump* haben wir als Tabelle „logs“ in eine Datenbank „casetrain“ auf den *Data-Warehouse-Server* importiert.

Tabelle „logs“ enthält 17 Attribute und 1.201.310 Instanzen. In Tabelle 6.13 werden Attribute näher beschrieben, die für die weitere Bearbeitung relevant waren.

Name:	**Beschreibung**	**Datentyp**	**Bemerkungen**
userid	Lernender	String(64)	-
mode	Feedbackmodus	String(32)	Mögliche Werte: null, exam, 0, 1, default, osce
sessionid	Fallbearbeitungs-ID	String (32)	Eindeutiger Schlüssel für eine Fallbearbeitung. Es gibt 169.386 verschiedene Werte.
rnumber	Resume-Number	Integer (5)	Wert meist *null*. Ansonsten Werte von „0“ bis „14“.
caseid	Fallid	String (128)	-
casev	Fallversion	String (256)	-
gettime	Clientzeit	Datetime	Ältester Wert ist „0000-02-04 21:09:09“, jüngster Wert ist „9200-02-05 11:05:30“.
gettimeServer	Serverzeit	Datetime	Ältester Wert ist „0027-07-07 16:07:37“, zweitältester Wert ist „2007-11-15 17:27:40“, jüngster Wert ist „2009-08-06 15:00:11“.
log	Log-String	String	-

Tab. 6.13: Beschreibung relevante Logdatenattribute

Wenn das Attribut „userid“ leer, *null* oder „0“ ist, ist der Lernende anonym und hat den Fall gestartet, ohne sich vorher mit einem Benutzernamen anzumelden. Es gibt mindestens 2.608 Studenten, die CaseTrain verwendet haben, denn es findet sich in den Logdaten diese Anzahl an Lernenden mit einer *S-Kennung* des Universitätsrechenzentrums, einer „userid“, die mit „s“ beginnt und einer Zahl endet. Dies ließ sich über eine einfache *SQL*-Abfrage unter Verwendung eines *Regulären Ausdrucks* durch den Operator „REGEXP“ feststellen.

Jeder Fall erhält vom Fallautor eine Version, wenn sie veröffentlicht wird. Die Attribute „caseid“ und „casev“ identifizieren somit eindeutig eine konkrete *Fallversion*.

Die Attribute „gettime“ und „gettimeServer“ geben die Zeit an, zu der ein Log an den Server geschickt worden ist, also eine neue Instanz in der Tabelle „logs“ erzeugt wurde. Ersteres nennt die Zeit, die der Lernende auf seinem Rechner eingestellt hat; sie ist häufig völlig unrealistisch. Bei „gettimeServer“ handelt es sich um die Zeit des Servers. Bis auf einer unsinnigen Zeitangabe scheinen die Werte korrekt zu sein.

Wenn ein Lernender während einer Fallbearbeitung eine Pause durchführt und den Fall anschließend wieder aufnimmt, werden neue Loginstanzen mit einer um eins hochgezählten Resume-Number erstellt. Die Pausefunktion ist noch nicht lange integriert, weshalb „rnumber“ meist *null* ist. Eine *Kontinuierliche Fallbearbeitung* besitzt eindeutige Werte der Attribute „sessionid“ und „rnumber“.

6.4.2 Logdaten - Beschreibung mit Vorverarbeitung

Für eine nähere Beschreibung der Tabelle mussten wir die Rohdaten vorverarbeiten. Wir haben mehrere *Transformationen* mit *Pentaho Data Integration* erstellt und in einem *Job* zusammengefasst. Bei Ausführung leert dieser *Job* zunächst die betroffenen Tabellen, so dass die Daten bei einer wiederholten Ausführung stets vollständig neu eingelesen werden.

Für jede *Kontinuierliche Fallbearbeitung* – eindeutige Werte für „sessionid“ und „rnumber“ – werden in gewissen Abständen neue Instanzen in Tabelle „logs“ erzeugt (durchschnittlich sieben). Dabei enthält ein neuer Eintrag stets die gesamte Information der Vorgänger, welche von dem Zeitpunkt an redundant vorliegen.

Wenn sich während einer *Kontinuierlichen Fallbearbeitung* der Lernende, der Fall oder der Feedbackmodus ändert, handelt es sich um eine ungültige Fallbearbeitung, die verworfen werden kann. Bei dieser Gruppe an Instanzen mit identischer „sessionid“ und „rnumber“ weisen Instanzen verschiedene Werte für „userid“, „caseid“ oder „mode“ auf; mit einer einfachen SQL-Abfrage konnten wir einige wenige solcher Fallbearbeitungen identifizieren und filtern.

Unsere Analyse hat sich auf die Verwendung von CaseTrain zur Vorbereitung auf Prüfungen bezogen. Daher haben wir Fallbearbeitungen verworfen, die zu anderen Zwecken durchgeführt worden waren, z.B. zur Durchführung von Prüfungen; dies war der Fall wenn das Attribut „mode“ weder den Wert *null* noch „default“ hatte.

In dem Zusammenhang wurde auch überlegt, Fallbearbeitungen mit CaseTrain-Autoren oder -Entwicklern als Lernende zu filtern. Davon wurde aber abgesehen, auf Grund einer verhältnismäßig kleinen Anzahl solcher Fallbearbeitungen.

Bei neuen Instanzen einer Kontinuierlichen Fallbearbeitung wird im Grunde lediglich eine weitere Zeichenkette an den Logstring angehängt. Um für jede Kontinuierliche Fallbearbeitung einen einzigen, und zwar den vollständigsten Eintrag aus der Tabelle „logs“ zu erhalten, haben wir eine SQL-Abfrage, siehe Quellcode 6.1 erstellt, die für jede Gruppe an Einträgen mit der selben „sessionid“ und „rnumber“ nur den Eintrag ausgibt, der den längsten Logstring besitzt.

```
1 select logs.sessionid, logs.rnumber, max(log)
2 from logs
3 group by sessionid, rnumber
4 order by sessionid, rnumber
```

Quellcode 6.1: Filtern redundanter Logeinträge

Die ursprüngliche Zahl von 1.201.310 Instanzen wurde auf 172.638, jeweils eindeutig in den Werten von „sessionid“ und „rnumber“, reduziert.

Der Logstring hat eine besondere Betrachtung erfordert. Er enthält den Großteil der Loginformation, kodiert als eine theoretisch unbegrenzt lange Zeichenkette, sog. *String*; ein typisches Beispiel zeigt Quellcode 6.2.

```
1 |||A|2|A|||A|5|W1|||F1C|37|7|9||0|||A|45|K1F1|||A|53|K0F1|||A|54|W2|||F2C|67|2||0.5|||
2 A|70|W3|||F3C|85|1|2|4||1|||A|88|W4|||F4C|100|1|3||1|||A|102|W5|||F5C|111|3||1
3 |||A|113|W6|||F6C|127|1|3|4||1|||A|128|W7|||F7C|137|1||1|||A|138|W8|||
4 F8C|146|1||1|||G|NaN||0.812250|||A|147|WA
```

Quellcode 6.2: Beispiel für einen Logstring

Im Logstring sind einzelne Logevents kodiert. Dieses Format ist nicht dafür geeignet, um direkt die Fragestellungen zu behandeln, da von dem gesamten Logstring nicht auf Eigenschaften der enthaltenen Events geschlossen werden kann. Daher hat unsere Aufgabe zunächst darin bestanden, für jede Kontinuierliche Fallbearbeitung den Logstring so aufzubereiten, dass er anschließend leicht weiterverarbeitet werden konnte – was beinhaltet hat, ihn in seine Einzelteile zu zerlegen und die Logevents zu extrahieren.

Innerhalb des Logstrings werden die Informationen zu einzelnen Logevents durch die Zeichenkette „|||“, drei sog. *Pipes*, getrennt. Diese Pipes haben wir in ein einzelnes Trennzeichen umgewandelt – dabei mussten wir darauf achten, ein Trennzeichen zu verwenden, das im String vorher nicht vorkam –, um mit dem *Step* „Split field to row“ aus Pentaho Data Integration die Logeventstrings aus dem Logstring zu extrahieren und ihre Reihenfolge in einem neuen Attribut „folge“ zu beschreiben; Tabelle 6.14 zeigt die Struktur der Ausgabe an einem Beispiel.

sessionid	**rnumber**	**event**	**folge**
0000163551542659	0	A|2|A	1
0000163551542659	0	A|5|W1	2
0000163551542659	0	F1C|37|7|9||0	3

Tab. 6.14: Extraktion der Events

Die Zeichnkette im Attribut „event“ besitzt wiederum eine feste Struktur: Getrennt durch jeweils ein Pipe wird zunächst der Eventtyp, dann die relative Eventzeit in Sekunden nach Beginn der Kontinuierlichen Fallbearbeitung und schließlich ein Eventlog, eine weitere Zeichenkette, genannt. Der Step „Regex evaluation“ aus Pentaho Data Integration hat es ermöglicht, einen

Regulären Ausdruck zu definieren, um diese einzelnen Teile zu extrahieren. Quellcode 6.3 zeigt den verwendeten Ausdruck.

```
1 ^[\|]*([^\|]*)\|([^\|]*)[\|]?(.*)$
```

Quellcode 6.3: Regulärer Ausdruck zur Extraktion der Inhalte im String „event“

Tabelle 6.15 zeigt die Struktur der Ausgabe an einem Beispiel.

sessionid	**rnumber**	**evrelfolge**	**evtype**	**evreltime**	**evlog**
0000163551542659	0	0	A	2	A
0000163551542659	0	0	A	5	W1
0000163551542659	0	0	F1C	37	7|9||0

Tab. 6.15: Extraktion der Inhalte im String „event“

Bevor die Inhalte aus den Logevents weiter analysiert werden, mussten wir das Konzept der Pause in CaseTrain näher betrachten; so gibt es zwei verschiedene Möglichkeiten, um die Fallbearbeitung zu pausieren und zu einem späteren Zeitpunkt fortzusetzen:

Explizite Pause Die ausdrückliche Betätigung über den Schaltknopf „Pause“. In diesem Fall wird ein Pauseevent ausgelöst.

Implizite Pause Die automatische Betätigung beim Schließen des Browserfensters. In diesem Fall wird kein Pauseevent ausgelöst.

Laut der Beschreibung der Anforderung aus Tabelle 6.6 sollen diese beiden Möglichkeiten nicht unterschieden werden. Deshalb sollen auch für Implizite Pausen Events erstellt werden. Die Erstellung der eigenen Pauseevents geschieht dabei in mehreren Schritten:

1. Es werden zunächst jegliche Events der *Expliziten Pausen* mit einer einfachen SQL-Abfrage aus den Events entfernt.
2. Für jede Fallbearbeitungs-ID und Resume-Number wird „evreltime“ und „evrelfolge“ des letzten Events mit einer einfachen SQL-Abfrage ermittelt und in den Attributen „uptime“ und „upfolge“ gespeichert.
3. Für jede Fallbearbeitungs-ID und Resume-Number zu denen es eine weiterführende Resume-Number gibt (es ist also nicht die letzte Resume-Number), wird ein Pauseevent erstellt. Dieses Pauseevent enthält neben der Fallbearbeitungs-ID und Resume-Number, den Wert aus „upfolge“ im Attribut „evrelfolge“, den konstanten Wert „N“ im Attribut „evtype“, „uptime“ im Attribut „evreltime“ und die Konkatenation des Wertes „P“ und dem Attribut „uptime“ im Attribut „evlog“.
4. Diese Events werden abschließend den Events beigefügt.

Mit Ausnahme der Anforderung zu *Kontinuierliche Fallbearbeitung* (siehe Tabelle 6.6) wird eine Fallbearbeitung stets als Ganzes – ggf. mit Pausen – betrachtet. Anstatt die Fallbearbeitungen aufgeschlüsselt nach einzelnen Kontinuierlichen Fallbearbeitungen zu behandeln, haben wir daher beschlossen, deren Events zueinander in Bezug zu setzen und gemeinsam zu betrachten.

Da Attribut „evtime“ die Anzahl an Sekunden nach Beginn der Kontinuierlichen Fallbearbeitung angibt, nach denen ein Event aufgetreten ist, handelt es sich hierbei um eine relative Angabe. Gleiches gilt für die Angabe der Eventfolge im Attribut „evfolge“. Um absolute Angaben zum Beginn der Fallbearbeitung zu erhalten, muss für jedes Event nicht nur die relative Zeit bzw. Folge der letzten Pause addiert werden, sondern auch die Zeiten bzw. Folgen der Pausen davor. Quellcode 6.4 zeigt den *SQL*-Ausdruck, der für jede Kontinuierliche Fallbearbeitung absolute Werte für „reltime“ und „relfolge“ angibt; diese haben wir mit dem *Step* „Calculator“ für jedes Event einer Kontinuierlichen Fallbearbeitung zur relativen Zeit und Folge addiert.

```
1 select time1.sessionid
2 , time1.rnumber
3 , sum(time2.uptime) as sumuptime
4 , sum(time2.upfolge) as sumupfolge
5 from rohdaten_logs_filtered_norm_nornr_manifest_table as time1
6 inner join rohdaten_logs_filtered_norm_nornr_manifest_table as time2 on
7 time1.sessionid = time2.sessionid and time2.rnumber < time1.rnumber
8 group by sessionid, rnumber
9 order by time1.sessionid, time1.rnumber
```

Quellcode 6.4: Berechnung absoluter Angaben

Zusammenfassend haben die Daten nach diesem Schritt folgendermaßen vorgelegen:

- Tabelle von Instanzen von Fallbearbeitungen mit Attributen zum Lernenden, Fall und Zeit der Fallbearbeitung
- Tabelle von Instanzen von Events, mit Attributen zur Fallbearbeitung, zur absoluten Zeit, zur absoluten Folge, zum Typ des Events und zum Logstring des Events – inklusive der neu erstellten Pauseevents.

Events stellen das wichtigste Konzept in der Analyse dar. Sie beschreiben, was innerhalb einer Fallbearbeitung geschehen ist und enthalten Informationen, die in Anforderungen der Lösung nötig sind – zu Evaluationen, Abbrüchen, Fallbearbeitungsaktionen und Scoreaktionen sowie Scores und Gesamtscores.

Aus der Menge an über 30 verschiedenen Events haben wir diejenigen ausgewählt, die zur Lösung beitragen konnten. Tabelle 6.16 zeigt eine Übersicht dieser relevanten Events und beschreibt den Aufbau der Strings in den Attributen „evtype“ und „evlog“.

In einer letzten Transformation wird jede Instanz aus der Tabelle der Events eingelesen und überprüft, ob sie einem der relevanten Events entspricht. Irrelevante Events werden verworfen,

Name	**evtype**	**evlog**
Score in Frage	Hintereinander „F“, ein positiver Integer und „C“ bzw. „E“	Liste von positiven Integers, getrennt durch Pipe bei „C“ bzw. ein Text bei „E“; nach zwei Pipes eine reale Zahl; Ggf. weitere Zahlen und Texte getrennt durch Pipes
Score in Info	Hintereinander „I“ und ein positiver Integer	Pipe und reale Zahl
Fragehinweis	„A“	Hintereinander „H“; „1“ bzw. „0“; „F“ und positives Integer
Lösungskommentar	„A“	Hintereinander „K“, „1“ bzw. „0“, „F“ und positives Integer
Introinfo	„A“	„I1“ bzw. „I0“
Bild	„A“	„M1“ bzw. „M0“
Fallverlauf	„A“	„Z1“ bzw. „Z0“
Hintergrundinfo	„A“	„H1“
Pause	„N“	Hintereinander „P“; positives Integer
Link	„A“	Hintereinander „L“; Pipe; beliebiger String
Beginn Fallbearbeitung	„A“	„W1“
Ende Fallbearbeitung	„A“	„WA“
Gesamtscore	„G“	Hintereinander Pipe und eine reale Zahl
Evaluationsnoten	„E1E“	reale Zahl; Pipe; reale Zahl; zwei Pipes; Reale Zahl
Feedbacktext	„E2E“	Hintereinander beliebige Anzahl an Zeichen; zwei Pipes; Reale Zahl

Tab. 6.16: Eine Übersicht relevanter Events

relevante Events werden zur Weiterverarbeitung gespeichert. Dazu haben wir ausgiebig den *Step* „Regex Evaluation“ genutzt. Er hat es uns erlaubt, die Events nicht nur auf relevante Typen in „evtype“ und Eventlogstrings in „evlog“ zu überprüfen, sondern in einer zweiten Verwendung den Inhalt dieser Attribute zu extrahieren. Um Ausnahmen zu berücksichtigen, waren teilweise aufwändige *Reguläre Ausdrücke* nötig. Beispielsweise waren in manchen Fällen die realen Zahlen in den Events „Score in Info“, „Score in Frage“, „Evaluationsnoten“ und „Feedbacktext“ durch ein Leerzeichen angeführt bzw. *null* als „NaN“, Not-a-Number, kodiert.

Im Folgenden wird erläutert, inwiefern die einzelnen Events verwendet werden.

