Universität Leipzig
Fakultät für Mathematik und Informatik
Institut für Informatik

SPSS Modeler Integration mit IBM DB2 Analytics Accelerator

Master-Thesis

Leipzig, Oktober 2012

vorgelegt von
Markus Nentwig
Studiengang Master Informatik

Betreuender Hochschullehrer:
Prof. Dr. Erhard Rahm
Fakultät für Mathematik und Informatik
Abteilung Datenbanken

Dr. Michael Hartung
Fakultät für Mathematik und Informatik
Abteilung Datenbanken

Externer Betreuer:
Dipl.-Inf. Oliver Benke
IBM Deutschland
Research & Developement GmbH

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

1.1 Motivation

Data-Warehouses finden ihren Einsatz bei der Integration verschiedener Datenquellen auf einer gemeinsamen Plattform. Diese zentrale Anlaufstelle wird dann zur weiteren Verwendung der Daten im Rahmen von analytischen Aufgaben eingesetzt. Die Datenbasis wird dabei mit Hilfe statistischer Methoden in einem Data-Mining-Prozess aufbereitet. Das Ziel dieses Prozesses ist die Erstellung von Modellen wie einer Assoziationsanalyse, welche dann über Regelsätze häufige Assoziationen in der Datengrundlage widergibt. Die berechneten Modelle können dann zur Beurteilung künftiger Entwicklungen genutzt werden, was als Scoring bezeichnet wird. Zum Beispiel kann neuen Kunden darüber eine Produktempfehlung auf Grundlage der Käufe anderer Kunden gegeben werden. Berechnungen direkt auf der Datenbank, bezeichnet als In-Database Analytics, oder intelligente Aktualisierungsstrategien für operationale Datenbestände in das Data-Warehouse sind dabei Entwicklungen, die schneller zu Ergebnissen führen, wodurch neue Einsatzmöglichkeiten wie Echtzeit-Betrugserkennung bei großen Datenmengen ermöglicht werden.

Im Mainframe-Umfeld existiert mit dem IBM DB2 Analytics Accelerator eine Data-Warehouse-Lösung, die durch ihre Appliance-Struktur schnell und einfach einsatzfähig ist. Der massiv parallele Aufbau ist dabei optimal für analytischen Workload geeignet. Durch die tiefe Integration in das bestehende DB2-System sowie der Möglichkeit, In-Database Analytics auszuführen, gibt es viele potentielle Verwendungen, ein Beispiel dafür stellt die nachfolgend präsentierte Integration von Data-Mining-Prozessen in den DB2 Analytics Accelerator dar.

1.2 Zielsetzung

Die vorliegende Arbeit beschreibt einen Architekturansatz, der im Rahmen einer Machbarkeitsstudie bei IBM entwickelt wurde. Dadurch wird der IBM DB2 Analytics Accelerator als eine Data-Warehouse-Appliance dazu in die Lage versetzt, über angepasste Schnittstellen Data-Mining-Modelle über entsprechende Algorithmen direkt auf dem Accelerator zu erstellen. Neben dieser Beschrei-

bung wird die bisherige Verwendung des DB2 Analytics Accelerators sowie das zugehörige Umfeld von Datenbanksystemen bis zum System z Mainframe vorgestellt.
Darauf aufbauend werden praxisnahe Anwendungsfälle präsentiert, die unter Anwendung von intelligenten Methoden auf gespeicherten Kundendaten statistische Modelle erstellen. Für diesen Prozess wird die Datengrundlage zuerst vorbereitet und angepasst, um sie dann in dem zentralen Data-Mining-Schritt nach neuen Zusammenhängen zu durchsuchen. Resultate wie Modelle können dann für weitere Prozesse wie Scoring von neuen Kundendaten verwendet werden. Dieser komplette Lebenszyklus von Data-Mining-Prozessen kann mit Software wie dem IBM SPSS Modeler dargestellt werden, so ausgeführt für die Anwendungsfälle auf der Datenbank DB2 für z/OS. Dieser Phase folgt eine Abbildung der Vorgehensweise unter SPSS Modeler auf den DB2 Analytics Accelerator, wobei für einige Algorithmen explizit untersucht wird, wie eine Portierung auf die neue Plattform stattfinden kann.
Im letzten Teil der Ausarbeitung werden die Resultate der durchgeführten Tätigkeiten aufgezeigt und jeweils für SPSS Modeler und den DB2 Analytics Accelerator vergleichend gegenübergestellt.