6.4.2.1 Scoreevents

Während der Fallbearbeitung werden dem Lernenden nicht nur Fragen gestellt, sondern auch die Auswahl von Informationen überlassen – für beides erhalten sie Scores, ausgedrückt als Event „Score in Frage“ oder „Score in Info“. Uns haben diese Scores allgemein interessiert (siehe Tabelle 6.9), weshalb wir zwischen den beiden nicht unterschieden haben. Quellcode 6.5 zeigt den Regulären Ausdruck, den wir verwendet haben, um aus dem Event „Score aus Frage“ den Score zu extrahieren; die konkrete Antwort wird dabei ignoriert. Event „Score in Info“ ließ sich vergleichbar einlesen.

```
1 ^.*\|\|(\d+.?\d*)$
```

Quellcode 6.5: Regulärer Ausdruck zum Extrahieren eines Scores

Im Attribut „evtyp“ ist die Information enthalten, auf welche Frage oder Informationsauswahl sich der Scoreevent bezieht. Darüber ist eine Verbindung zu den Scoreaktionen „Fragehinweis“ und „Lösungskommentar“ möglich, wie sie nötig ist, um den Bericht aus Tabelle 6.9 zu erstellen. Deshalb werden sie extrahiert und in Attributen „fragInfo“ und „nummer“ gespeichert.

6.4.2.2 Scoreaktionen

Die Events „Fragehinweis“ und „Lösungskommentar“ entstehen, wenn der Lernende eines der gleichnamigen *Features* der Scoreaktionen ausführt, die in den Anforderungen aus den Tabellen 6.8 und 6.9 genannt werden.

Attribut „evlog“ enthält neben dem Typ, kodiert als Buchstabe, die Information darüber, ob es sich um eine Aktion handelt, die das Feature öffnet („1“) oder schließt („0“). Diese Informationen werden extrahiert, öffnende und schließende Events werden in getrennten Tabellen zwischengespeichert, um sie später zueinander in Beziehung zu setzen. Weiterhin wird die Nummer der Frage extrahiert, auf die sich der Event bezieht – als Teil eines Fremdschlüssels zu den *Scoreevents*.

6.4.2.3 Öffnende und schließende Fallbearbeitungsaktionen

Die Events „Introinfo“, „Bild“, „Fallverlauf“ oder „Hintergrundinfo“ beschreiben je nach Typ (gespeichert in einem Attribut „type“) gleichnamige Features der Fallbearbeitungsaktionen, die in den Tabellen 6.7 und 6.4 genannt werden. Auch bei diesen werden öffnende und schließende Events unterschieden und in separate Tabellen zwischengespeichert.

6.4.2.4 Pausen

In solchen Events wird die Fallbearbeitungsaktion „Pause“ sowie die relative Dauer seit dem Beginn der *Kontinuierlichen Fallbearbeitung* beschrieben, die wir selbst generiert haben. Diese werden in zwei Anforderungen in Abschnitt 6.2.4.3 genannt (siehe Tabellen 6.7 und 6.6).

6.4.2.5 Links

Diese Events beschreiben die Fallbearbeitungsaktionen „Link“, die in Anforderung aus Tabelle 6.7 genannt werden. Der konkrete Web-Link ist dabei nicht von Interesse.

6.4.2.6 Bearbeitungsstatus

Die Events „Beginn Fallbearbeitung“ und „Ende Fallbearbeitung“ nennen für jede Fallbearbeitung den Zeitpunkt des Beginns der Beantwortung der Fragen und den Zeitpunkt der Beendigung der Beantwortung der Fragen (bzw. Erhalt eines *Gesamtscores*, entspricht nicht dem Schließen eines Falls). Bei mehreren solchen Events für eine Fallbearbeitung gilt der minimale, also älteste bzw. erste Zeitpunkt des Beginns oder Endes, was wir durch den *Step* „Group by“ in *Pentaho Data Integration* erreicht haben. Die erhaltenen Attribute „mittetime“ als Übergangszeit zwischen Bearbeitungsstatus Anfang und Mitte (siehe Tabelle 6.7) und „endetime“ als Beginn des Bearbeitungsstatus Ende haben wir anschließend durch den *Step* „Row denormaliser“ zu einer Instanz pro Fallbearbeitung verbunden.

Des Weiteren beinhaltet der Event „Ende Fallbearbeitung“ die *Bearbeitungsdauer*, die in Anforderungen aus Tabellen 6.4 und 6.10 nötig ist. Außerdem ermöglicht er uns zwischen beendeten und abgebrochenen Fallbearbeitungen zu unterscheiden, wie es laut Tabelle 6.3 gefordert ist.

6.4.2.7 Gesamtscores

Der Gesamtscore einer beendeten Fallbearbeitung ist für die Anforderung aus Tabelle 6.4 nötig.

6.4.2.8 Evaluationsnoten und Feedbacktexte

Standard-Evaluationen, wie sie in den meisten Fällen vorliegen, besitzen eine feste Struktur – wie sie die Events „Evaluationsnoten“ und „Feedbacktext“ zeigen. Wir haben die Events auf diese Struktur überprüft, anhand der Meta-Informationen (siehe 6.2.3.2) können wir später feststellen, in welchen Fällen die Standard-Evaluation auch tatsächlich enthalten ist. Aus den Evaluationsevents haben wir Attribute „Fallnote“, „Bediennote“ und „Feedbacktext“ extrahiert, die für Anforderungen aus Tabellen 6.1 und 6.2 nötig sind.

6.4.3 Meta-Informationen

Die Meta-Informationen (vgl. 6.2.3.2) liegen in der Logdatendatenbank in der Tabelle „cases“ vor. Darin sind für jede Version eines Falls Zusatzinformationen enthalten. Mit einer einfachen Transformation mit *Pentaho Data Integration* haben wir diese Daten in eine Tabelle, „cases“, auf den Rechner des *Data-Warehouse* kopiert.

Diese Tabelle enthält 36 Attribute und 3.470 Instanzen. Von Relevanz sind davon vier Attribute. Einerseits das Attribute „eval“. Mit dem Wert „1“ gibt es an, dass ein Fall standardmäßig evaluiert wird. Dies wird für Anforderungen aus Tabelle 6.1 und 6.2 benötigt. Außerdem ist das Attribut „meta_duration“, das die *Autordauer* in Minuten angibt, von Bedeutung für die Anforderung aus Tabelle 6.10. Die Bezeichnung und Version der Fälle in den Attributen „meta_id“ und „meta_version“ sind der Fremdschlüssel für die Zuordnung zu den Fallbearbeitungen.

6.4.4 Prüfungsergebnisse

Die Prüfungsergebnisse (vgl. 6.2.3.3) jeder Prüfung liegen in einer *Microsoft Excel*-Datei vor. Sie bestehen für jeden Studenten aus einer Schulnote und dem Prüfungsergebnis, dem Verhältnis zwischen erreichter und möglicher Punktzahl. Die Excel-Datei besitzt eine unstrukturierte Form, daher blieb uns nichts anderes übrig, als für jede Prüfung die Matrikelnummer, die Prüfungsnote und das Prüfungsergebnis manuell mit *Microsoft Excel* zu entnehmen und zusammen mit dem bekannten Namen als Attribut „Prüfung“ und dem bekannten Prüfungsdatum als Attribut „Datum“ in einer *CSV*-Datei „Studenten_Pruefung_Uebungsfälle.csv“ zu speichern. Uns wurde weiterhin eine Datei „login.csv“ übergeben, die für jeden Prüfling die Matrikelnummer und den Benutzernamen in CaseTrain nennt. Diese Information war entscheidend, um die Prüfungsergebnisse eines Studenten mit seinen Fallbearbeitungen in CaseTrain in Beziehung zu setzen. Jede Instanz aus „Studenten_Pruefung_Uebungsfälle.csv“ haben wir in einer *Transformation* mit *Pentaho Data Integration* mittels der Matrikelnummer als Fremdschlüssel um die Attributwerte einer Instanz aus „login.csv“ ergänzt und in eine Tabelle „rohdaten_pruefungsleistung“ eingegeben.

Wir besitzen das Hintergrundwissen darüber, welche Fälle für eine Prüfung relevant sind. Auch dieses Wissen haben wir in einer Datei „Pruefung_Uebungsfaelle.csv“ in Tabellarische Form gebracht. Jede Instanz beschreibt den bekannten Namen und das bekannte Datum der Prüfung sowie einen relevanten Fall. Diese CSV-Datei haben wir anschließend in einer Tabelle „rohdaten_uebungsfaelle“ gespeichert.

Um Tabelle 6.5 zu realisieren, muss herausgefunden werden können, für welche Fälle und Lernende aus CaseTrain wir Prüfungsergebnisse besitzen. Deshalb haben wir eine *SQL*-Abfrage erstellt, die die Tabellen „rohdaten_pruefungsleistung“ und „rohdaten_uebungsfaelle“ über den

Fremdschlüssel „Prüfung“ verbindet und Fall, CaseTrain-Benutzername, Prüfung, Datum, Note und Ergebnis als Attribute in einer Tabelle „rohdaten_pruefungen“ speichert.

6.5 Data-Warehouse

Ein weiteres Ziel bestand darin, mit Unterstützung der Informationen und Vorbereitungen aus dem Data-Assay, die Daten in fest-definierte Strukturen zu bringen, um eine Erstellung der Berichte über den Rechner des *Data-Warehouse* zu ermöglichen. Die Konzeption und Umsetzung solcher Strukturen im Data-Warehouse hat insgesamt ca. 40 Stunden benötigt.

Wir haben in diesem Projekt die Werkzeuge *MySQL Server*, *Mondrian OLAP Server* und *Pentaho BI-Server* sowie einige Verwaltungswerkzeuge, z.B. *MySQL Workbench* und *Pentaho Schema Workbench* verwendet.

6.5.1 ER-Modell

Aus den bisher erhaltenen Informationen haben wir ein *Entity-Relationship-Modell* entwickelt, das die Objekte und Attribute enthält, die in den Anforderungen aus dem Business-Case verlangt sind.

In Abbildung 6.13 wird das ER-Modells als Klassendiagramm dargestellt.

Im Folgenden werden die enthaltenen Objekte beschrieben. Es werden insbesondere abgeleitete Attribute erklärt; Nicht-abgeleitete Attribute haben wir direkt aus bestehenden Attributen übernommen und wurden bereits zu einem früheren Zeitpunkt beschrieben.

6.5.1.1 Fallbearbeitung

Eine Fallbearbeitung besitzt eine Zeit, zu der sie gestartet wurde. Für die Anforderung aus Tabelle 6.2 interessiert nur das Startdatum, ohne genaue Uhrzeit. Wir haben sie jeweils aus dem Attribut „gettimeServer“ der ersten *Kontinuierlichen Fallbearbeitung* einer Fallbearbeitung erhalten – mittels Funktionen „min“ und „group by“ in einer einfachen Abfrage mit *SQL*. Dieses Attribut ist z.B. für die Tabellen 6.11 und 6.12 notwendig.

Die Gesamtdauer gibt den Zeitpunkt des Abschließens oder Abbruchs einer Fallbearbeitung an und entspricht dem größten Wert des Attributs „evtime“ aller relevanten Events einer Fallbearbeitung. Falls kein Event empfangen wurde, ist dieser Wert „0“. Die Bearbeitungsdauer entspricht entweder dem Wert des Attributs „endetime“ oder, falls dieser *null* ist, der Gesamtdauer. Gesamtdauer und Bearbeitungsdauer werden laut Anforderung aus Tabelle 6.5 benötigt.

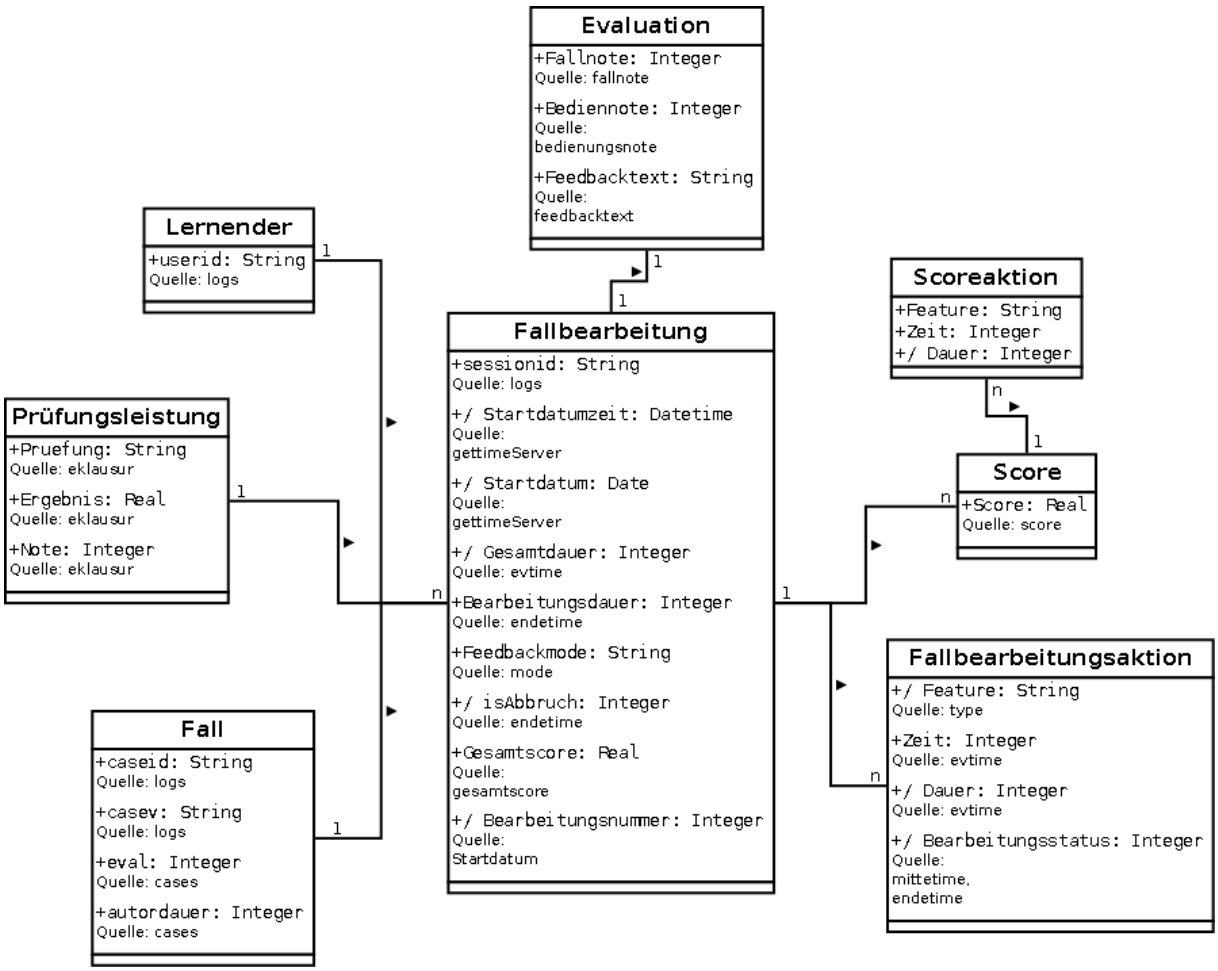


Abb. 6.13: CaseTrain ER-Modell

Das Attribut „isAbbruch“ gibt mit dem Wert „0“ an, wenn kein Abbruch, mit dem Wert „1“, wenn ein Abbruch vorliegt. Letzteres ist dann der Fall, wenn das Attribut „endetime“ den Wert *null* besitzt. Wir haben nicht den Gesamtscore als Kriterium verwendet, weil es Fallbearbeitungen gibt, die für den Gesamtscore *null* erhalten, allerdings korrekt beendet worden sind. Dieses Attribut ist z.B. für Tabelle 6.3 nötig.

Die *Bearbeitungsnummer*, wie sie in den Tabellen 6.8 und 6.6 gefordert wird, wird mit Hilfe des Startdatums jeder Fallbearbeitung erhalten. Da die *Bearbeitungsnummer* eine Reihenfolge ausdrückt, wird für die Kodierung ein Integer verwendet: Erstbearbeitungen (1), Zweitbearbeitungen (2), mittlere Bearbeitungen (55) und Letztbearbeitungen (99). Für die Umsetzung haben wir die Möglichkeit von *SQL* verwendet, zwei Datumswerte mittels Operator „=“ zu vergleichen.

Zu einer Fallbearbeitung gehören ein Lernender, ein Fall, eine Evaluation und eine Prüfungsleistung; außerdem eine beliebige Anzahl an Scores und Fallbearbeitungsaktionen. Sie werden in den nächsten Abschnitten beschrieben.

Die Verbindung zu diesen Objekten wird mittels *Fremdschlüsseln* umgesetzt. Für den Lernenden besteht der Fremdschlüssel aus dem Attribut „userid“. Für den Fall besteht der Fremdschlüssel aus den Attributen „caseid“ und „casev“. Die Verbindung zur Evaluation, zu den Scores und zu den Fallbearbeitungsaktionen erfolgt über das Attribut „sessionid“; dieses stellt gleichzeitig den *Primärschlüssel* der Fallbearbeitungstabelle dar. Auch für die Prüfungsleistung besteht der Fremdschlüssel „sessionid“.

6.5.1.2 Lernender

Hier wird das Attribut „userid“ fast unverändert übernommen; lediglich bei Fallbearbeitungen, bei denen es *null*, leer oder „0“ ist, erhält es den Wert „anonym“. Das Attribut „userid“ stellt auch den *Primärschlüssel* dar, der zum *Fremdschlüssel* aus der Tabelle der Fallbearbeitungen korrespondiert.