1.3 Aufbau der Arbeit

Nachdem in diesem Kapitel das Thema der Arbeit motiviert wurde, folgt in Kapitel 2 ein Überblick der eingesetzten Technologien wie dem Mainframe oder Datenbanksystemen sowie Methoden und Prozesse zur Wissensentdeckung in Data-Warehouse-Systemen. Im Kapitel 3 wird der Stand der Technik dargestellt, wobei Themen aus den Grundlagen neu verknüpft werden und damit eine Einsatzmöglichkeit zeigen. Im Kapitel 4 wird die Umsetzung präsentiert, dabei wird das erlangte Wissen angewandt und Anwendungsfälle von einer bestehenden Architektur auf eine neue abgebildet. Im Anschluß daran werden in Kapitel 5 einige Ergebnisse eingeschätzt und am Ende erfolgt in Kapitel 6 eine Zusammenfassung.

2 Grundlagen

In diesem Kapitel erfolgt eine Vorstellung von Technologien, die im weiteren Verlauf der Ausarbeitung verwendet werden. Dabei wird im Abschnitt 2.1 der Mainframe als zentrale Schaltstelle gezeigt und daraufhin im Abschnitt 2.2 um den Begriff Datenbanksystem erweitert, sowie mit DB2 ein Vertreter vorgestellt. Mit dem allgemeinen Anwachsen der Rechenleistung ist es interessant geworden, neues Wissen aus bereits vorhandenen Datenbanken zu gewinnen, um damit bestehende Prozesse zu optimieren. Diese Wissensentdeckung wird im Abschnitt 2.3 behandelt. Weiterhin gibt es an dieser Stelle eine Einordnung der Begriffe Business Intelligence und Predictive Analytics und abschließend wird IBM SPSS Modeler als mögliches Werkzeug für Data-Mining-Prozesse präsentiert. Im letzten Abschnitt 2.5 wird dann dargestellt, wie ein Data-Warehouse-System bei diesem Prozess helfen kann. Abschließend wird mit dem IBM DB2 Analytics Accelerator eine Data-Warehouse-Appliance beschrieben, die schnell einsatzfähig ist.

2.1 Mainframe IBM System z

In vielen Großunternehmen bilden IBM-Mainframes das zentrale Rückgrat der IT-Infrastruktur und zeichnen sich für ihre *Zuverlässigkeit*, *Verfügbarkeit* und *Wartbarkeit* aus. Diese drei Eigenschaften sind ein zentrales Designziel bei System z und werden etwa über umfangreiche Funktionen zur Systemüberprüfung, die kritische Fehler möglichst zeitig erkennen und melden oder redundante Hardwarekomponenten, die ohne Einfluß auf den Betrieb ersetzt werden können, gewährleistet. Dabei können tausende Programme sowie Ein- und Ausgabegeräte unterstützt werden, um damit gleichzeitig tausenden Benutzern Dienste bereitzustellen. In diesem Kontext kann zudem eine Verfügbarkeit von bis zu 99,999% erreicht werden [EKOO11]. Im Vergleich zur Konkurrenz im Bereich Server Plattformen ist der Mainframe System z zudem seit Jahren am besten bewertet [Spr10, SR11]. Für große Kunden in Geschäftsbereichen wie Bank- oder Versicherungswesen zählen neben den eben genannten Punkten auch die Abwärtskompatibilität für Software zu den Stärken von den IBM Systemen. Dies bedeutet, dass der Maschinencode von Programmen auf älteren als auch aktuellen System z Mainframes ohne Anpassungen lauffähig sind. Auch im Bereich Sicherheit und Skalierbarkeit bietet System z Eigenschaften,

die von Plattformen nicht erreichen [HKS04]. Einige Mechanismen zur Erfüllung dieser Eigenschaften werden im Folgenden kurz vorgestellt.

Parallel Sysplex Bis zu 32 Einzelsysteme können - wie in Abbildung 2.1 dargestellt - miteinander verbunden werden und bieten so mehr Leistung und eine erhöhte Verfügbarkeit, falls eines der Systeme ausfallen sollte. Der Verbund wird über eine zentrale Instanz, die *Coupling Facility (CF)*, gesteuert. Die CF verwaltet unter anderem Locks beim Zugriff auf Ressourcen, steuert die Cache-Verwaltung und verteilt die Workloads mithilfe der eingehenden Status-Informationen. Zudem existiert eine zentrale Zeitgebung über den *Sysplex Timer* für alle im Verbund befindlichen Systeme. [HKS04]

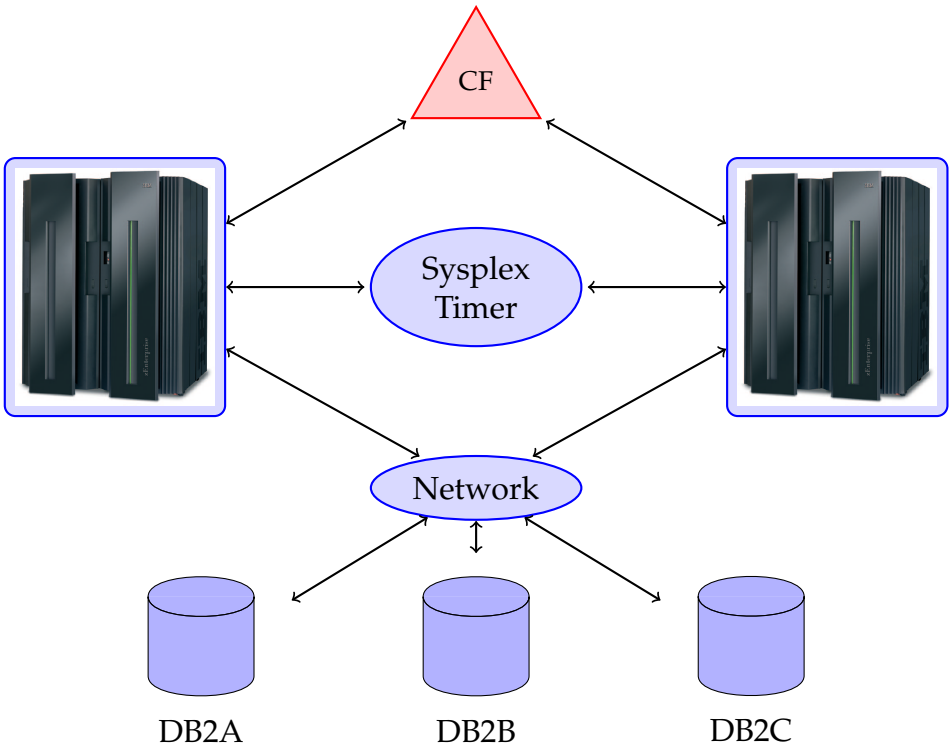

Abbildung 2.1: Ein Parallel Sysplex ist ein Verbund von mehreren System z Maschinen, mit dem die Verfügbarkeit und die Skalierbarkeit weiter steigt. (Quelle: [BAP+06])

GDPS Eine Erweiterung vom Sysplex ist der *Geographically Dispersed Parallel Sysplex*, der es ermöglicht, trotz Totalausfall eines Rechenzentrums, etwa aufgrund einer Naturkatastrophe, den Betrieb an entfernter Stelle aufrecht zu erhalten. Detailliertere Informationen dazu sind in [KCPS12] veröffentlicht.

Betriebsysteme System z unterstützt verschiedene Betriebssysteme, die jeweils auf einer *logischen Partition (LPAR)* residieren. Wichtige Beispiele sind hier *z/OS*, *z/VM* und *Linux on System z*, unter z/VM wiederum können verschiedene weitere Betriebssysteminstanzen laufen. Über eine TCP/IP-Verbindung (*HiperSocket*) kann ein Datenaustausch zwischen Betriebsystemen in getrennten LPARs auf einer Maschine über den jeweiligen Hauptspeicher stattfinden. [HKS04]

WLM Die Verwaltung von Workload findet auf einem System z über den integrierten *Workload Manager (WLM)* statt. Dieser ermöglicht es, verschiedene Workloads auf mehreren Instanzen von z/OS dynamisch innerhalb festgelegter Grenzen zu steuern. Dabei werden Prioritäten für bestimmte Aufgaben explizit beeinflusst, um Service Level Agreements, etwa für Antwortzeiten bei Datenbankanwendungen, zu erfüllen. Dafür werden Prozesse in verschieden wichtige Service-Klassen eingeteilt, worüber der Workload über Ziel-Definitionen dynamisch gesteuert werden kann. Der WLM steuert zudem auch Ressourcen bei der Verwendung von einem Parallel Sysplex in Zusammenarbeit mit der Coupling Facility. [CDF+08]

Sicherheit Auf System z existieren verschiedene Mechanismen, um die Sicherheit von Daten und die Trennung von verschiedenen Betriebssystemen auf einer Maschine zu gewährleisten. Dazu zählt RACF *Resource Access Control Facility*, was ein Sicherheitsframework für System z darstellt, wodurch Funktionen zur Authentifizierung und Autorisierung von Nutzer und Ressourcen, also zur Zugriffskontrolle sowie zur Protokollierung aller Vorgänge bereitstellt werden. Diese werden zudem auch von Subsystemen wie CICS oder DB2 für eine bessere Transaktionssicherheit eingesetzt. Weiterhin ist der *PR/SM (Processor Resource/System Manager)* als Hypervisor dafür verantwortlich, die verfügbare Hardware logisch aufzuteilen und somit jedem laufenden Betriebssystem begrenzte Ressourcen zuzuteilen. Festgelegte Datenpartitionen werden dabei als LPAR bezeichnet, das entsprechende Betriebssystem kennt nur die ihm zugewiesene Hardware und hat keinen Zugriff auf andere Systeme. [HKS04]