6.5.1.3 Fall

Die Attribute „caseid“ und „casev“ aus der Tabelle „logs“ stellen einerseits den *Primärschlüssel* der Falltabelle und die Verbindung zum *Fremdschlüssel* der Fallbearbeitungen dar. Andererseits können mit ihnen für jeden Fall die Attribute „eval“ und „autordauer“ aus der Tabelle „cases“ ausgelesen werden. Allerdings hat sich herausgestellt, dass solche Meta-Informationen nicht für alle Fälle vorhanden sind; dann sind „eval“ und „autordauer“ *null*.

6.5.1.4 Evaluation

Jede Evaluation besteht aus Fallnote, Bediennote und Feedbacktext, die jeweils auch *null* sein können. Genauso auf *null* gesetzt werden Evaluationen von Fallversionen, die nicht standardmäßig evaluiert werden, deren Wert „eval“ also nicht „1“ entspricht. Der *Primärschlüssel* besteht aus dem Attribut „sessionid“, das auch die Verbindung zum Primärschlüssel „sessionid“ aus der Tabelle der Fallbearbeitungen herstellt.

6.5.1.5 Fallbearbeitungsaktion

Hier werden die extrahierten Events „Bild“, „Fallverlauf“, „Hintergrundinfo“, „Introinfo“, „Pause“ und „Link“ zu Fallbearbeitungsaktionen vereinheitlicht.

Aus dem Event „Pause“ wird neben der Fallbearbeitungsaktion „Pause“ eine weitere Aktion abgeleitet: die „Bearbeitung“. Im Gegensatz zur Pause, die den Zeitpunkt einer Unterbrechung der Fallbearbeitung angibt und die Dauer 0 besitzt, stellt die Bearbeitung den Startzeitpunkt einer *Kontinuierlichen Fallbearbeitung* dar und gibt deren Dauer bis zum Abschließen, Abbruch

oder einer Pause der Fallbearbeitung an. Die Information über Kontinuierliche Fallbearbeitungen wurde im Data-Assay zeitweise ignoriert, um eine globale Sicht auf Fallbearbeitungen zu erhalten. Dennoch ist es nötig, diese Informationen zu erhalten, um die Anforderung aus Tabelle 6.6 zu bearbeiten; mittels eines *SQL*-Ausdrucks haben wir für jede Fallbearbeitung aus den Zeiten zwischen dem Start oder der Wiederaufnahme einer Fallbearbeitung und entweder einer Pause, dem Abschluss oder dem Abbruch der Fallbearbeitung eine Instanz „Bearbeitung“ erstellt.

In Abhängigkeit des Attributs „type“ des Event erhält das Attribut „Feature“ den Wert „Bild“, „Fallverlauf“, „Hintergrundinfo“, „Introinfo“, „Pause“, „Link“ oder „Bearbeitung“.

Das Attribut „Zeit“ entspricht dem Zeitpunkt, an dem die Aktion begonnen wurde, dem Wert des Attributs „evtime“ des Event bzw. des öffnenden Event bei öffnenden und schließenden Fallbearbeitungsaktionen.

Neben der „Pause“ gibt es auch für „Link“ und „Hintergrundinfo“ jeweils nur ein Event, weshalb die Dauer den Wert „0“ erhält.

Für „Bild“, „Fallverlauf“ und „Introinfo“ gibt es öffnende und schließende Aktionen. Um daraus die Nutzungsdauer zu berechnen, wurde für jeden öffnenden Event eines Typs die Zeit des nächsten schließenden Event desselben Typs bestimmt. Beide Events werden zu einer Fallbearbeitungsaktion zur Zeit des öffnenden Event zusammengefasst, die Differenz zwischen der Zeit des öffnenden und des schließenden Event ergibt die Dauer. Quellcode 6.6 zeigt den verwendeten *SQL*-Ausdruck zur Kombination von öffnenden und schließenden Aktionen.

```
1  select
2  opent.sessionid
3  , opent.evtime as opentime
4  , opent.type
5  , opent.evfolge as ordering
6  , min(closet.evtime) as closetime
7  , (min(closet.evtime) - opent.evtime) as dauer
8  from rohdaten_logs_filtered_norm_nornr_event_fbaktion_open as opent left join
9  rohdaten_logs_filtered_norm_nornr_event_fbaktion_close as closet on
10 opent.sessionid = closet.sessionid and
11 opent.type = closet.type and
12 opent.evfolge < closet.evfolge left join
13 rohdaten_logs_filtered_norm_nornr_event_fallb_zeit as zeitt on
14 opent.sessionid = zeitt.sessionid
15 where not exists(
16 select sessionid
17 from ermodell_logs_sessions_ung_manifest_table
18 where sessionid = opent.sessionid
19 )
20 group by opent.sessionid, opent.evfolge
```

Quellcode 6.6: SQL-Ausdruck zum Errechnen der Dauer

Der „Bearbeitungsstatus“ ist für die Tabellen 6.4 und 6.7 von Bedeutung. Um ihn zu bestimmen, haben wir den Aktionszeitpunkt mit den Attributen „mittetime“ und „endetime“ verglichen, die wir in Abschnitt 6.4.2.6 aus den Events „Beginn Fallbearbeitung“ und „Ende Fallbearbeitung“ erhalten. Dafür haben wir den *Step* „Formula“ aus *Pentaho Data Integration* mit einer Formel verwendet, die in Quellcode 6.7 dargestellt wird. Wir haben das Attribut „Bearbeitungsstatus“ als *Integer* kodiert, da es eine Reihenfolge angibt: Eine Fallbearbeitungsaktion kann demnach am Anfang (1), in der Mitte (2) und am Ende (3) auftreten.

```
1 if(
2   isblank([mittetime]);1;if (
3     ([mittetime]-[opentime]) > 0;1;if (
4       isblank([endetime]);2;if(([opentime]-[endetime]) > 0;3;2)
5     )
6   )
7 )
```

Quellcode 6.7: Formel zum Bestimmen des Bearbeitungsstatus

Eine Fallbearbeitungsaktion besitzt zusätzlich noch den *Fremdschlüssel* „sessionid“, über den die Verbindung zur zugehörigen Fallbearbeitung geschehen kann.

6.5.1.6 Score

Hier werden die Scoreevents zusammengefasst. Wenn das Attribut „score“ des Event kleiner als Null bzw. größer als Eins war, hat es sich um einen ungültigen Score gehandelt und das Attribut wurde auf *null* gesetzt.

Es kommt einige wenige Male vor, dass während einer Fallbearbeitung eine Frage mehrmals beantwortet und ein Score vergeben wird; dies darf in unseren Fällen, die keine Prüfungen beschreiben, nicht sein, weshalb diese Fallbearbeitungen verworfen werden.

Der *Fremdschlüssel* zur Fallbearbeitung besteht aus der „sessionid“. Daneben besitzt ein Score eine beliebige Anzahl an Scoreaktionen – ein *Fremdschlüssel*, bestehend aus den Attributen „sessionid“, „fragInfo“ und „nummer“, ermöglicht die Verbindung.

6.5.1.7 Scoreaktion

Vergleichbar mit einigen Typen von Fallbearbeitungsaktionen wurden Scoreevents unterschieden, die *Features* öffnen und schließen. Wir haben sie jeweils zu einer Scoreaktion mit Attributen „Zeit“ und „Dauer“ zusammengefasst. Das Attribut „Feature“ enthält je nach „type“ der Aktion den Wert „Fragehinweis“ oder „Lösungskommentar“. Die Verbindung zum Score geschieht über einen *Fremdschlüssel* aus den Attributen „sessionid”, „fragInfo“ und „nummer“.

6.5.1.8 Prüfungsleistung

Eine Prüfungsleistung entspricht der Leistung in der Prüfung, für die in einer Fallbearbeitung gelernt worden ist.

Mit einem *SQL*-Ausdruck (siehe Quellcode 6.8) haben wir für jede Fallbearbeitung eine Prüfungsleistung bestimmt. In einer Fallbearbeitung wird auf eine Prüfung gelernt, wenn für den Lernenden und bearbeiteten Fall ein Prüfungsergebnis vorliegt, dessen Zeitpunkt nach der Fallbearbeitung liegt. Wenn kein Ergebnis vorliegt, sind die Werte *null*. Wenn mehrere mögliche Prüfungsergebnisse vorliegen, wird dasjenige verwendet, das zeitlich am nähesten zur Fallbearbeitung liegt.

```
1  select ermodell_logs_session_user_case_table.sessionid
2  , startdatum as datum_fallb
3  , ermodell_logs_sessions_cases_manifest_table.caseid
4  , min(rohdaten_pruefungen.Datum) as min_pruef_datum
5  , count(rohdaten_pruefungen.Pruefung) as count_faelle
6  , rohdaten_pruefungen.Datum, rohdaten_pruefungen.Pruefung
7  , rohdaten_pruefungen.Note, rohdaten_pruefungen.Ergebnis
8  from ermodell_logs_session_user_case_table
9  left join ermodell_logs_sessions_cases_manifest_table
10 on ermodell_logs_sessions_cases_manifest_table.caseid =
       ermodell_logs_session_user_case_table.caseid
11 and ermodell_logs_sessions_cases_manifest_table.casev =
       ermodell_logs_session_user_case_table.casev
12 natural left join ermodell_logs_sessions_ende_start_manifest_table
13 join rohdaten_pruefungen
14 on ermodell_logs_sessions_cases_manifest_table.caseid = rohdaten_pruefungen.Fall
15 and startdatum < rohdaten_pruefungen.Datum
16 and ermodell_logs_session_user_case_table.user_key = rohdaten_pruefungen.LOGIN
17 group by ermodell_logs_session_user_case_table.sessionid, startdatum
18 having min(rohdaten_pruefungen.Datum) = rohdaten_pruefungen.Datum
```

Quellcode 6.8: SQL-Ausdruck für Prüfungsleistung

Anschließend haben wir das ER-Modell mittels *Pentaho Data Integration* so umgesetzt, wie es die Konzeption vorgibt.

6.5.2 MD-Modell

Wir haben drei *Data-Cubes*, jeweils mit Sicht auf Fallbearbeitungen, Fallbearbeitungsaktionen und Scores modelliert. Die Data-Cubes werden in einem Klassendiagramm, siehe Abbildung 6.14, dargestellt.

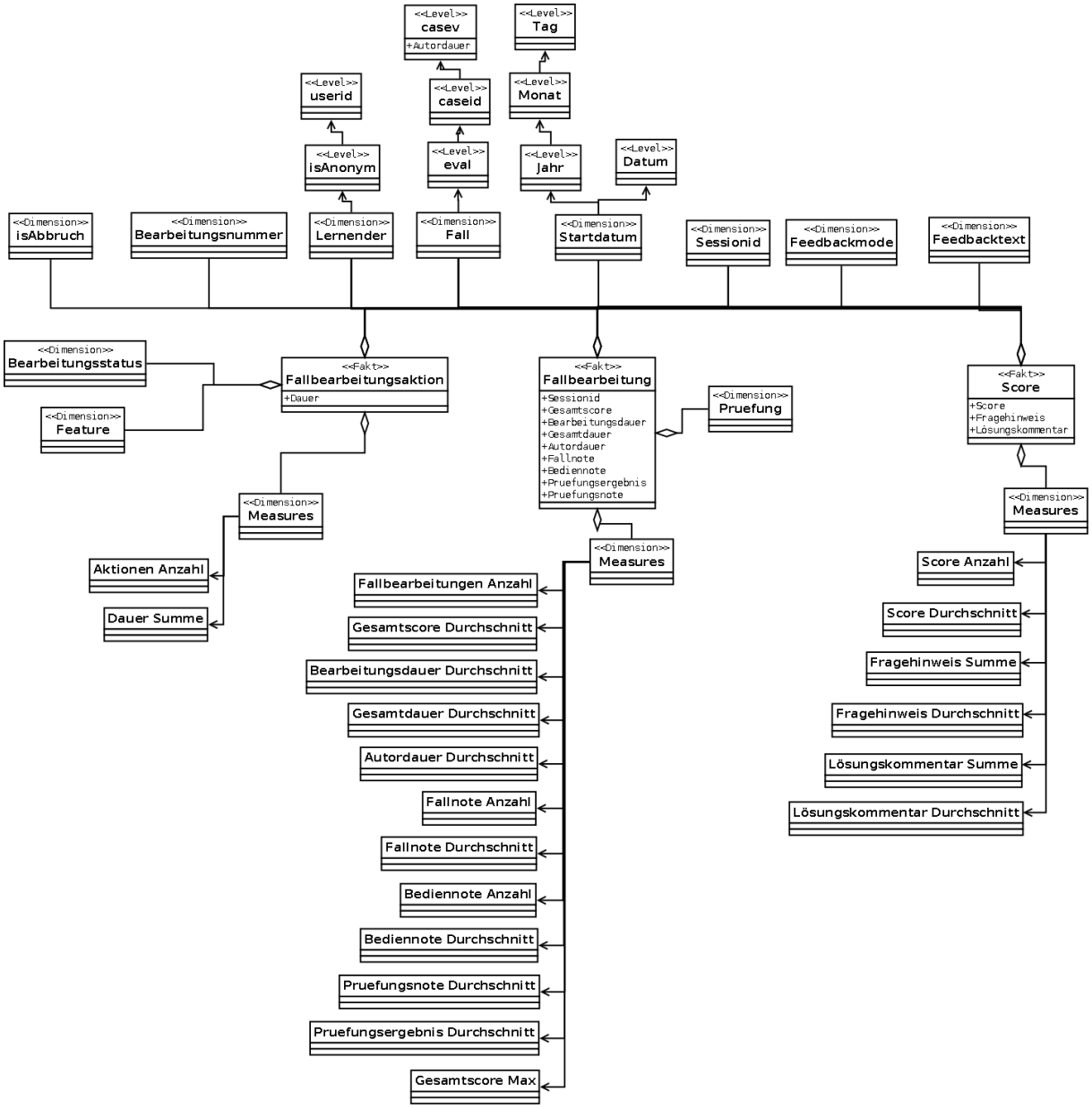


Abb. 6.14: CaseTrain MD-Modell

6.5.2.1 Gemeinsame Dimensionen

Die *Data-Cubes* teilen sich folgende Gemeinsame Dimensionen, über die sich Kennzahlen zu mehreren Data-Cubes gleichzeitig abfragen lassen. Die Attribute „isAbbruch“, „Bearbeitungsnummer“, „Sessionid“, „Feedbackmode“, „Feedbacktext“ werden direkt in Dimensionen, jeweils ohne Hierarchie, umgewandelt. Das Attribut „userid“ wird durch ein zusätzliches Attribut „isAnonym“ beschrieben. Wenn „userid“ den Wert „anonym“ besitzt, wird der Wert von „isAnonym“ auf „1“, anderenfalls auf „0“ gesetzt. Die Attribute „userid“ und „isAnonym“ bilden eine Hierar-

chie der Dimension „Lernender“. Eine Fallversion kann über die Hierarchie der Attribute „eval“, „caseid“ und „casev“ erreicht werden. Für das Startdatum wird eine eigene Dimension und zwei Hierarchien erstellt. Eine Hierarchie nennt direkt das Startdatum, eine andere teilt das Startdatum in die Attribute „Jahr“, „Monat“ und „Tag“ auf.

6.5.2.2 Fallbearbeitung

Neben den Gemeinsamen Dimensionen besitzt dieser *Data-Cube* eine eigene Dimension „Prüfung“. Damit wird für jede Fallbearbeitung die Prüfung beschrieben, die im *ER-Modell* im Attribut „Pruefung“ (ggf. *null*) der zugeordneten Prüfungsleistung enthalten ist.

Über diesen Datenwürfel der Fallbearbeitungen werden folgende *Kennzahlen* direkt durch Aggregation der Kennzahlattribute, z.B. „Sessionid“, „Gesamtdauer“ oder „Pruefungsnote“ einer Fallbearbeitung errechnet:

Fallbearbeitungen Anzahl zählt die Anzahl an gesetzten Attributwerten von „sessionid“ und benötigen wir für Tabellen 6.3, 6.4, 6.7, 6.10, 6.11 und 6.12.

Gesamtscore Durchschnitt berechnet den Durchschnitt an gesetzten Attributwerten von „Gesamtscore“ und wird in den Tabellen 6.4 und 6.5 genannt.

Gesamtscore Summe ist laut Tabelle 6.5 nötig.

Bearbeitungsdauer Durchschnitt nennen die Tabellen 6.5 und 6.10.

Bearbeitungsdauer Summe benötigen wir für die Tabelle 6.5.

Gesamtdauer Durchschnitt ist für die Tabellen 6.4 und 6.5 verlangt.

Gesamtdauer Summe leitet sich aus Tabelle 6.5 ab.

Autordauer Durchschnitt verlangt die Tabelle 6.10. Die Autordauer einer Fallbearbeitung hängt nur von der Fallversion ab; daher ist dieses Multidimensionale Modell nicht in der ersten Normalform (vgl. [44]). Solange wir für diese Kennzahl einzelne Fallversionen betrachten, gibt der Durchschnitt die Autordauer der Fallversion selbst an, wie es die Anforderung verlangt.

Fallnote Anzahl nennt Tabelle 6.1.

Bediennote Anzahl benötigen wir für die Tabelle 6.1.

Fallnote Durchschnitt ergibt sich aus Tabelle 6.1.

Bediennote Durchschnitt ergibt sich genauso aus 6.1.