Channel Subsystem Das *Channel Subsystem* entlastet die zentralen Prozessoren (CP), indem es den Datenfluss zwischen dem Hauptspeicher und Ein-/Ausgabegeräten koordiniert. Die Besonderheit dabei ist, dass über ein Channel Subsystem bis zu 256 I/O-Geräte angebunden werden können, die dann direkt von einer LPAR genutzt werden. Damit können Verbindungen zu externen Geräten aufrecht erhalten werden und zeitgleich von den CPs weitere Operationen bearbeitet werden. [HKS04, EKOO11]

2.2 Datenbanksysteme

Die heutige Informationsgesellschaft baut auf Computersystemen auf, die Daten bereitstellen und abspeichern. In vielen Unternehmen und Organisationen gehört die Verwaltung von Datenbeständen zu den wichtigsten Herausforderungen - die Umsetzung erfolgt in der Regel mithilfe von Datenbanksystemen. Ein Datenbanksystem ist ein Software-Produkt, welches zum einen die *Datenbank* mit den Nutzerdaten beinhaltet, zum anderen ein *Datenbankmanagementsystem (DBMS)*, um bestimmte Anforderungen beim Umgang mit der Datenbank zu erfüllen. Durch diese Struktur wird eine zentrale Verwaltung der Daten losgelöst von möglichen Anwendungen erreicht. Ein wichtiges Konzept bei der Arbeit mit Datenbanksystemen ist das *Transaktionsparadigma*, welches Mechanismen zur Wahrung von Qualität und Korrektheit einzelner Transaktionen auf der Datenbank beschreibt. Eine Transaktion ist dabei eine modellhafte Abbildung von Ereignissen wie einer Geldüberweisung oder einer Ausleihe in einer Bibliothek. Das Paradigma fordert, dass jede Transaktion den folgend kurz beschriebenen *ACID-Eigenschaften* genügt, was mit wachsender Datenbankgröße und steigenden Zugriffszahlen durch verschiedene Benutzer neue Herausforderungen mit sich bringt. [HR99]

Atomicity Es gibt zwei Möglichkeiten, wie eine Transaktion verläuft: Entweder wird sie ohne Probleme ausgeführt oder beim Auftreten von Fehlern nicht durchgeführt. In diesem zweiten Fall werden eventuell durchgeführte Teilschritte wieder rückgängig gemacht. [HR99]

Consistency Mit der Konsistenz-Eigenschaft wird gewährleistet, dass nach einer Transaktion alle Integritätsbedingungen erfüllt sind und die Datenbank insgesamt wie vor der Trankaktion in einem widerspruchsfreien Zustand ist. Bei einer Banküberweisung bleibt etwa die Summe der beiden beteiligten Konten gleich. Zudem wird mit dieser Eigenschaft die korrekte physikalische Speicherung der Daten zugesichert. [HR99]

Isolation Diese Eigenschaft bietet trotz mehreren Nutzern, die auf die Datenbank zugreifen wollen, jedem einzelnen Nutzer eine logische Trennung und gewährleistet damit parallele Verwendung der Daten. Dafür werden etwa Verfahren zur Synchronisierung wie Sperr-Mechanismen implementiert. [HR99]

Durability Eine einmal vollständige ausgeführte Transaktion kann durch nachfolgende Fehler nicht mehr korrumpiert werden, das heißt, sie ist dauerhaft. [HR99]

Für einen produktiven Einsatz müssen zusätzlich Aspekte wie etwa Verfügbarkeit und Fehlertoleranz betrachtet werden [HR99]. Zur Unterscheidung von Datenbanksystemen können Kriterien wie die Art der Externspeicheranbin-

dung oder Rechnerkopplung herangezogen werden, womit drei Klassen gebildet werden können:

Shared-Everything Bei dieser Klasse greifen alle Prozessoren des Rechners auf einen gemeinsamen Arbeitsspeicher zu. Zudem existiert nur eine Betriebssysteminstanz, wodurch das DBMS auf die komplette Datenbank zugreifen kann. Dieser Ansatz skaliert für Installationen ab einer bestimmten Größe schlechter als die beiden folgenden Architekturansätze. [Rah94]

Shared-Disk wird auch als Database Sharing bezeichnet. Die wichtigste Eigenschaft ist hier, dass jedes DBMS wieder auf die gesamte Datenmenge zugreifen kann. Allerdings existieren mehrere Rechner, die über eine Cluster-Implementierung umgesetzt sind. Dabei kann die Rechnerkopplung lose (jeder Prozessor hat eigenen Hauptspeicher, Kooperation über voneinander unabhängige Systeme) oder nah (eigener Hauptspeicher pro Prozessor sowie gemeinsamer Hauptspeicher für alle Prozessoren) sein. Vor allem bei naher Rechnerkopplung profitieren alle Prozessoren von einer vereinfachten Kommunikation etwa bei Kontrollaufgaben, somit lässt sich in diesem Verbund die Verteilung von Workload am besten durchführen. [Rah94]

Shared-Nothing Auch Database Distribution. Auf mehreren lose gekoppelten Rechnern (jeder Prozessor hat hat eigenen Hauptspeicher) wird je ein DBMS verwendet, um auf einen exklusiven Teil der Gesamtdatenmenge zuzugreifen. Dadurch bietet ein Shared-Nothing-System die beste Erweiterbarkeit / Skalierbarkeit, weil keine Kommunikation bei parallel stattfindendem Datenzugriff auf verschiedenen Teilsystemen nötig ist. [Rah94]

Der Zugriff auf angebundene Daten erfolgt dabei über das Clustermodell *Data Sharing*, das heißt, jedes System kann auf alle Daten und jedes andere System direkt zugreifen, wenn es die Rechte dafür besitzt.

DB2 für z/OS ist ein Vertreter der relationalen Datenbanksysteme. DB2 zieht Vorteile aus den verfügbaren technischen Komponenten wie dem Parallel Sysplex (siehe Abbildung 2.1) oder dem Workload Manager (WLM), so können etwa über *Shared Data* mehrere DB2-Subsysteme (Instanzen) auf mehreren Mainframes zusammengefasst werden, die dann über die Coupling Facility (CF) auf gemeinsame Buffer Pools zugreifen können. Shared Data ist dabei eine erweiterte Form des Shared-Disk-Ansatzes, wobei die CF bei parallelen Zugriffen für eine bessere Nebenläufigkeit und Synchronisierung der Prozesse sorgt. Mit dem Shared Data Prinzip kann zudem eine erhöhte Verfügbarkeit sowie Lastbalancierung erreicht werden [BAP+06]. Die Buffer Pools der CF arbeiten dann als zentraler Cache zwischen dem DB2 und den physikali-

schen Platten. Die zudem für Verwaltungsaufgaben zuständige CF entlastet die Einzelsysteme durch die Behandlung von Sperren und der Kohärenzkontrolle. Gleichzeitig kann mit dem WLM der anfallende Workload entsprechend eingestellter Zielvereinbarungen gesteuert werden, womit Engpässe verhindert werden. In einem DB2-Subsystem werden bestimmte logische und physikalische Strukturen bereitgestellt, mit deren Hilfe die Abbildung von Benutzerdaten auf Festplatten erfolgt. Die logischen Konstrukte lassen sich grob in Tabellen für die eigentlichen Datensätze und Datenbanken zur Gruppierung von Table-/Indexspaces unterteilen, die physikalischen Datenstrukturen werden nachfolgend eingeführt. [EKOO11, HKS04, Rah94]

Tablespace Ein Tablespace beinhaltet eine oder mehrere Tabellen. Von dem logischen Konstrukt Tabelle erfolgt mit dem Bezug zu einem Tablespace die Zuordnung zu einem physikalischen Datensatz.

Indexspace Für jeden Index in einer Tabelle wird ein Indexspace außerhalb der Tablespaces angelegt.

Storage Group Die Storage Group beinhaltet normalerweise alle einer Datenbank zugehörigen Konstrukte (Table-/Indexspaces) und wird physikalisch einem oder mehreren Laufwerken zugeordnet.

Metadaten wie Logs, der Datenbankkatalog oder sonstige Parameter werden nicht pro Datenbank gespeichert, sondern zentral im zugehörigen DB2-Subsystem. Auf System z existieren ausgereifte Hilfsprogramme zur DB2 Administration, diese werden unter anderem zur Datenverwaltung, zum Ausführen von Backup- oder Recovery-Aufgaben oder zur Konsistenzprüfung verwendet. Im Hintergrund werden zur Verwendung der Tools JCL-Skripte einsetzt, die über das Job Entry System (JES) Subsystem in z/OS ausgeführt werden. [EKOO11]

2.3 Wissensentdeckung in Datenbanken

Dieser Abschnitt beschäftigt sich vor allem mit den begrifflichen Einordnungen beim Durchsuchen von Datenbanken nach bisher verborgenen Zusammenhängen. Im Englischen als Knowledge Discovery in Databases (KDD) und synonym oft als Data-Mining bezeichnet, beschreibt der Prozess das Vorgehen, Daten vorzubereiten und durch die Anwendung von Algorithmen zu untersuchen, um schlußendlich die neu gewonnenen Informationen zu visualisieren und weiterzuverwenden. Dieses Vorgehen beschreibt zudem den Prozess der Datenanalyse passend, wobei das Ziel dieser Analyse die Wissensentdeckung darstellt. Dieses Ziel lässt sich mit der Ausführung folgender Schritte nach [HK00] und [AMH08] erreichen:

1. Initiale Überprüfung der Daten

2. Benötigte Datenquellen miteinander kombinieren
3. Für die Analyse relevante Daten auswählen
4. Data-Mining im eigentlichen Sinne - Anwendung von intelligenten Methoden, um Muster zu erkennen
5. Muster-Auswertung mit Hilfe weiterer Algorithmen und Metriken
6. Präsentation des erarbeiteten Wissens über Möglichkeiten der Visualisierung und Wissensrepräsentation

Diese Punkte umfassen wiederum im einzelnen verschiedene Teilaufgaben, welche dadurch mehr oder weniger aufwendig in der Umsetzung sind. Bei der initialen Prüfung können Inkonsistenzen wie fehlende Werte sowie Rauschen oder extreme Werte aus den Rohdaten gefiltert oder angepasst werden. Zudem können verschiedene Datenquellen wie transaktionale Datenbanken, komplexe Data-Warehouse-Konstrukte oder aber das World Wide Web zur Auswahl stehen, bei der Mustererkennung gibt es unterschiedliche Algorithmen, etwa Entscheidungsbäume oder Assoziationsmodelle, die jeweils ihren optimalen Einsatzbereich haben. All diese Punkte haben entscheidende Auswirkungen auf eine Datenanalyse und sollten mit Bedacht gewählt werden. [HK00, AMH08]

Eine weitere methodische Vorgehensweise wird durch den *Cross Industry Standard Process for Data Mining (CRISP-DM)* beschrieben. Dabei werden sechs Phasen unterschieden, die wie in Abbildung 2.2 aufgezeigt miteinander verknüpft sind. So etwa müssen für bestimmte Modellierungsschritte die Daten in der vorherigen Stufe speziell angepasst werden oder aber die Bewertung der Modelle ergibt ein neues Verständnis für die auszuführende Aufgabe. Der oben beschriebene Data-Mining-Schritt mit der Anwendung von intelligenten Methoden ist dabei im CRISP-DM die Phase der Modellierung. Unteraufgaben sind dabei die Auswahl eines passenden Modells gefolgt von Testläufen, ob das Modell geeignet ist. Nach erfolgreichem Test kann das Modell mit den richtigen Parametern erstellt werden, wonach eine erste Einschätzung der Ergebnisse und eventuell nötige Neuerstellung vorgenommen werden kann. Die letzte Phase, der Einsatz, beschreibt dabei die tatsächliche Verwendung der Ergebnisse, um Prozesse zu verbessern. Dieser Einsatz kann in seinem Umfang stark variieren, eine Möglichkeit ist, dass ein einfacher Report über die erzielten Ergebnisse verfasst wird. Eine weitere Möglichkeit ist, dass die Resultate in weitere Prozesse integriert werden, berechnete Modelle können so etwa zur Bewertung von neuen Daten verwendet werden. [CCK+00] Diese Verwendung wird als *Scoring* bezeichnet. Mit Hilfe von Modellen wird demnach eine Grundlage zur Bewertung von neuen Datensätzen geschaffen. Je nach Anforderung, wie schnell die Ergebnisse über ein Scoring bewertet werden sollen, gibt es unterschiedliche Umsetzungen. Ein Einsatz für Scoring in Echtzeit ist etwa, dass bei einem Einkauf aufgrund der gewählten Produkte ein Rabatt-Coupon auf die Rechnung passend zum Einkaufsverhalten gedruckt wird. [LK12, BL07] Im