Prüfungsnote Durchschnitt nennt Tabelle 6.5. Das Attribut „Prüfungsnote“ ist bei einer Prüfung eines Lernenden für alle Fälle identisch, weshalb der Durchschnitt die in der

Anforderung nötige Kennzahl ergibt.

Prüfungsergebnis Durchschnitt benötigen wir für die Tabelle 6.5. Auch das Attribut „Prüfungsergebnis“ ist bei einer Prüfung eines Lernenden für alle Fälle identisch, weshalb der Durchschnitt die in der Anforderung nötige Kennzahl ergibt.

6.5.2.3 Fallbearbeitungsaktion

Neben den Gemeinsamen Dimensionen besitzt dieser Data-Cube zwei eigene Dimensionen: „Bearbeitungsstatus“ und „Feature“, die aus dem *ER-Modell* bekannt sind. Jede „Fallbearbeitungsaktion“ besitzt eine Dauer.

Folgende *Kennzahlen* lassen sich über Aggregation direkt errechnen:

Aktionen Anzahl zählt die Anzahl an Fallbearbeitungsaktionen und benötigen wir für die Tabelle 6.7.

Dauer Summe benötigen wir für die Tabellen 6.4 und 6.7.

6.5.2.4 Score

Jedes Fakt „Score“ besitzt einen Score; außerdem werden durch Abfrage des ER-Modells für die zugehörigen Scoreaktionen „Fragehinweis“ und „Lösungskommentar“ die Dauern aufsummiert und dem Fakt als Kennzahlattribute hinzugefügt. Die Information darüber, wie häufig ein *Feature* jeweils für einen Score angezeigt wurde, ist nicht nötig und wird verworfen. Falls ein Feature nie verwendet wurde, ist die jeweilige Kennzahl *null*.

Folgende *Kennzahlen* lassen sich über Aggregation direkt errechnen:

Score Anzahl zählt die Anzahl an gesetzten Scores und benötigen wir für die Tabelle 6.8.

Score Durchschnitt benötigen wir für die Tabelle 6.8. Leere Scores werden im Durchschnitt nicht berücksichtigt.

Fragehinweis Summe summiert die Dauer der Anzeige des Fragehinweis und benötigen wir für die Tabelle 6.8.

Fragehinweis Durchschnitt berechnet den Durchschnitt der Dauer der Anzeige des Fragehinweis und benötigen wir für die Tabelle 6.8.

Lösungskommentar Summe benötigen wir für die Tabelle 6.8.

Lösungskommentar Durchschnitt benötigen wir für die Tabelle 6.8.

Speziell für Tabelle 6.4 benötigen wir sowohl die Kennzahlen „Fallbearbeitungen Anzahl“ und „Gesamtscore Durchschnitt“ des *Data-Cube* „Fallbearbeitung“, als auch die Kennzahlen „Dauer

Summe“ des *Data-Cube* „Fallbearbeitungsaktion“ – und zwar für den Fallverlauf am Ende. Auch dies können wir erreichen, da beide Data-Cubes über die gemeinsame Dimension „Lernender“ verbunden sind.

Das *MD-Modell* haben wir mit einem *Job* in *Pentaho Data Integration* umgesetzt. Jeder *Data-Cube* wird darin in einer eigenen *Transformation* so erstellt, wie es das Modell vorgibt. Das Vorgehen für einen Data-Cube besteht aus der denormalisierten Abfrage aller relevanten Attribute (Dimensions- und Kennzahlattribute) des Fakts mittels *SQL* bzw. dem *Step* „Merge Join“ und dem Eintragen von Attributen in die Dimensionstabellen mit dem *Step* „Dimension lookup/update“, mit dem gleichzeitig *Fremdschlüssel* für die Faktentabellen erstellt werden. Attribute der Kennzahlen und die Fremdschlüssel werden anschließend jeweils in eine Faktentabelle eingegeben. Bei dem Erstellen von Dimensionen ist darauf zu achten, dass die Attribute, die einen Dimensionseintrag eindeutig identifizieren, nicht *null* sind, weshalb diese, z.B. „Feedbackmode“ vor dem Eintrag in einen String, z.B. „unbekannt“ umgewandelt werden.

6.6 Reporting

Ziel des Reporting sind die Berichte, die in den Anforderungen des Business-Case genannt werden. Für das Reporting im Projekt CaseTrain haben wir insgesamt ca. 15 Stunden aufgebracht.

Wir haben die Inhalte jedes Berichts mittels eines *SQL*- bzw. *MDX*-Ausdrucks aus dem Data-Warehouse abgefragt. Den eigentlichen Bericht haben wir anschließend jeweils mit einem Portfolio an Werkzeugen erstellt und auf dem Data-Warehouse-Server veröffentlicht. Anschließend war es jederzeit möglich, Auszüge des Berichts als PDF-, CSV- oder Excel-Datei zu erstellen. Soweit nicht anders beschrieben, haben wir Standard-Funktionen von *Pentaho BI-Server*, *Pentaho Report Designer* und *Pentaho Design Studio* verwendet.

Wir haben im Reporting mittels einfacher Abfragen auf dem Data-Warehouse unter anderem die Anzahl an Fallbearbeitungen, Durchschnittsevaluationsnoten, die Abbruchquote aller Fallbearbeitungen oder den durchschnittlichen Gesamtscore abgefragt, um sie der Business-Story hinzuzufügen. Außerdem wurden solche Zahlen mehrmals mit bestehenden Informationen aus dem Data-Assay verglichen, um das ER-Modell und MD-Modell zu evaluieren.

6.6.1 Reporting auf ER-Modell: SQL

Bereits das *ER-Modell* hat mittels der Abfragesprache *SQL* die Umsetzung von Berichten ermöglicht.

6.6.1.1 Feedbacktexte

Quellcode 6.9 zeigt den *SQL*-Ausdruck, der den Inhalt des Berichts abfragt, der in der Anforderung aus Tabelle 6.2 beschrieben wird. Zu beachten ist die Darstellung von ausschließlich nicht-leeren Feedbacktexten und pro *Fallversion* eine Sortierung nach Datum. Damit der Bericht auf eine DIN-A-4-Seite gepasst hat, war es nötig, in der Spalte des Attributs „feedbacktext“ Zeilenumbrüche zu erlauben.

```
1  select
2  ER_Fall.caseid
3  , ER_Fall.casev
4  , ER_Fallbearbeitung.startdatum
5  , ER_Evaluation.fallnote
6  , ER_Evaluation.bedienungsnote
7  , ER_Evaluation.feedbacktext
8  from ER_Fallbearbeitung left join  ER_Fall on ER_Fallbearbeitung.case_key =
       ER_Fall.case_key
9  left join ER_Evaluation on ER_Fallbearbeitung.sessionid = ER_Evaluation.sessionid
10 where ER_Evaluation.feedbacktext is not null and ER_Evaluation.feedbacktext != ""
11 order by ER_Fall.caseid asc, ER_Fall.casev asc, ER_Fallbearbeitung.startdatum desc
```

Quellcode 6.9: SQL-Ausdruck für Feedbacktexte

6.6.1.2 Lösungskommentar und Score

Quellcode 6.10 zeigt den *SQL*-Ausdruck, der den Inhalt des Berichts aus Tabelle 6.9 abfragt. Zu beachten ist die Filterung nach Lösungskommentaren und beendeten Erstbearbeitungen[1] nach dem 1. Juni 2009 sowie für jeden Score die Aufsummierung der Dauer des Lösungskommentars[2]. Für Scores, bei denen der Lösungskommentar nicht angezeigt wurde, wird die Dauer Null ergänzt. Eine Typumwandlung von String zu Datumswert sowie ein mathematischer Operator haben es in SQL erlaubt, die Begrenzung des Startdatums zu realisieren. Dieser Bericht wurde im Rahmen des späteren Data-Mining auch mit anderen Schwellen der Dauer des Lösungskommentars erzeugt.

```
1  select
2  ER_Score.score
3  , if(sum(ER_Scoreaktion.dauer) is null, 0, sum(ER_Scoreaktion.dauer))
4  from ER_Score left join ER_Scoreaktion on ER_Score.score_key = ER_Scoreaktion.score_key
5  left join ER_Fallbearbeitung on ER_Score.sessionid = ER_Fallbearbeitung.sessionid
6  where (ER_Scoreaktion.dauer is null or ER_Scoreaktion.dauer < 7200)
7  and (ER_Scoreaktion.feature = "Loesungskommentar")
8  and (ER_Fallbearbeitung.isAbbruch = 0)
9  and (ER_Fallbearbeitung.startdatum > STR_TO_DATE('2009-06-01', '%Y-%m-%d'))
```

[1] Das Attribut der Bearbeitungsnummer hieß während des Projekts „nummerLernerFall“, was in der Abfrage zur besseren Lesbarkeit geändert worden ist.

[2] In der ursprünglichen Abfrage war „Loesungskommentar“ mit Umlaut geschrieben. Dies hat das verwendete Textsatzsystem dieser Arbeit nicht unterstützt.

```
10 and (ER_Fallbearbeitung.bearbeitungsnummer = 1)
11 group by ER_Score.score_key
```

Quellcode 6.10: SQL-Ausdruck Lösungskommentar Score

6.6.1.3 Kontinuierliche Fallbearbeitung

Quellcode 6.11 zeigt den *SQL*-Ausdruck, der den Inhalt des Berichts aus Tabelle 6.6 abfragt. Fallbearbeitungen besitzen neben dem Bearbeitungsstatus[3] immer eine Dauer, weshalb wir darin *null*-Werte nicht berücksichtigen mussten.

```
1 select ER_Fallbearbeitung.isAbbruch
2 , ER_Fallbearbeitung.bearbeitungsstatus
3 , ER_Fallbearbeitungsaktion.dauer
4 from ER_Fallbearbeitung
5 left join ER_Fallbearbeitungsaktion
6 on ER_Fallbearbeitung.sessionid = ER_Fallbearbeitungsaktion.sessionid
7 where ER_Fallbearbeitungsaktion.dauer < 7200
8 and (ER_Fallbearbeitung.startdatum > STR_TO_DATE('2009-06-01', '%Y-%m-%d'))
9 and (ER_Fallbearbeitungsaktion.type = "Bearbeitung")
```

Quellcode 6.11: SQL-Ausdruck Kontinuierliche Fallbearbeitung

6.6.2 Reporting auf MD-Modell: MDX

Es folgen Berichte, die wir mit Hilfe der Standard-Abfragesprache für *Multidimensionale Modelle*, *MDX*, erstellt haben.

6.6.2.1 Evaluationsnoten

Quellcode 6.12 zeigt den *MDX*-Ausdruck, der den Inhalt des Berichts aus Tabelle 6.1 abfragt.

```
1 select NON EMPTY
2 {[Measures].[fallbearbeitungen count]
3 , [Measures].[fallnote count]
4 , [Measures].[fallnote avg]
5 , [Measures].[bedienungsnote count]
6 , [Measures].[bedienungsnote avg]} ON COLUMNS,
7 NON EMPTY
8 Order({Descendants([fall].[1], [fall].[caseversion])}, [Measures].[fallnote avg], DESC)
      ON ROWS
9 from [fallbearbeitung]
```

Quellcode 6.12: MDX-Ausdruck Evaluationsnoten

[3] Das Attribut der Bearbeitungsnummer hieß während des Projekts „nummerLernerFall“, was in der Abfrage zur besseren Lesbarkeit geändert worden ist.

Zu beachten ist die absteigende Sortierung nach der durchschnittlichen Fallnote und die ausschließliche Nennung von standardmäßig evaluierten Fallversionen durch den Attributwert „1“ der Dimension „fall“.

Die Evaluationsnoten haben wir in *Pentaho Report Designer* auf drei Nachkommastellen runden lassen. Wenn für einen Fall keine Evaluationsnoten vorhanden waren, haben wir das über den Wert „-“ gekennzeichnet, was der Lesbarkeit gedient hat und möglich war, da es nicht vorgesehen war, diesen Bericht mittels Data-Mining-Werkzeugen einzulesen.

6.6.2.2 Abbruchquote

Für diesen Bericht aus Tabelle 6.3 haben wir eine *Abgeleitete Kennzahl* „Abbrüche Prozent“ aus bestehenden Kennzahlen errechnet. Diese Kennzahl errechnet für einen *Cube* den prozentualen Anzeil der Anzahl an Fallbearbeitungen, die abgebrochen wurden, von der Anzahl an allen Fallbearbeitungen.

Quellcode 6.13 zeigt den *MDX*-Ausdruck. Die Sortierung nach „Abbrüche Prozent“[4] soll für die tatsächlichen Versionen der Fälle bestehen, was mit dem Ausdruck „BDESC“ erreicht wird. Die Abbruchquote wird bereits in der Abfrage als Prozentzahl ausgedrückt.

```
1  with member [Measures].[Abbrueche Prozent]
2  as '(([isAbbruch].[All isAbbruchs].[1], [Measures].[fallbearbeitungen count])
3  / [Measures].[fallbearbeitungen count])'
4  , FORMAT_STRING = "#,###.00%"
5  select NON EMPTY
6  {[Measures].[fallbearbeitungen count], [Measures].[Abbrueche Prozent]} ON COLUMNS,
7  NON EMPTY
8  Order(Hierarchize({Descendants([fall].[All falls], [fall].[caseversion])})
9  , [Measures].[Abbrueche Prozent], BDESC) ON ROWS
10 from [fallbearbeitung]
```

Quellcode 6.13: MDX-Ausdruck Abbruchquoten

6.6.2.3 Fallverlauf am Ende

Für diesen Bericht aus Tabelle 6.4 haben wir neben der Anzahl an Fallbearbeitungen (darunter nur die, die nicht abgebrochen wurden), dem Durchschnitt des Gesamtscore und dem Durchschnitt der Gesamtdauer zusätzlich abgeleitete Kennzahlen benötigt. Diese verwenden neben *Kennzahlen* aus dem Data-Cube „Fallbearbeitung“ auch Kennzahlen aus dem Data-Cube „Fallbearbeitungsaktion“, weshalb wir für diese Abfrage beide Data-Cubes mit *Mondrian Schema Workbench* in einem *Virtuellen Data-Cube* „AktionenGesamtScoreDauer“ zusammengefasst haben. Die abgeleiteten Kennzahlen erklären sich wie folgt:

[4] Das Textsatzsystem hat innerhalb des Quellcodes keine Umlaute ermöglicht, weshalb hier „Abbrueche Prozent“ verwendet wurde.

Fallverlauf Ende Sum errechnet die Summe der Dauer des Fallverlaufs am Ende; wir haben dazu die Dimensionsattributwerte „Fallverlauf“ und „3“ (entspricht Ende) verwendet.

Fallverlauf Ende Durchschnitt errechnet den Durchschnitt der Dauer des Fallverlaufs am Ende. Hier haben wir zusätzlich leere Werte durch „0“ ersetzt, damit alle Lernenden im Bericht genannt werden.

Fallverlauf Prozent errechnet schließlich aus „Fallverlauf Ende Durchschnitt“ und „Gesamtdauer Durchschnitt“ das Verhältnis, das zwischen der Dauer des Fallverlaufs am Ende und der Gesamtdauer im Durchschnitt besteht. Des Weiteren wird der Anteil als Prozentzahl angezeigt.

Es wurden nur beendete Fallbearbeitungen im Bericht berücksichtigt. Quellcode 6.14 zeigt die Abfrage.

```
1  with member [Measures].[Fallverlauf Ende Summe]
2  as '([typ].[All typs].[Fallverlauf], [status].[All statuss].[3], [Measures].[fbak_dauer
       sum])'
3  member [Measures].[Fallverlauf Ende Durchschnitt]
4  as 'IIf(IsEmpty(([Measures].[Fallverlauf Ende sum] / [Measures].[fallbearbeitungen
       count])),
5  0.0, ([Measures].[Fallverlauf Ende sum] / [Measures].[fallbearbeitungen count]))'
6  member [Measures].[Fallverlauf Prozent]
7  as '([Measures].[Fallverlauf Ende Durchschnitt] / [Measures].[gesamtdauer avg])'
8  , FORMAT_STRING = "#,###.00%"
9  select NON EMPTY
10 {[Measures].[fallbearbeitungen count], [Measures].[gesamtscore avg],
       [Measures].[gesamtdauer avg]
11 , [Measures].[Fallverlauf Ende Durchschnitt], [Measures].[Fallverlauf Prozent]} ON
       COLUMNS,
12 NON EMPTY
13 Filter(Order([lernender].[All lernenders].[0].Children, [Measures].[Fallverlauf
       Prozent], DESC)
14 , (NOT IsEmpty([Measures].[fallbearbeitungen count]))) ON ROWS
15 from [AktionenGesamtScoreDauer]
16 where [isAbbruch].[All isAbbruchs].[0]
```

Quellcode 6.14: MDX-Ausdruck Fallverlauf am Ende

6.6.2.4 Prüfungsergebnisse

Wir hatten die Prüfungsergebnisse zunächst fast ausschließlich manuell analysiert. Dabei haben wir für jede Prüfung einen einzelnen Bericht über Studenten und ihre Leistungen in relevanten Fällen (damals noch ausschließlich der durchschnittliche Gesamtscore) erstellt und in einer *Transformation* mit den Prüfungsleistungen zusammengebracht. Dieses Vorgehen war aufgrund der vielen manuellen Schritte fehleranfällig und unflexibel. So mussten wir, nachdem weitere Attribute als Vergleichskriterien ausgewählt worden waren, den Großteil der Schritte zum Er-

stellen des Berichts wiederholen. Daher wurde entschieden, die Prüfungsergebnisse in das ER- und MD-Modell aufzunehmen, was ohne große Änderungen der bestehenden Elemente möglich war.