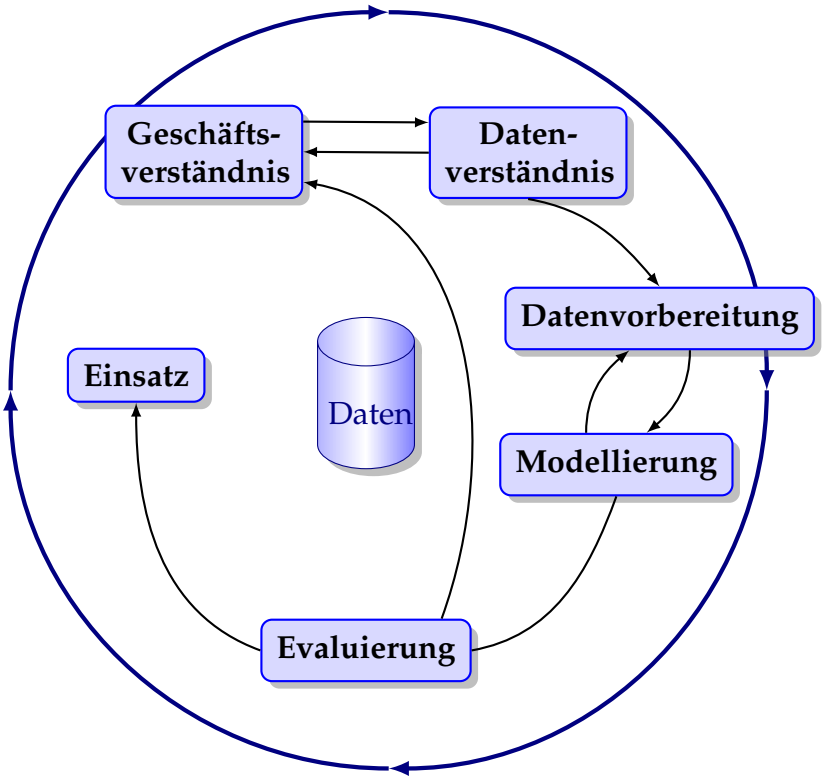

Abbildung 2.2: CRISP-DM Referenz-Modell, welches sechs ineinandergreifende Schritte des allgemeinen Verständnisses und der Datenbearbeitung zeigt. (Quelle: [CCK+00])

Gegensatz dazu wird beim Batch Scoring eine größere Menge an Datensätzen bewertet, dies erfolgt in der Regel nach der Ausführung der Transaktionen, so zum Beispiel können an einer Börse alle Transaktionen der letzten Stunde auf Unregelmäßigkeiten überprüft werden. Bei Batch Scoring wird das Ergebnis demnach nicht sofort benötigt, dafür auf großen Datenmengen berechnet, wohingegen beim Echtzeit-Scoring eine schnelle Reaktion von einer Anwendung gefordert wird, allerdings nur auf einem speziellen Datensatz. [Hof12]

Dem allgemeinen Vorgehen folgt die Einordnung, welche Art von Zusammenhängen in den Daten gesucht werden soll. Dies kann etwa allgemeine Äußerungen über den Datenbestand betreffen, der ohne technische Unterstützung nicht zu erkennen wäre, also *beschreibende* Aussagen über die bestehenden Daten, was im Geschäftsumfeld Prozesse aus dem Bereich Business Intelligence kennzeichnet. Andererseits geht es auch darum, mit den vorhandenen Daten *vorhersagende* Aussagen über neue Transaktionen oder zukünftige Entwicklungen zu treffen, was in Unternehmen in den Bereich Predictive Analytics eingeordnet werden kann. Eine kurze Beschreibung von Business Intelligence respektive Predictive Analytics erfolgt im folgenden Abschnitt. [HK00]

2.3.1 Business Intelligence und Predictive Analytics

In [KMU06] wird Business Intelligence wie folgt beschrieben:

> Unter Business Intelligence (BI) wird ein integrierter, unternehmensspezifischer, IT-basierter Gesamtansatz zur betrieblichen Entscheidungsunterstützung verstanden.

Dieses Zitat zeigt deutlich auf, dass BI als kompletter Prozess verstanden wird, der mit Methoden wie Data-Mining und Systemen wie einem Data-Warehouse als Hilfsmittel und damit über den Zugriff auf möglichst viele Informationen im Unternehmen Möglichkeiten bietet, bestehende Prozesse zu optimieren. Zudem zeigt die Definition den starken Fokus auf Technik, speziell geht es um die Optimierung von Geschäftsprozessen mit Informationstechnik. [KMU06]

Predictive Analytics erweitert den bestehenden Begriff des BI um die analysierenden Komponenten, die mit Informationen aus dem Unternehmen und Data-Mining-Konzepten Erkenntnisse über zukünftige Entwicklungen gewinnen. Dabei liegt der Fokus auf der Erstellung von Modellen, mit welchen dann möglichst exakte Vorhersagen getroffen werden können. [Koe12] Mit dieser kurzen Einführung wird im nächsten Abschnitt mit SPSS Modeler ein Vertreter aus dem Bereich Predictive Analytics vorgestellt.