Quellcode 6.15 zeigt diesen Bericht aus Tabelle 6.5. Man betrachte, dass ausschließlich nicht-anonyme Lernende aus den vorhandenen Prüfungen „QMA“, „QMA-Pre“ und „Wiwi2“ ausgegeben werden.

```
1  select NON EMPTY
2  {[Measures].[fallbearbeitungen count]
3  , [Measures].[gesamtscore avg]
4  , [Measures].[gesamtscore sum]
5  , [Measures].[bearbeitungsdauer avg]
6  , [Measures].[gesamtdauer avg]
7  , [Measures].[pruefungsnote avg]
8  , [Measures].[pruefungsergebnis avg]
9  , [Measures].[bearbeitungsdauer sum]
10 , [Measures].[gesamtdauer sum]
11 , [Measures].[gesamtscore max]} ON COLUMNS,
12 NON EMPTY
13 Crossjoin({[pruefung].[All pruefungs].[QMA]
14 , [pruefung].[All pruefungs].[QMA-Pre]
15 , [pruefung].[All pruefungs].[Wiwi2]}
16 , [lernender].[All lernenders].[0].Children) ON ROWS
17 from [fallbearbeitung]
```

Quellcode 6.15: MDX-Ausdruck Prüfungsergebnisse

6.6.2.5 Fallbearbeitungsaktionen Frequenztabelle

Für diesen Bericht aus Tabelle 6.7 haben wir drei abgeleitete Kennzahlen verwendet:

Anzahl Fallbearbeitungen gibt die Anzahl aller Fallbearbeitungen an.

Dauer Durchschnitt errechnet die durchschnittliche Dauer von Fallbearbeitungsaktionen pro Fallbearbeitung.

Anzahl Durchschnitt errechnet die durchschnittliche Anzahl von Fallbearbeitungsaktionen pro Fallbearbeitung.

Für jede Kennzahl gilt die Filterung nach beendeten Fallbearbeitungen. Die verwendeten *Kennzahlen* stammen aus den *Data-Cubes* „Fallbearbeitung“ und „Fallbearbeitungsaktion“, weshalb wir wieder den *Virtuellen Data-Cube* „AktionenGesamtScoreDauer“ verwendet haben. Quellcode 6.16 zeigt die Abfrage.

```
1 with member [Measures].[Dauer Durchschnitt]
2 as '([Measures].[fbak_dauer sum] / [Measures].[Anzahl Fallbearbeitungen])'
3
4 member [Measures].[Anzahl Durchschnitt]
```

```
5  as '([Measures].[aktionen count] / [Measures].[Anzahl Fallbearbeitungen])'
6
7  member [Measures].[Anzahl Fallbearbeitungen]
8  as '([status].[All statuss], [typ].[All typs], [Measures].[fallbearbeitungen count])'
9
10 select
11 {[Measures].[Anzahl Fallbearbeitungen]
12 , [Measures].[fbak_dauer sum]
13 , [Measures].[aktionen count]
14 , [Measures].[Dauer Durchschnitt]
15 , [Measures].[Anzahl Durchschnitt]} ON COLUMNS,
16 Crossjoin({[typ].[All typs].[Bearbeitung]
17 , [typ].[All typs].[Bild]
18 , [typ].[All typs].[Fallverlauf]
19 , [typ].[All typs].[Hintergrundinfo]
20 , [typ].[All typs].[Introinfo]
21 , [typ].[All typs].[Link]
22 , [typ].[All typs].[Pause]}
23 , {[status].[All statuss].[1]
24 , [status].[All statuss].[2]
25 , [status].[All statuss].[3]}) ON ROWS
26 from [AktionenGesamtScoreDauer]
27 where [isAbbruch].[All isAbbruchs].[0]
```

Quellcode 6.16: MDX-Ausdruck Fallbearbeitungsaktionen Frequenztabelle

6.6.2.6 Scoreaktionen Frequenztabelle

Auch für diesen Bericht aus Tabelle 6.8 waren zusätzlich Abgeleitete Kennzahlen nötig.

Fragehinweis Dauer Avg berechnet die durchschnittliche Dauer des Fragehinweis pro Score.

Fragehinweis Anzahl Avg errechnet, wie hoch der Anteil der Häufigkeit des Fragehinweises an der Zahl der Scores ist.

Lösungskommentar Dauer Avg berechnet die durchschnittliche Dauer des Lösungskommentars pro Score.

Lösungskommentar Anzahl Avg errechnet, wie hoch der Anteil der Häufigkeit des Lösungskommentars an der Zahl der Scores ist.

Quellcode 6.17 zeigt den *MDX*-Ausdruck, der den Inhalt des Berichts generiert. Man beachte, dass die Kennzahlen jeweils für abgebrochene und nicht-abgebrochene Fallbearbeitungen insgesamt und nach einzelnen Bearbeitungsnummern getrennt genannt werden[5].

```
1  with member [Measures].[Fragehinweis Dauer Avg]
2  as '([Measures].[fragehinweis sum] / [Measures].[score count])'
```

[5]Das Attribut der Bearbeitungsnummer hieß während des Projekts „nummerLernerFall“, was in der Abfrage zur besseren Lesbarkeit geändert worden ist.

```
3
4 member [Measures].[Fragehinweis Anzahl Avg]
5 as '([Measures].[fragehinweis count] / [Measures].[score count])'
6
7 member [Measures].[Loesungskommentar Dauer Avg]
8 as '([Measures].[loesungskommentar sum] / [Measures].[score count])'
9
10 member [Measures].[Loesungskommentar Anzahl Avg]
11 as '([Measures].[loesungskommentar count] / [Measures].[score count])'
12
13 select NON EMPTY
14 {[Measures].[score count]
15 , [Measures].[score avg]
16 , [Measures].[fragehinweis avg]
17 , [Measures].[loesungskommentar avg]
18 , [Measures].[Fragehinweis Dauer Avg]
19 , [Measures].[Fragehinweis Anzahl Avg]
20 , [Measures].[Loesungskommentar Dauer Avg]
21 , [Measures].[Loesungskommentar Anzahl Avg]} ON COLUMNS,
22 NON EMPTY Hierarchize(Union(Union(Crossjoin({[isAbbruch].[All isAbbruchs].[0]}
23 , {[bearbeitungsnummer].[All bearbeitungsnummers]})
24 , Crossjoin({[isAbbruch].[All isAbbruchs].[0]}
25 , [bearbeitungsnummer].[All bearbeitungsnummers].Children))
26 , Union(Crossjoin({[isAbbruch].[All isAbbruchs].[1]}
27 , {[bearbeitungsnummer].[All bearbeitungsnummers]})
28 , Crossjoin({[isAbbruch].[All isAbbruchs].[1]}
29 , [bearbeitungsnummer].[All bearbeitungsnummers].Children)))) ON ROWS
30 from [score]
```

Quellcode 6.17: MDX-Ausdruck Scoreaktionen Frequenztabelle

6.6.2.7 Autordauer

Auch für diesen Bericht aus Tabelle 6.10 haben wir *Abgeleitete Kennzahlen* benötigt:

Bearbeitungsdauer (Min) berechnet die durchschnittliche Bearbeitungsdauer in Minuten.

Differenz berechnet die absolute Differenz zwischen „Bearbeitungsdauer (Min)“ und „autordauer avg“. Wenn keine Autordauer angegeben ist, wird *null* ausgegeben.

Daneben ist die Filterung nach beendeten Fallbearbeitungen und eine absteigende Sortierung nach der Differenz zu beachten. Die Funktion „Descendants“ hat es uns erlaubt, die Dimension „fall“ direkt auf der Granularität der Fallversionen zu betrachten, siehe Quellcode 6.18.

```
1 with member [Measures].[Bearbeitungsdauer (Min)]
2 as '([Measures].[bearbeitungsdauer avg] / 60.0)'
3
4 member [Measures].[Differenz]
5 as 'IIf(IsEmpty([Measures].[autordauer avg])
6 , NULL, Abs(([Measures].[Bearbeitungsdauer (Min)] - [Measures].[autordauer avg])))'
7
```

```
8  select NON EMPTY {[Measures].[fallbearbeitungen count]
9  , [Measures].[Bearbeitungsdauer (Min)]
10 , [Measures].[autordauer avg]
11 , [Measures].[Differenz]} ON COLUMNS,
12 NON EMPTY Order({Descendants([fall].[All falls], [fall].[caseversion])}
13 , [Measures].[Differenz], BDESC) ON ROWS
14 from [fallbearbeitung]
15 where [isAbbruch].[All isAbbruchs].[0]
```

Quellcode 6.18: MDX-Ausdruck Autordauer

6.6.2.8 Bearbeitungsspitzen

Auch die Berichte aus Tabelle 6.11 und 6.12 haben wir mit einer *MDX*-Abfrage erstellt. Erstere Abfrage wird in Quellcode 6.19 genannt, die zweite Abfrage ist vergleichbar aufgebaut.

```
1 select NON EMPTY Crossjoin({[Measures].[fallbearbeitungen count]}
2 , {[isAbbruch].[All isAbbruchs].[0], [isAbbruch].[All isAbbruchs].[1]}) ON COLUMNS,
3 NON EMPTY {(Descendants([time].[All times],[month]))} ON ROWS
4 from [fallbearbeitung]
```

Quellcode 6.19: MDX-Ausdruck Bearbeitungsspitzen Abbrüche

6.7 Data-Mining

Ziel des Data-Mining ist es, die geforderten Muster aus dem Business-Case zu entdecken. Wir haben insgesamt ca. 15 Stunden mit Data-Mining verbracht.

6.7.1 Evaluationsnoten

Das Data-Mining hat sich hierbei darauf beschränkt, Fallversionen nach der Fallnote absteigend sortiert darzustellen, damit besonders schlecht bewertete Fälle an sichtbarer erster Stelle stehen.

6.7.2 Feedbacktexte

Bei den Feedbacktexten handelt es sich um einen textuellen Bericht. Die Betonung von relevanten Feedbacktexten hat sich darauf beschränkt, diese für jede *Fallversion* nach dem Datum absteigend zu sortieren, damit aktuellere, und damit relevantere Feedbacktexte, an erster Stelle stehen.

6.7.3 Abbruchquote

Um *Fallversionen*, die besonders häufig abgebrochen werden, sichtbar zu machen, haben wir den Bericht nach der Abbruchquote absteigend sortiert.

Zur Evaluation haben wir stichprobenartig die Abbruchquote mittels *SQL* auf den Rohdaten im Data-Assay überprüft.

6.7.4 Fallverlauf am Ende

Wir haben den Bericht als CSV-Datei exportiert und direkt mit *RapidMiner* eingelesen. Darin ließ sich ein *Scatter-Plot* erstellen, der das Muster sichtbar gemacht hat, welches im Business-Case in Tabelle 6.4 gefordert wird. Das Diagramm wird in der Business-Story gezeigt, siehe Abbildung 6.3.

Das Ergebnis, bestehend aus Bericht und Muster, wurde durch Stichproben evaluiert; auffällige Werte bei zwei Lernenden wurden mit *SQL*-Ausdrücken auf die im Data-Assay vorverarbeiteten Daten überprüft.

6.7.5 Prüfungsergebnisse

Wir haben den Bericht als CSV-Datei exportiert und in *Rattle* eingelesen. Damit ließ sich die *Korrelationstabelle* der numerischen Attribute errechnen und in einer Visualisierung darstellen; diese ist in der Business-Story in Abbildung 6.4 enthalten. Wieder mit *Rattle* wurde ein *Scatter-Plot* erstellt, um die Korrelation zwischen Gesamtscore und Prüfungsergebnis weiter zu untersuchen, siehe Abbildung 6.5 in der Business-Story. Außerdem haben wir in diesem Zusammenhang die Gesamtdauer näher behandelt und mit *Rattle* einen *Box-Plot* anzeigen lassen, der auch in der Business-Story zu finden ist, siehe Abbildung 6.6.

6.7.6 Kontinuierliche Fallbearbeitung

Den Bericht haben wir zunächst als *CSV-Datei* exportiert. *Rattle* hat darin alle Attribute als numerisch erkannt; da wir jedoch die Dauer in Abhängigkeit der Abbruchinformation und Bearbeitungsnummer untersuchen wollten, mussten diese als kategorische Attribute verwendet werden. In einer ARFF-Datei können solche Informationen hinzugefügt werden. Deshalb haben wir die CSV-Datei mit *Notepad++* geöffnet, manuell solche Informationen hinzugefügt und die Datei anschließend mit der Endung „arff“ gespeichert. Quellcode 6.20 zeigt die manuell hinzugefügten ersten Zeilen der neuen ARFF-Datei. Diese Datei haben wir mit Rattle eingelesen und jeweils zwei *Box-Plots* und *Verteilungen* erstellt, siehe Abbildungen 6.8 und 6.7 in der Business-Story.

```
1 @relation default
2 @attribute isAbbruch {0,1}
3 @attribute nummerLernerFall {1,2,55,99}
4 @attribute dauer numeric
5 @data
6 1,99,3
7 0,2,25
8 ...
```

Quellcode 6.20: ARFF-Header für Kontinuierliche Fallbearbeitungen

6.7.7 Fallbearbeitungsaktionen Frequenztabelle

Dieser Bericht hat vor allem der Evaluation des ER- und MD-Modells gedient. Zum Beispiel wurde der Fallverlauf zeitweise bereits am Anfang einer Fallbearbeitung angezeigt, was technisch nicht möglich war. Der Grund lag darin, dass das zugehörige Event nur einmal erwartet wurde, jedoch mehrmals pro Fallbearbeitung auftreten kann; dies erklärt auch die drei Anzeigen der Introinfo in der Mitte einer Fallbearbeitung. Der Bericht wurde in die Business-Story eingefügt, siehe Abbildung 6.9.

6.7.8 Scoreaktionen Frequenztabelle

Auch hier hat sich das Data-Mining auf die bloße Betrachtung und den Vergleich mit vorhandenem Wissen beschränkt.

6.7.9 Lösungskommentar und Score

Der Bericht wurde in *Rattle* und in *RapidMiner* eingelesen. Mit Ersterem wurde der Korrelationskoeffizient bestimmt sowie ein *Box-Plot* erstellt. RapidMiner ließ uns das geforderte Muster mit einem *Scatter-Plot* erstellen und einen *Jitter* einfügen. Die Diagramme finden sich in der Business-Story in Abbildungen 6.12 und 6.11.

6.7.10 Autordauer

Das Data-Mining auf diesen Bericht hat sich auf eine Sortierung und das Betrachten der erstgenannten Fälle beschränkt.

6.7.11 Bearbeitungsspitzen

Für beide Berichte wurden direkt im *Pentaho BI-Server* zwei Säulendiagramme erstellt, die die Anzahl an abgebrochenen und nicht-abgebrochen Fallbearbeitungen bzw. Fallbearbeitungen mit einer bestimmten Bearbeitungsnummer aufeinander gestapelt visualisieren. Wir haben diese Diagramme in die Business-Story aufgenommen, siehe Abbildungen 6.1 und 6.2.

7 Ergebnis: Eine Bewertung der Methodologie

In diesem Kapitel sollen die Stärken und Schwächen der in dieser Arbeit vorgestellten Methodologie aufgezeigt werden. Es wird begründet, weshalb sie allgemein anwendbar ist auf beliebige Problembereiche. Dazu werden die einzelnen Teile der Methodologie betrachtet und anhand der beiden Einzelfallstudien bewertet.

7.1 Business-Case

Die Hauptaufgabe des Business-Case bestand darin, die Kommunikation zwischen Entscheidungsträger und Team bei der Verfeinerung von Anforderungen zu unterstützen. Er ist allgemein verwendbar, weil er sich auf zwei Konzepte stützt: Berichte aus Tabellen und die Beschreibung der Tabelleninhalte in natürlicher Sprache.

Der Business-Case hat einen Rahmen über die Projekte gespannt. Die Entwicklung der Ergebnisse über Data-Assay, Data-Warehouse, Reporting und Data-Mining, bis hin zur Präsentation in der Business-Story werden durch ihn vorgegeben:

- Für jede Anforderung werden die Datenquellen, die das Team zur Umsetzung einer Lösung benötigt und die der Entscheidungsträger zur Verfügung stellt, als Eingaben genannt. Die initiale Vorbereitung der Rohdaten im Data-Assay hängt von diesen Datenquellen ab. Wenn z.B. ein kontinuierliches Data-Mining mit aktuellen Daten anvisiert wird, muss das beim Einlesen der Datenquellen bedacht werden.
- Jeder Bericht einer Anforderung besteht aus Tabellen, diese wiederum aus Zeilen und Spalten. Ihre Inhalte beschreiben Objekte und Attribute. Um im Reporting, Data-Mining und letztendlich in der Business-Story behandelt zu werden, müssen diese Objekte und Attribute im Data-Warehouse abgebildet sein, zumindest im ER-Modell, meist aber auch im MD-Modell. Die Abfragesprachen SQL und MDX können grundsätzlich jede Information im Data-Warehouse abfragen, besitzen jedoch Einschränkungen, was die Struktur der Ausgabe anbelangt. So kann SQL z.B. keine dynamische Anzahl an Spalten abfragen. MDX kann in jeder Spalte (bzw. Zeile, wenn die Kennzahlen in den Zeilen vorgegeben werden) nur eine Kennzahl berechnen lassen, übereinanderstehende Zellen (bzw. nebeneinander stehende bei zeilenweisen Kennzahlen) der ausgegebenen Tabelle müssen also eine identi-

sche Semantik besitzen. Wenn die Inhalte der Zeilen und Spalten direkt mittels SQL oder MDX abgefragt werden können, kann der Bericht direkt im Reporting umgesetzt werden. Für Berichte, die eine aufwändigere Struktur besitzen kann es sein, dass diese nicht mittels Reporting, sondern im Rahmen der Business-Story umgesetzt werden müssen. So lassen sich Ausgaben des Reporting z.B. in Tabellenkalkulationsprogrammen einlesen und nach Belieben verändern.

- Das Muster einer Anforderung wird mittels Data-Mining-Techniken entdeckt, die auf einen Bericht ausgeführt werden. Je nach Technik muss der Bericht vorher angepasst werden.
- Letztendlich wird auch die Darstellung der Ergebnisse in der Business-Story im Business-Case festgelegt. Jeder Entscheidungsträger hat seine eigenen Vorlieben, wie ausführlich und mit welchen Schwerpunkten die Business-Story umgesetzt werden soll.

Mit den Anforderungen aus der ersten Version des Business-Case wurden auch die Entwicklungsprozesse gestartet und erste Ausgaben im Data-Assay, Data-Warehouse, Reporting und Data-Mining erstellt. Aus ihren Ergebnissen haben sich Änderungen an den Anforderungen ergeben, insbesondere wurden diese immer spezifischer, aber auch neue Anforderungen sind dabei aufgetreten. Mit Kenntnissen zu den Abhängigkeiten zwischen Business-Case und der Umsetzung kann der Aufwand für eine geänderte Anforderung abgeschätzt werden. Dabei ist der Aufwand umso größer, je mehr Schnittstellen zwischen einzelnen Teilen der Umsetzung betroffen sind:

- Wenn die Datenquellen neue bzw. aktualisierte Daten enthalten, die in die Analysen einbezogen werden sollen, impliziert das normalerweise keine Änderungen an den Entwicklungen. Stattdessen werden die aktualisierten Daten genau so analysiert, wie es die herausgearbeiteten Entwicklungsprozesse vorgeben. Dazu gehört: Das Einlesen der Daten im Data-Assay, die Integration der Daten im Data-Warehouse, das Erstellen von Auszügen der Berichte im Reporting und das Finden von konkreten Mustern im Data-Mining.
- Bei einer strukturellen Änderung der Datenquellen muss das Data-Assay abgeändert werden. Wenn die Bedeutung dieser Datenquellen unverändert bleibt, werden keine weiteren Modifikationen der Entwicklung impliziert. Eine strukturelle Änderung stellt z.B. ein verändertes Passwort zum Zugriff auf eine Datenquelle dar.
- Werden neue Muster in bestehenden Berichten verlangt, erfordert das ausschließlich Änderungen im Data-Mining.
- Werden neue Berichte und Muster über die abgebildeten Objekte und Attribute im Data-Warehouse verlangt, impliziert das Änderungen im Reporting und Data-Mining.
- Werden neue Berichte und Muster über andere Objekte und Attribute im Data-Warehouse verlangt, werden Änderungen im Data-Warehouse, Reporting und Data-Mining notwendig. Falls es sich um neue Objekte und Attribute handelt, können bestehende Berichte

und Muster unverändert bleiben. Falls es sich jedoch um abgeänderte Objekte und Attribute handelt, kann es erforderlich sein, jegliche vorhandenen Berichte und Muster zu aktualisieren.

Ein wichtiges Kriterium zum Bewerten eines Vorgehens zum Data-Mining ist die Zeit, die zum Lösen eines Problems benötigt wird. Tabelle 7.1 gibt eine Übersicht über den zeitlichen Aufwand des Teams während der Data-Mining-Projekte. Die Zeitaufwände des Data-Mining-Experten konnten durch das KDDM-Wiki abgeschätzt werden, in dem die nötige Anzahl an Stunden für Aufgabendurchführungen dokumentiert wurde. Der Aufwand der Domänen- und Daten-Experten wurde anhand der Gesamtdauer der Teambesprechungen abgeschätzt und ausschließlich als zusätzliche Stunden dem Data-Assay hinzugefügt. Das Herausarbeiten der Anforderungen im Business-Case und das Zusammenstellen der Ergebnisse in der Business-Story sind nicht Teil der Entwicklung, ihre Aufwände wurden nicht vollständig dokumentiert und waren daher nicht abzuschätzen.

	DA	**DW**	**RP**	**DM**
Bachelor	(60) 39%	(50) 32%	(40) 26%	(5) 3%
CaseTrain	(100) 59%	(40) 23%	(15) 9%	(15) 9%

Tab. 7.1: Für die Entwicklungsteile Data-Assay (DA), Data -Warehouse (DW), Reporting (RP) und Data-Mining (DM) grobe Abschätzung der Ein-Mann-Stunden sowie des prozentualen Anteils an der Entwicklungszeit

Die Lebenszyklen der beiden Data-Mining-Projekte haben aus kontinuierlich spezifizierten Anforderungen bestanden. Bei einer Änderung der Anforderungen musste überprüft werden, welche Teile der Entwicklungsprozesse zu modifizieren waren. Der Lebenszyklus wurde daher jeweils von vorne gestartet und neue Meilensteine für Data-Assay, Data-Warehouse, Reporting und Data-Mining erstellt. Außer dieser „Wasserfallmethode“ [48] sind weitere Lebenszyklen denkbar; Lebenzyklen aus der Software-Entwicklung können als Vorlage dienen (vgl. [30]), müssen jedoch erweitert werden, um die hohe Iterativität und Interaktivität von Data-Mining-Projekten zu berücksichtigen. Unnötiger Aufwand für Projekte lässt sich vermeiden, wenn die Anforderungen immer so konkret wie möglich formuliert werden. Eine Anforderung an die Data-Mining-Projekte in Form eines Berichts und Musters war grundsätzlich verständlich für den Entscheidungsträger. Auch war es für den Entscheidungsträger teilweise möglich, eigene Anforderungen zu bilden oder vorhandene zu verfeinern, dies allerdings häufig nur dann, wenn ihm gleichzeitig eine erste Umsetzung der Anforderung gezeigt wurde. Möglicherweise hilft es dem Entscheidungsträger, Anforderungen spezifischer zu formulieren, wenn Berichte und Muster nicht textuell in einer Tabelle beschrieben, sondern in Konzeptmodellen dargestellt werden, ähnlich den Use-Case-Diagrammen der Unified-Modeling-Language (vgl. [48]). Aggregation Workflows [19] und Werke von Zubcoff und Trujillo (vgl. [72]; [73]) bieten dazu erste Ansätze.

7.2 Business-Story

Auch die Business-Story stützt sich auf die natürliche Sprache und stellt keine Einschränkung für den Problembereich dar. Der Nachteil besteht darin, dass das Zusammenstellen der Ergebnisse hauptsächlich ein aufwändiger manueller Prozess ist. Langfristig könnte sich eine alternative Möglichkeit zur Präsentation der Ergebnisse empfehlen:

Dabei werden die Ergebnisse nicht in einer Business-Story zusammengefasst, sondern in gemeinsamen Sitzungen zwischen Entscheidungsträger und Team erarbeitet. Dazu sind laut Kohavi [39] folgende Voraussetzungen nötig, die von der vorgestellten Methodologie in Ansätzen erfüllt werden.

- Die Daten sind in Form von Berichten über eine webbasierte Benutzerschnittstelle erreichbar.
- Die Berichte sind parametrisierbar, damit dem Entscheidungsträger individuelle Fragen „out-of-the-box“ [39] beantwortet werden können.
- Die Berichte sind von technisch versierten Personen jederzeit frei modifizierbar.

Allerdings sind für diesen Ansatz noch zu viele manuelle Arbeiten notwendig, als dass effiziente Sitzungen realistisch erscheinen. Eine nachträgliche Anpassung der Berichte wäre z.B. zu zeitaufwändig; Diese ist bisher häufig nötig, damit die Berichte von einem Data-Mining-Werkzeug gelesen werden können. Zudem ist der Funktionsumfang von Reporting- und Data-Mining-Werkzeugen noch nicht ausreichend, um individuelle Bedürfnisse von Entscheidungsträgern zu erfüllen.

7.3 Data-Assay

In den Einzelfallstudien war es jeweils möglich, die Rohdaten in Tabellarische Form umzuwandeln. Davon kann man nicht immer ausgehen: Daten können auch als Streams, Zeitreihen oder Sequentielle Date sowie qualitative Texte oder mediale Daten (Musik, Bilder, Videos) vorliegen. Diese können entweder in Tabellarische Form transformiert werden und nach der vorgestellten Methodologie analysiert werden, oder bedürfen einer anderen Herangehensweise. In dieser Arbeit konnte bestätigt werden, dass die Tabellarische Form sehr allgemein anwendbar ist und auch genügend Flexibilität bietet, um ausgefallenere Daten zu analysieren. So handelte es sich beispielsweise bei den Logdaten im Projekt CaseTrain um annähernd sequentielle Daten, die sich durch Vorverarbeitung in Tabellarische Form bringen ließen und analysiert werden konnten.

Das Data-Assay war für die Evaluation von großer Bedeutung. Aussagen der Experten, aber auch Ergebnisse aus Data-Warehouse, Reporting und Data-Mining wurden immer wieder direkt

anhand der Rohdaten überprüft, da in diese stets das größte Vertrauen bestand.

Allerdings kann nicht alles anhand des Data-Assay evaluiert werden. Aufwändig abgeleitete Kennzahlen auf spezielle Teilmengen des MD-Modells lassen sich beispielsweise nicht so einfach mittels SQL auf die Rohdaten wiederholen. Stattdessen müssen die Ergebnisse in Data-Assay, Data-Warehouse, Reporting und Data-Mining jeweils davon ausgehen, dass die vorherigen Ergebnisse korrekt sind. Dies kann nur durch Tests und dem Hintergrundwissen der Daten- und Domänen-Experten sichergestellt werden. Im Projekt Bachelor wurde beispielsweise mit SQL auf die Rohdaten, mit SQL auf das ER-Modell und mittels MDX auf das MD-Modell die Anzahl an Studenten abgefragt, miteinander verglichen und so das Vertrauen in die Ergebnisse gestärkt. In diesem Zusammenhang hat sich die Frage gestellt, ob im Data-Mining ähnliche Konzepte wie die JUnit-Tests in der Java-Programmierung möglich sind. Durch solche Tests könnten Standard-Ausgaben aus Data-Warehouse, Reporting und Data-Mining mit Ausgaben aus dem Data-Assay verglichen und Fehler bei Änderungen automatisch erkannt werden.

7.4 Data-Warehouse

Während der Entwicklung von ER-Modell und MD-Modell für beide Projekte sind keine Einschränkungen aufgetreten, um Objekte aus der Realität in der Datenbank abzubilden. Zudem wurde die Erfahrung gemacht, dass das Multidimensionale Modell mit Datenwürfeln und einer multidimensionalen Abfragesprache sehr gut geeignet ist, um nicht nur innerhalb des Teams über die Daten, sondern auch mit einem Entscheidungsträger über die Umsetzung von Anforderungen zu sprechen.

Nach der Konzeption der Modelle war die Umsetzung relativ fest vorgegeben, z.B. die Erstellung von eigenen Fremdschlüsseln für das ER-Modell aus fachlichen Fremdschlüsseln der Rohdaten. Dies hatte zum Vorteil, dass dabei grundsätzlich wenig Fehler aufgetreten sind. Trotz der festen Vorgaben aus der Konzeption war es jedoch meist eine aufwändige manuelle Arbeit, ER-Modell und MD-Modell zu erstellen, auch aufgrund umständlicher Bedienung der entsprechenden Werkzeuge. Die Erfahrungen während der Projekte geben den Eindruck, dass der Entwicklungsaufwand deutlich vermindert werden könnte, wenn Teile der Umsetzung automatisiert werden würden. So gibt es bereits erste Ansätze, um ein MD-Modell automatisch aus einem ER-Modell zu erstellen (vgl. [64]).

7.5 Reporting und Data-Mining

Diese Arbeit hat das Reporting als essentiellen Teil der Entwicklungsprozesse gesehen. Berichte bieten eine verständliche Sicht auf die Rohdaten. Dabei erlauben es die Abfragesprachen,

jede Art der Information aus den Rohdaten zu extrahieren. Im Data-Mining lassen sich anschließend eine Vielzahl an verborgenen Informationen aus den Berichten extrahieren. In der Business-Story können dann die Informationen in beliebigen Darstellungsformen präsentiert werden. Daher besitzen auch Reporting und Data-Mining keine Einschränkungen für andere Anwendungsbereiche. Zudem wurde die Erfahrung gemacht, dass die strikte Trennung von Reporting und Data-Mining zum Verständnis des Entscheidungsträgers beiträgt. Allerdings ist sie mit höherem Aufwand verbunden, insbesondere wenn Berichte manuell angepasst werden müssen, um von den Data-Mining-Werkzeugen eingelesen werden zu können. Technisch lassen sich die Funktionen des Reporting, Data-Mining und der Ergebnisdarstellung besser intergrieren. Ein vielversprechender Ansatz ist das Online-Analytical-Mining (OLAM), das Han bereits 1997 beschrieben hat [31], dessen Möglichkeiten jedoch bisher weitgehend unbeachtet geblieben sind. Im OLAM werden Ergebnisse des Data-Mining zusammen mit den analysierten Daten in einem Data-Cube angezeigt. Ramakrishnan hat „Prediction Cubes“ [18] vorgestellt, bei denen die Vorhersage eines Wertes direkt im Data-Cube vorgenommen wird – für interaktiv bestimmbare Teilmengen oder Granularitätsstufen. Ein einfaches Beispiel für OLAM im Deskriptiven Data-Mining sind Diagramme, wenn z.B. die ECTS-Punkte-Verteilung der betrachteten Studierenden direkt im Bericht oder im OLAP angezeigt wird.

7.6 Dokumentations- und Wissensmanagementsystem

Das KDDM-Wiki und Versionierungssystem wurden exzessiv während der Projekte verwendet, das Erstgenannte enthält bereits 161 Inhaltsseiten, das Zweitgenannte ca. 2 GB an Daten. Häufige Seiten des Wikis sind Aufgaben, Durchführungen, Artefakte und Todos, sehr selten erstellt wurden Resultate und Evaluationen, die stattdessen meist direkt auf den Ziel- oder Aufgabenseite beschrieben wurden. Wenn auch zum Mehrbenutzerbetrieb ausgelegt, wurde das Dokumentations- und Wissensmanagementsystem bisher hauptsächlich im Einzelbenutzerbetrieb verwendet. Daher können keine empirisch unterstützten Aussagen zur Eignung für das Wissensmanagement innerhalb einer Organisation gemacht werden.

Jedoch hat sich auch im Einzelbenutzerbetrieb bestätigt, dass genaue Dokumentation unerlässlich ist. So war es innerhalb der Projekte häufig nötig, Aufgaben anzupassen oder Durchführungen zu wiederholen – Informationen aus vergangenen Aufgaben oder Durchführungen waren dabei hilfreiche Anknüpfungspunkte. Beispielsweise wurden die SQL- und MDX-Abfragen mit kurzen Erklärungen im Wiki notiert und konnten als Vorlagen für weitere Abfragen verwendet werden. Die beiden Projekte – parallel durchgeführt – profitierten dabei nicht nur von ihren eigenen Dokumentationen, sondern auch von der Dokumentation des jeweils anderen Projekts. Insbesondere Lessons-Learned konnten grundsätzlich in beiden Projekten berücksichtigt werden, z.B.: „Die Benennung der Tabellen in der Datenbank sollte so

geschehen, dass zugehörige Tabellen ein gemeinsames Präfix erhalten und in einer alphabetischen Liste nebeneinander stehen. Dies erhöht die Wartbarkeit.“ Der Wissensimport geschah auch nicht nur zu Beginn eines Projekts (vgl. [48]), sondern über das gesamte Projekt hinweg – von Planung im Business-Case, über die Entwicklung, bis hin zur Präsentation der Ergebnisse.

Während der Projekte wurden mehrere Lessons-Learned gesammelt. Dabei ist aufgefallen, dass Lessons-Learned immer nur in einem speziellen Kontext gelten. Nur wenn dieser Kontext vorhanden ist, kann auch die Lesson-Learned hilfreich sein. Ein Beispiel für einen solchen Kontext stellt die Visuelle Prozessmodellierung im ETL dar. Im Folgenden einige Beispiele für Lessons-Learned:

- Das Einstellen von Primärschlüsseln in der Datenbank, wenn auch aus Performanzgründen nicht immer zwingend notwendig, stellt einen sinnvollen Test dar, ob die Tabelle die erwarteten Daten enthält.
- Ungültige Werte in den Rohdaten, die keinerlei Bedeutung für die Anforderungen haben, sollten direkt gefiltert und nicht in ER- und MD-Modell aufgenommen und dort als ungültig markiert werden. Ansonsten ist das Risiko groß, dass sie versehentlich in Analysen einfließen.
- Auch während des Projekts ist es wichtig, das Glossar aus dem Business-Case zu pflegen, um mehrdeutige oder neue Begriffe für alle Beteiligten eindeutig zu definieren.

Eine Best-Practice kann als erweiterte oder kombinierte Lesson-Learned gesehen werden, die von vielen Personen einer Organisation berücksichtigt wird. Zwei Data-Mining-Projekte haben für die Entwicklung von Best-Practices noch nicht ausgereicht, manche Lessons-Learned motivieren jedoch mögliche Best-Practices:

- Trotz Visueller Prozessmodellierung können ETL-Umsetzungen schnell unübersichtlich werden. Es ist zu empfehlen, diese möglichst stark in einzelne Aufgaben zu modularisieren, z.B. sollte die Umsetzung des ER-Modells in separaten Dateien zur Umsetzung des MD-Modells gespeichert werden. Solche Änderungen können auch nachträglich vorgenommen werden, als Refaktorisierungsmaßnahme zur verbesserten Wartung.
- Wenn eine Anforderungsänderung entweder Änderungen im Reporting oder im Data-Warehouse impliziert, muss entschieden werden, wo die Änderungen durchgeführt werden sollen. Änderungen am Data-Warehouse sind aufwändiger, können sich aber schnell amortisieren, wenn sie für mehrere Berichte nötig sind.
- Es sind Situationen aufgetreten, in denen mehrere Versionen von ETL-Maßnahmen alternativ verwendet wurden. So waren von den Bachelordaten teilweise bereits neue Daten in einem anderen Format verfügbar und sollten eingelesen werden, gleichzeitig musste jedoch sichergestellt sein, dass auch die älteren Daten noch einlesbar waren. Es müssen

Möglichkeiten gefunden werden, um solche Situationen effektiv zu organisieren, z.B. durch Versionsverzweigungen, wie in der Software-Entwicklung.

- Viele Werkzeuge speichern ihre Arbeitsdateien in sog. Workspaces. Es hat sich als praktisch erwiesen, für jedes Projekt und jedes Werkzeug einen eigenen Workspace-Ordner zu erstellen, der auch an das Versionierungssystem angebunden ist. Diese Ordner können normalerweise uneingeschränkt anderen Personen zur Verfügung gestellt werden, da die enthaltenen Workspace-Dateien im Normalfall keine konkreten Daten sondern Meta-Informationen enthalten. Ein Beispiel sind die ETL-Werkzeuge, die in ihren Dateien beschreiben, wie die Daten aufgebaut sind und verarbeitet werden, jedoch nicht, welche Inhalte die Daten besitzen.

7.7 Software- und Hardware-Komponenten

Abgesehen von dem Betriebssystem Microsoft Windows XP wurden die Projekte ausschließlich unter Verwendung von Open-Source-Werkzeugen durchgeführt, auch das Betriebssystem hätte durch eine freie Linux-Variante ersetzt werden können, da die Werkzeuge für verschiedene Plattformen geeignet waren. Selbst in einem kommerziellen Szenario wären damit keinerlei Lizenzgebühren angefallen. Die Entscheidung, ein Werkzeug zu nutzen, wurde maßgeblich davon beeinflusst, ob mit dem Werkzeug bereits Erfahrungen gemacht worden sind. Verschiedene Data-Mining-Werkzeuge sind unterschiedlich gut für spezielle Techniken geeignet, z.B. ist *VIKAMINE* spezialisiert auf die Subgruppenentdeckung, *Rattle* dagegen erlaubt die Erstellung von verschiedensten Diagrammen; deshalb wurde jeweils ein Portfolio an Data-Mining-Werkzeugen verwendet. Zusammenfassend kann man feststellen: Die etablierten, kommerziellen Hersteller von *Closed-Source*-Werkzeugen haben es zwar geschafft, den gesamten KDDM-Prozess in ihren Werkzeugen abzudecken. Dies war noch vor einigen Jahren nicht der Fall (vgl. [39]), als sie zwar verschiedene Data-Mining-Algorithmen, aber keine Möglichkeiten für aufwändiges ETL, Berichte oder Visualisierungen boten. Auf der anderen Seite bieten jedoch *Open-Source*-Werkzeuge mittlerweile identische Funktionen, ohne finanzielle Anfangsinvestition. Daher können sie zum Data-Mining allgemein ohne Einschränkungen eingesetzt werden.

Zwei Erfahrungen, die zu Open-Source-Werkzeugen gemacht worden sind:

- Die Entwicklungszyklen von Open-Source-Werkzeugen sind sehr kurz, ggf. mit großen Änderungen zwischen Versionen. Zum Beispiel war zu Beginn dieser Arbeit *Business Intelligence Server* in der Version „2.0-stable“, bei Abschluss in der Version „3.5-stable“ verfügbar. Für eine Versionsänderung war jeweils ein erneuter manueller Aufwand zur Installation und Konfiguration notwendig; dieser Zusatzaufwand hat sich jedoch auf Grund zahlreicher Neuerungen und Fehlerbehebungen stets gelohnt.

- Open-Source-Werkzeuge erfordern keine finanzielle Anfangsinvestition, aber erhöhten Aufwand. Wohingegen Closed-Source-Werkzeuge stets mit einer Dokumentation ausgeliefert werden, muss diese bei Open-Source-Werkzeugen oft mit einigem Aufwand aus verschiedenen Quellen zusammengesucht werden. Beispielsweise diente für Werkzeuge von Pentaho ein Forum, ein Wiki sowie mehrere verschiedene Projektwebseiten als Informationsquelle. Diese Informationen waren selten so aufbereitet, dass man sie ohne größere technische Vorkenntnisse und Erfahrungen versteht. Der Zugang zu solchen Informationsquellen kann als weiteres Auswahlkriterium für ein Open-Source-Werkzeug gelten.

Im Bezug auf die Hardware-Komponenten wurde es für wichtig empfunden, Hardware-Ausfällen durch regelmäßige Backups auf verschiedenen Medien vorzubeugen; insbesondere am *Dokumentationsserver* und *Data-Warehouse-Server* können solche Ausfälle Projekte deutlich verzögern. Die Leistungskapazitäten der Rechner waren zur Durchführung der Projekte ausreichend.

8 Diskussion und Ausblick

Im folgenden Kapitel soll das Ergebnis dieser Arbeit diskutiert und ein Ausblick für vielversprechende Fortführungen gegeben werden.

In Kapitel 3 dieser Arbeit wurde eine Data-Mining-Methodologie vorgestellt und behauptet, dass sie allgemein anwendbar und Entscheidungsträger-verständlich ist. Kapitel 7 begründet die Allgemeinheit durch eine Gesamtbetrachtung zweier Einzelfallstudien. Die Verständlichkeit für Entscheidungsträger wird durch Literatur zum Thema gestützt. Beide Data-Mining-Projekte wurden aus Sicht der Entscheidungsträger erfolgreich durchgeführt. Allerdings besitzen die Hauptentscheidungsträger einen technischen Hintergrund und werden daher nicht als repräsentativ genug gesehen, um eine formale Evaluation der Hypothese zu ermöglichen. Die Behauptung kann bestärkt werden, indem weitere Data-Mining-Projekte mit repräsentativeren Entscheidungsträgern mittels der Methodologie durchgeführt und in Fallstudien betrachtet werden.

Stattdessen soll die vorgestellte Methodologie im Folgenden mit einem weiteren Ansatz verglichen werden, der sich während der Data-Mining-Projekte angeboten hat: Anstatt ein *vorgegebenes Framework* zur Umsetzung der Projekte zu verwenden, wird in diesem Ansatz besonderer Wert auf Flexibilität gelegt und der Kern der technischen Durchführung durch ein *individuelles System* realisiert. Dazu zunächst eine kurze Definition der Ansätze:

Vorgegebenes Framework: Dieser Ansatz besteht aus einem Open-Source-Komponenten-basierten System, das verschiedene bewährte Konzepte (ER-Modell, MD-Modell) und Standards (SQL, MDX) integriert. Es enthält den technischen Teil der in dieser Arbeit vorgeschlagenen Methodologie.

Individuelles System: Auch dieser Ansatz wird vorhandene Werkzeuge verwenden müssen, wenn die Entwicklungszeit überschaubar gehalten werden soll. So werden zumindest für das ETL, die Datenbank und das Data-Mining – wenn für die Analysen notwendig – bestehende Systeme genutzt. Desweiteren wird ein eigenes System Software-Bibliotheken nutzen, um allgemeine Funktionen, wie den Zugriff auf die Datenbank oder den Export von Daten als PDF- oder XLS-Dateien nicht selbst implementieren zu müssen. Mittels einer Programmiersprache, z.B. Java, wird nun ein System programmiert, das die Daten einliest, in eigene Datenstrukturen übersetzt, beliebig kombiniert und transformiert sowie als Berichte oder

in Data-Mining-Werkzeuge exportiert.

Im Folgenden soll gezeigt werden, für welche Anwendungen die Ansätze geeignet sind. Da beide Ansätze beim ETL, bei der relationalen Datenbank sowie beim Data-Mining vorhandene Werkzeuge und Bibliotheken nutzen, beschränkt sich der Vergleich größtenteils auf das Reporting.

8.1 Vorgegebenes Framework

Ein vorgegebenes Reporting-Framework basiert vollständig auf Open-Source-Werkzeugen und ist damit größtenteils von den Funktionen abhängig, die von solchen Werkzeugen angeboten werden. Größtenteils deshalb, weil ein Open-Source-Werkzeug grundsätzlich auch selbständig erweitert werden kann, was jedoch einen gewissen Einarbeitungsaufwand sowie die Kommunikation mit anderen Entwicklern erfordert. Auch in den Data-Mining-Projekten dieser Arbeit war es für einige Anforderungen nötig, die Standardfunktionen des Framework mittels einer Programmiersprache zu erweitern. Für manche Änderungen am Framework, z.B. zur Verbesserung der Performanz, ist jedoch der Einarbeitungsaufwand zu hoch; spezielle Anforderungen lassen sich daher nicht immer direkt umsetzen, sondern können einen Kompromiss erfordern.

Allerdings kann davon ausgegangen werden, dass die wichtigsten Funktionen in einem Framework enthalten sind. Denn die Werkzeuge werden von einer Open-Source-Community gepflegt und weiterentwickelt. Sie müssen verschiedene Parteien zufrieden stellen, werden daher von Beginn an sehr allgemein gehalten, mit einer großen Anzahl an unterschiedlichen Funktionen, die zudem direkt aus der Praxis heraus gefordert wurden. Daher kann auch eine gewisse Qualität der Funktionen angenommen werden.

Wenn man ausschließlich bereits vorhandene und getestete Funktionen nutzt, ergeben sich einige Vorteile: Das Framework wird beinahe ausschließlich über festdefinierte Schnittstellen bedient und konfiguriert. So ist z.B. MDX sehr gut dokumentiert und auch von einem Außenstehenden jederzeit erlernbar. Außerdem kann einem Entscheidungsträger die Funktionalität meist anhand von einfachen Beispielen demonstriert werden, das Ausarbeiten von speziellen Anforderungen fällt leichter, genauso wie das Begründen von Aufwandseinschätzungen. In dieser Arbeit wurde die Erfahrung gemacht, dass der Aufwand einer Umsetzung anhand der Dokumentation und eigenen Erfahrungen zu einfachen Beispielen grundsätzlich gut abgeschätzt werden kann.

Letztendlich bietet ein solches Framework den laut Kapitel 7 zum Wissensmanagement nötigen Kontext für Lessons-Learned und eignet sich auf Grund der Lerneffekte insbesondere, wenn mehrere Projekte in verschiedenen Domänen durchgeführt werden.

8.2 Individuelles System

Ein individuelles Reporting-System erstellt die Berichte nicht direkt mittels SQL oder MDX, sondern aus seinen eigenen Datenstrukturen heraus. Da das System sich nicht auf bestimmte Datenstrukturen festzulegen hat, kann jede beliebige Anforderung umgesetzt werden.

Die Analyse, der Entwurf und die Umsetzung eines solchen Systems gleichen damit der reinen Software-Entwicklung. Dort kann eine Anforderung meist durch viele verschiedene Lösungen erfüllt werden. In dieser Arbeit wird die Auffassung vertreten, dass Aufwandseinschätzungen umso schwieriger sind, je allgemeiner das Spektrum an möglichen Lösungen ist.

Die Datenstruktur kann genau an die Anforderungen angepasst werden. Zwei Beispiele:

- Optimierungen der Performanz sind direkt möglich.
- Das System kann über das Reporting heraus Teile des Data-Mining und der Ergebnispräsentation integrieren.

Die individuelle Umsetzung der Anforderungen bringt jedoch auch Nachteile mit sich:

Jegliche Funktionen des Systems müssen explizit ergänzt werden. Auch unter Verwendung von Bibliotheken besteht sicherlich ein höherer Anfangsaufwand als mit einem bestehenden Framework.

Nur wenn die Datenstrukturen so abstrahiert werden können, dass sie ein Entscheidungsträger versteht, kann das Team mit ihm auf Umsetzungsebene sprechen und Aufwandseinschätzungen begründen, wie es z.B. bei Data-Cubes möglich ist.

Änderungen an den Anforderungen können auch immer Änderungen an den Datenstrukturen erforderlich machen. Bei größeren Änderungen, die eigene Performanzoptimierungen erfordern, kann es nötig sein, neue Datenstrukturen zu definieren. Dies kann mit erheblichem Zusatzaufwand verbunden sein, der zudem wieder nicht sehr leicht abgeschätzt oder begründet werden kann.

Genauso wenig wie die Datenstrukturen wird auch die Dokumentation des Systems vorgeschrieben. Wenn diese vernachlässigt wird – was häufig der Fall ist – ist ein solches System schlecht wartbar, insbesondere nach einem längeren Zeitraum oder durch Personen, die nicht an der Entwicklung beteiligt waren.

Insgesamt ist ein individuelles System auf den speziellen Anwendungsfall spezialisiert und auch beschränkt. Gleichzeitig stellt die Software-Entwicklung einen sehr allgemeinen Umsetzungskontext dar, es ist daher unwahrscheinlich, dass Lessons-Learned auch in weiteren Projekten ihre Anwendung finden.

Zusammenfassend kann man sagen, dass das vorgegebene Framework mit Sicherheit besser geeig-

net ist, um mehrere verschiedene Data-Mining-Projekte durchzuführen. Das individuelle System dagegen kann besser geeignet sein, um spezielle Anforderungen eines Projekts zu erfüllen. In der Praxis wird meist eine Kombination beider Ansätze auftreten: Der Großteil der Anforderungen wird durch ausgereifte Funktionen eines Frameworks erfüllt, nur für spezielle Anforderungen ist es nötig, Teile der Lösung selbst zu programmieren. Dabei bin ich der Meinung, dass die Open-Source-Werkzeuge auf einem guten Weg sind, um in Zukunft auch sehr spezielle Data-Mining-Projekte in einem vorgegebenen Framework mit deutlich weniger Aufwand als mit einem individuellen System durchzuführen.

Um diese Behauptung zu stärken, könnten weitere Data-Mining-Projekte – eine Hälfte durch Anwendung eines vorgegebenen Frameworks, die andere Hälfte durch Entwicklung eines individuellen Systems – durchgeführt und ihr Aufwand in einer Mehrfachfallstudie verglichen werden. Stattdessen halte ich es für vielversprechender, die *Schnittstellen des Data-Mining* genauer zu untersuchen, z.B. zwischen Daten-, Domänen und Data-Mining-Experten, zwischen Endnutzer und Team, zwischen einzelnen Techniken oder zwischen einzelnen Komponenten. Kenntnisse zu den Schnittstellen werden Frameworks ermöglichen, die auch die Flexibilität von individuellen Systemen aufweisen.

9 Zusammenfassung

Diese Arbeit stellt einen Ansatz zum Entscheidungsträger-verständlichen Deskriptiven Data-Mining vor und bestätigt seine Allgemeinheit durch Anwendung in zwei Projekten. Denn häufig nutzt die Wissensentdeckung in Daten ihr Potenzial nicht vollständig aus. Als Hauptursache gilt die Unzugänglichkeit von Data-Mining für den Endnutzer. Der Ansatz dieser Arbeit beschreibt in Form einer Data-Mining-Methodologie, dass die Erklärung und Beschreibung von Daten im Deskriptiven Data-Mining verständlich für den Endnutzer der Ergebnisse, den Entscheidungsträger, ist, wenn es folgendermaßen durchgeführt wird: Das Entwicklerteam und der Entscheidungsträger arbeiten gemeinsam Anforderungen heraus, die das Team in einem Projekt erfüllen soll, um zur Lösung eines Problems des Entscheidungsträgers beizutragen. Jede Anforderung nennt einen Bericht mit verständlichen Informationen und ein Muster, das unter diesen Informationen entdeckt werden soll. Um die Berichte und die Interpretationen ihrer Muster schließlich zusammenhängend präsentieren zu können, erstellt das Team vier Ausgaben: Eine Beschreibung der Daten unter Berücksichtigung von Hintergrundwissen; ein Data-Warehouse, für einen wohldefinierten Zugriff auf die Daten; außerdem die in den Anforderungen genannten Berichte und Muster. In der Arbeit werden ferner Komponenten definiert, mit denen diese Ausgaben erzeugt werden können. Darunter wird auch eine Komponente beschrieben, die der Dokumentation und dem Wissensmanagement in solchen Projekten dient. Die Komponenten können vollständig durch frei verfügbare Open-Source-Werkzeuge abgedeckt werden.

Die Methodologie wurde in zwei Data-Mining-Projekten erfolgreich durchgeführt. Im ersten Projekt werden die im Rahmen des Bologna-Prozesses neu eingeführten Bachelorstudiengänge an der Universität Würzburg mittels elektronisch erfasster Prüfungsdaten bewertet. Die Ergebnisse aus diesem Projekt, wenn auch aus datenschutzrechtlichen Gründen nicht vollständig offengelegt, sind höchst relevant, was nicht zuletzt durch zahlreiche Bildungsstreiks in ganz Deutschland, darunter auch in Würzburg, bestätigt wird. Im zweiten Projekt wird der Nutzen des fallbasierten Lehrsystems CaseTrain, das den Studierenden von der Universität Würzburg über das Internet angeboten wird, mittels Benutzungsdaten beurteilt. Jedes Jahr muss die Finanzierung des Systems mittels Studiengebühren erneut beantragt werden, weshalb auch die Ergebnisse dieser Leistungskontrolle relevant sind.

Erfahrungen aus diesen beiden Data-Mining-Projekten belegen den Schluss, dass die Methodologie auf beliebige Domänen übertragbar ist. Eine Anwendung der Methodologie über mehrere

Projekte hinweg bietet Lerneffekte mit zahlreichen Lessons-Learned bis hin zu Best-Practices.

Vorgehensweisen im Data-Mining, die für Entscheidungsträger verständlich sind – das Thema dieser Arbeit in neuen Forschungsvorhaben weiter auszubauen, verspricht auch in Zukunft besonders praxisrelevante Ergebnisse. Denn Data-Mining ist diskriminierend – eine häufige Aufgabe besteht darin, Unterschiede aufzudecken (vgl. [50, S. 32]). Eine Person, die ausgehend von Daten anders behandelt wird, als eine andere Person, hat das Recht zu erfahren, was die Daten sprechen, auch ohne technisches Verständnis. Jeder, der darüber zu entscheiden hat, personenbezogene Daten herauszugeben, ist ein Entscheidungsträger. Um Entscheidungen treffen zu können, muss ihm erklärt werden, wofür seine Daten verwendet werden (vgl. [50, S. 33]), und zwar in einer Sprache, die er versteht.

A Anhang

Dieser gedruckten Ausgabe der Diplomarbeit ist eine DVD-ROM beigelegt. Sie enthält die Latex-Dateien sowie eine PDF-Version der Diplomarbeit. Außerdem sind die Open-Source-Werkzeuge aus den Data-Mining-Projekten der Einzelfallstudien enthalten. Des Weiteren ist auf der DVD-ROM die Dokumentation der Projekte in Form eines Exports des KDDM-Wikis sowie einer Kopie der erstellten Dokumente aus dem Versionierungssystem. Die Rohdaten wurden anonymisiert und als Beispiele beigelegt.

```
Verzeichnisstruktur auf der DVD-ROM:

Latex        Dateien der Diplomarbeit
Werkzeuge    Werkzeuge der Data-Mining-Projekte
Bachelor     Dokumente und Daten zum Projekt Bachelor
CaseTrain    Dokumente und Daten zum Projekt CaseTrain
KDDM-Wiki    Export des KDDM-Wikis
```

Literaturverzeichnis

[1] Richard Adderley and Peter B. Musgrove. Data mining case study: modeling the behavior of offenders who commit serious sexual assaults. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 215–220, New York, NY, USA, 2001. ACM.

[2] Rakesh Agrawal, Heikki Mannila, Ramakrishnan Srikant, Hannu Toivonen, and A. Inkeri Verkamo. Fast discovery of association rules. In *Advances in knowledge discovery and data mining*, pages 307–328, Menlo Park, CA, USA, 1996. American Association for Artificial Intelligence.

[3] Sarabot S. Anand and Alex G. Buchner. *Decision Support Using Data Mining.* Trans-Atlantic Publications, 1998.

[4] Martin Atzmueller and Florian Lemmerich. Fast subgroup discovery for continuous target concepts. In *Proc. 18th International Symposium on Methodologies for Intelligent Systems (ISMIS 2009), in press.*, 2009.

[5] Martin Atzmuller. *Knowledge-Intensive Subgroup Mining: Techniques for Automatic and Interactive Discovery.* IOS Press, Amsterdam, The Netherlands, The Netherlands, 2007.

[6] Heide Balzert. *Lehrbuch der Objektmodellierung: Analyse und Entwurf.* Lehrbücher der Informatik. Spektrum Akademischer Verlag, Heidelberg, 1999.

[7] Kai Bartlmae and Michael Riemenschneider. Case based reasoning for knowledge management in kdd-projects - concepts, organizational setting, categorization into km and application in the case of knowledge discovery in databases. In *Proceedings of the 3rd International Conference on Practical Aspects of Knowledge Management (PAKM 2000)*, 2000.

[8] Karin Becker and Cinara Ghedini. A documentation infrastructure for the management of data mining projects. *Information & Software Technology*, 47(2):95–111, 2005.

[9] Andreas Bitterer. Who's who in open-source business intelligence. Technical report, Gartner Inc., 2008.

[10] Remco R. Bouckaert, Eibe Frank, and Mark Hall. *WEKA Manual for Version 3-7-0*, 6 2009.

[11] Ronald J. Brachman and Tej Anand. The process of knowledge discovery in databases. pages 37–57, 1996.

[12] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Chapman & Hall, New York, NY, 1984.

[13] Paola Britos, Oscar Dieste, and Ramón García-Martínez. Requirements elicitation in data mining for business intelligence projects. In *Advances in Information Systems Research, Education and Practice*, pages 139–150. Springer Boston, 2008.

[14] Michael Brydon and Andrew Gemino. Classification trees and decision-analytic feedforward control: a case study from the video game industry. *Data Min. Knowl. Discov.*, 17(2):317–342, 2008.

[15] A. G. Buchner, M. D. Mulvenna, S. S. Anand, and J. G. Hughes. An internet-enabled knowledge discovery process. In *International database conference; Heterogeneous and internet databases*. Hong Kong: City University of Hong Kong, 1999.

[16] Joseph Bugajski, Chris Curry, Robert L. Grossman, David Locke, and Steve Vejcik. Data quality models for high volume transaction streams: A case study. In *Proceedings of the Second Workshop on Data Mining Case Studies and Success Stories*. ACM, 2007.

[17] Pete Chapman, Julian Clinton, Randy Kerber, Thomas Khabaza, Thomas Reinartz, Colin Shearer, and Rudiger Wirth. Crisp-dm 1.0 step-by-step data mining guide. Technical report, The CRISP-DM consortium, August 2000.

[18] Bee-Chung Chen, Lei Chen, Yi Lin, and Raghu Ramakrishnan. Prediction cubes. In *VLDB '05: Proceedings of the 31st international conference on Very large data bases*, pages 982–993. VLDB Endowment, 2005.

[19] Lei Chen, Raghu Ramakrishnan, Paul Barford, Bee-Chung Chen, and Vinod Yegneswaran. Composite subset measures. In *VLDB '06: Proceedings of the 32nd international conference on Very large data bases*, pages 403–414. VLDB Endowment, 2006.

[20] Wei Chu, Seung-Taek Park, Todd Beaupre, Nitin Motgi, Amit Phadke, Seinjuti Chakraborty, and Joe Zachariah. A case study of behavior-driven conjoint analysis on yahoo!: front page today module. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1097–1104, New York, NY, USA, 2009. ACM.

[21] Krzysztof Cios, Lukasz Kurgan, Krzysztof J. Cios, and Lukasz A. Kurgan. Trends in data mining and knowledge discovery. In *In: Pal N.R., Jain, L.C. and Teoderesku, N. (Eds.), Knowledge Discovery in Advanced Information Systems*, pages 200–2. Springer, 2005.

[22] Dave DeBarr and Zach Eyler-Walker. Closing the gap: automated screening of tax returns

to identify egregious tax shelters. *SIGKDD Explor. Newsl.*, 8(1):11–16, 2006.

[23] J. C. W. Debuse. Extending data mining methodologies to encompass organizational factors. In *Systems Research and Behavioral Science*, pages 183–190, Maroochydore DC, Queensland 4558, Australia, 2007.

[24] Marcus Deininger, Horst Lichter, Jochen Ludewig, and Kurt Schneider. *Studien-Arbeiten.* Vdf Hochschulverlag, Zürich, 5 edition, 2005.

[25] Andreas Entenmann. Tabellenvirtuosen - Interaktive Datenanalyse mit den Pivot-Funktionen von Excel. *c't*, 9/2009:170ff, 9 2009.

[26] Timm Euler. Publishing operational models of data mining case studies. In *Proceedings of the Workshop on Data Mining Case Studies at the 5th IEEE International Conference on Data Mining (ICDM)*, 2005.

[27] Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy. *Advances in knowledge discovery and data mining.* American Association for Artificial Intelligence, Menlo Park, CA, USA, 1996.

[28] Dirk Friedrich. Business Intelligence etabliert sich im Mittelstand. Technical report, Business Application Research Center - BARC GmbH, 2009.

[29] Prof. Dr. Peter Gluchowski and Christian Schieder. Quelloffene Werkzeuge für Reporting, OLAP und Data Mining im Vergleich. Technical report, Business Application Research Center - BARC GmbH, 2 2009.

[30] P. González-Aranda, Ernestina Menasalvas Ruiz, Socorro Millán, Carlos Ruiz, and Javier Segovia. Towards a methodology for data mining project development: The importance of abstraction. In *Data Mining: Foundations and Practice*, pages 165–178. 2008.

[31] Jiawei Han. Olap mining: An integration of olap with data mining. In *Proceedings of the 7th IFIP 2.6 Working Conference on Database Semantics (DS-7*, pages 1–9, 1997.

[32] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, second edition, 2006.

[33] Bernd Held. *Advanced Controlling mit Excel. Unternehmenssteuerung mit OLAP und PALO.* Franzis, 2006.

[34] Karim K. Hirji. A proposed process for performing data mining projects. In *Managing data mining technologies in organizations: techniques and applications*, pages 88–105, Hershey, PA, USA, 2003. IGI Publishing.

[35] Markus Hofmann and Brendan Tierney. The involvement of human resources in large scale data mining projects. In *ISICT '03: Proceedings of the 1st international symposium on Information and communication technologies*, pages 103–109. Trinity College Dublin, 2003.

[36] Mitja Jermol, Nada Lavraç, and Tanja Urbançiç. Managing business intelligence in a virtual enterprise: A case study and knowledge management lessons learned. *J. Intell. Fuzzy Syst.*, 14(3):121–136, 2003.

[37] Tom Khabaza. Hard hat area: Myths and pitfalls of data mining. Technical report, SPSS Inc., 2007.

[38] Willi Klösgen and Jan M. Zytkow, editors. *Handbook of data mining and knowledge discovery*. Oxford University Press, Inc., New York, NY, USA, 2002.

[39] Ron Kohavi, Llew Mason, Rajesh Parekh, and Zijian Zheng. Lessons and challenges from mining retail e-commerce data. *Mach. Learn.*, 57(1-2):83–113, 2004.

[40] Lukasz Kurgan, Krzysztof Cios, Ryszard Tadeusiewicz, Marek Ogiela, and Lucy Goodenday. Knowledge discovery approach to automated cardiac spect diagnosis. In *Artificial Intelligence in Medicine*, pages 149–169, 2001.

[41] Lukasz A. Kurgan and Petr Musilek. A survey of knowledge discovery and data mining process models. *Knowl. Eng. Rev.*, 21(1):1–24, 2006.

[42] Nada Lavrac, Bojan Cestnik, Dragan Gamberger, and Peter Flach. Decision support through subgroup discovery: Three case studies and the lessons learned. *Machine Learning*, 57(1-2):115–143, October 2004.

[43] R. Lawrence, C. Perlich, S. Rosset, J. Arroyo, M. Callahan, J. M. Collins, A. Ershov, S. Feinzig, I. Khabibrakhmanov, S. Mahatma, M. Niemaszyk, and S. M. Weiss. Analytics-driven solutions for customer targeting and sales-force allocation. *IBM Syst. J.*, 46(4):797–816, 2007.

[44] Jens Lechtenbörger and Gottfried Vossen. Multidimensional normal forms for data warehouse design. *Inf. Syst.*, 28(5):415–434, 2003.

[45] Gordon S. Linoff. *Data Analysis Using SQL and Excel*. John Wiley and Sons, 2007.

[46] Sergio Luján-Mora and Juan Trujillo. Physical modeling of data warehouses using uml. In *DOLAP '04: Proceedings of the 7th ACM international workshop on Data warehousing and OLAP*, pages 48–57, New York, NY, USA, 2004. ACM.

[47] Sergio Luján-Mora, Juan Trujillo, and Il-Yeol Song. A uml profile for multidimensional modeling in data warehouses. *Data Knowl. Eng.*, 59(3):725–769, 2006.

[48] Oscar Marbán, Javier Segovia, Ernestina Menasalvas, and Covadonga Fernández-Baizán. Toward data mining engineering: A software engineering approach. *Information Systems*, 34(1):87 – 107, 2009.

[49] Steve Moyle, Marko Bohanec, and Eric Ostrowski. Large and tall buildings: A case study in the application of decision support and data mining. In Marko Bohanec, Branko Kavšek,

Nada Lavrač, and Dunja Mladenic, editors, *IDDM02*, pages 88–99. Helsinki University Printing House, August 2002.

[50] Robert Nisbet, John Elder, and Gary Miner. *Handbook of Statistical Analysis and Data Mining Applications*. Academic Press, 2009.

[51] Parag C. Pendharkar, editor. *Managing data mining technologies in organizations: techniques and applications*. IGI Publishing, Hershey, PA, USA, 2003.

[52] Gregory Piatetsky-shapiro. Data mining tools used poll. *KDnuggets News*, 12 2009.

[53] Dorian Pyle. *Business Modeling and Data Mining*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.

[54] Dorian Pyle. Nine simple rules you won't want to follow. *DB2 magazine*, 9(1), 2004.

[55] J. Ross Quinlan. Induction of decision trees. In *Proceedings of the First International Conference on Machine Learning*, pages 81–106. Morgan Kaufman, San Mateo, CA, 1986.

[56] Ross Quinlan. *Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[57] Raghu Ramakrishnan and Bee-Chung Chen. Exploratory mining in cube space. *Data Min. Knowl. Discov.*, 15(1):29–54, 2007.

[58] R. Bharat Rao, Sriram Krishnan, and Radu Stefan Niculescu. Improved cardiac care via automated mining of medical patient records. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 12–32, New York, NY, USA, 2001. ACM.

[59] Donald J. Reifer. Software management's seven deadly sins. *IEEE Softw.*, 18(2):12–15, 2001.

[60] Karl Rexer. 2nd annual data miner survey - summary report. 9 2008.

[61] Stefano Rizzi, Alberto Abelló, Jens Lechtenbörger, and Juan Trujillo. Research in data warehouse modeling and design: dead or alive? In *DOLAP '06: Proceedings of the 9th ACM international workshop on Data warehousing and OLAP*, pages 3–10, New York, NY, USA, 2006. ACM.

[62] Heinz Schelle, Roland Ottmann, and Astrid Pfeiffer. *ProjektManager*. GPM, Dt. Ges. für Projektmanagement, Nürnberg, 2. aufl., nachdr edition, 2007.

[63] Larissa Terpeluk Moss Shaku Atre. *Business Intelligence Roadmap: The Complete Project Lifecycle for Decision-Support-Applications*. Addison-Wesley Longman, Amsterdam, 2003.

[64] Il Yeol Song, Ritu Khare, and Bing Dai. Samstar: a semi-automated lexical method for generating star schemas from an entity-relationship diagram. In *DOLAP '07: Proceedings*

of the ACM tenth international workshop on Data warehousing and OLAP, pages 9–16, New York, NY, USA, 2007. ACM.

[65] Dennis Wegener and Michael May. Extensibility of grid-enabled data mining platforms: A case study. In *Proc. of the 5th International Workshop on Data Mining Standards, Services and Platforms*. KDD 2007, 2007.

[66] Jason Westland. *The project management life cycle*. Kogan Page, London, United Kingdom, 2006.

[67] Mark Whitehorn, Robert Zare, and Mosha Pasumansky. *Fast Track to MDX*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[68] Graham Williams. *Rattle: A graphical user interface for data mining in R*, 2009.

[69] Zhenming Xu, Mia Zhang, and Xiaodan Jiang. Business intelligence - a case study in life insurance industry. In *ICEBE '05: Proceedings of the IEEE International Conference on e-Business Engineering*, page 129, Washington, DC, USA, 2005. IEEE Computer Society.

[70] Qiang Yang and Xindong Wu. 10 challenging problems in data mining research. *International Journal of Information Technology & Decision Making (IJITDM)*, 5(04):597–604, 2006.

[71] Robert K. Yin. *Case study research*. Number 5 in Applied social research methods series. Sage, Thousand Oaks, Calif. [u.a.], 4. ed. edition, 2009.

[72] Jose Zubcoff and Juan Trujillo. A uml 2.0 profile to design association rule mining models in the multidimensional conceptual modeling of data warehouses. *Data Knowl. Eng.*, 63(1):44–62, 2007.

[73] José Jacobo Zubcoff and Juan Trujillo. Conceptual modeling for classification mining in data warehouses. In *DaWaK*, pages 566–575, 2006.

Index