2.3.2 IBM SPSS Modeler

Als mögliches Werkzeug für das eben vorgestellte Verfahren Predictive Analytics wird im Folgenden SPSS Modeler als eine Data-Mining-Workbench von IBM vorgestellt. SPSS Modeler wird zudem für die bearbeiteten Szenarien benötigt, um mit der jeweiligen Datengrundlage Data-Mining durchführen zu können. Dabei können mehrere Datenbanken miteinander verknüpft werden, die Informationen aufbereitet und daraufhin mit diesen Datensätzen neue Modelle erstellt werden. Diese können eingesetzt werden, um zum Beispiel Vorhersagen über mögliche Betrugsfälle im Versicherungsbereich zu liefern oder um nach Herzanfällen entsprechend der medizinischen Vorgeschichte eine optimale Behandlung zu leisten. Die wichtigsten Bestandteile des SPSS Modeler sind zum einen der Modeler Client, beschrieben im nächsten Abschnitt, sowie der Modeler Server, der im Hintergrund eine zentrale Verarbeitungsstelle für die Daten darstellt, näher erläutert in Abschnitt 2.3.2.2. Zudem existiert mit dem Component-Level Extension Framework (CLEF) eine Grundlage zur Erstellung von Erweiterungen, diese wird im Abschnitt 2.3.2.3 beschrieben.

2.3.2.1 IBM SPSS Modeler Client

Der SPSS Modeler Client ist die grafische Benutzeroberfläche für den SPSS Modeler, mit welcher über verschiedene Arbeitsschritte Daten aus mehreren Quellen analysiert werden können. Arbeitsabläufe werden über diverse Knoten, verbunden durch gerichtete Kanten, in Streams als eine Art Flußdiagramm abgebildet, eine einfache Variante inklusive der Bedienoberfläche ist in Abbildung 2.4 exemplarisch zu sehen. Über die untere Menüleiste können aus verschiedenen Paletten Modellierungs- und Analyseknoten oder Datenbankoperationen ausgewählt werden und zu Streams kombiniert werden. Je nach Ziel des Streams bestehen Teilaufgaben des Modeler Clients aus Datenvorbereitung, Modellerstellung und Modellscoring. Die Vorbereitung der Daten umfasst dabei zum Beispiel das Einbinden von Datenbanken oder Dateien, in Abbildung 2.3 das erste Icon. Weiterhin sind die folgenden beiden Icons in der Abbildung der Datenvorbereitung zuzuordnen, in SPSS Modeler wird dieser Knotentyp auch Feldfunktion oder -operation genannt. Dabei werden etwa Datentypen angepasst, zusammengeführt oder Rollen für eine folgende Modell-Erstellung vergeben. Die zweite Zeile von Knoten in der Abbildung 2.3 zeigt zuerst ein Knoten zur Modellberechnung, der dann ein „goldenes“ und damit berechnetes Modell folgt. Der letzte Knoten zeigt dann einen Ausgabeknoten in eine Tabelle einer Datenbank. In Abbildung 2.4 ist somit eine Datenvorbereitung zu sehen, an deren Ende mit dem Knoten `RESPONSE` die Modellberechnung stattfindet. Dies geschieht mit Trainingsdaten, bei denen das Ergebnis bereits bekannt ist, um das Modell möglichst gut zu trainieren. Die zweite Zeile

zeigt dann einen Scoring-Prozess, der mit dem vorher berechneten Modell Datensätze bewertet und die Ergebnisse in eine Tabelle schreibt. [sps12b, sps11c]

Abbildung 2.3: SPSS Modeler Übersicht verschiedene Knoten

2.3.2.2 IBM SPSS Modeler Server

Zentrale Aufgaben des SPSS Modeler Server sind die Durchführung von Berechnungen, die in Streams genutzt werden, und die Bereitstellung von Erweiterungen sowie Diensten. So können Modelle direkt über entsprechende Schnittstellen auf z/OS DB2 berechnet und bereitgestellt werden. Diese können dann über den SPSS Modeler Server Scoring Adapter genutzt werden, um Transaktionen direkt aus dem OLTP-Umfeld heraus über das Modell zu bewerten (Scoring). [ECRS12] Im Modeler Server werden zudem die Datenquellen eingebunden, sodass nicht jeder Client einzeln Daten erfragen muss. Zwei verschiedene Ansätze spiegeln einen typischen Einsatz des Modeler Servers wider. Dies ist zum einen die Nutzung des Servers auf dem gleichen Rechner, auf dem auch der Modeler Client installiert ist. In diesem Fall wird SPSS Modeler für kleinere Datenanalysen verwendet, etwa in kleineren Unternehmen oder bei geringem Datenvolumen. Der andere Ansatz ist eine Installation mit mehreren Modeler Clients, die über einen Modeler Server auf die Daten zugreifen. Dieser Server befindet sich dann zum Beispiel auf einem virtualisierten Linux auf System z und hat damit einen schnellen Zugriff auf angebundene Daten. [sps11d]

2.3.2.3 IBM SPSS Modeler CLEF

Das Component-Level Extension Framework (CLEF) ist eine Möglichkeit, die Funktionalität des SPSS Modelers um benutzerspezifische Aufgaben zu ergänzen. Eine Verwendung von CLEF kann sein, neue Nodes oder Menüpunkte im

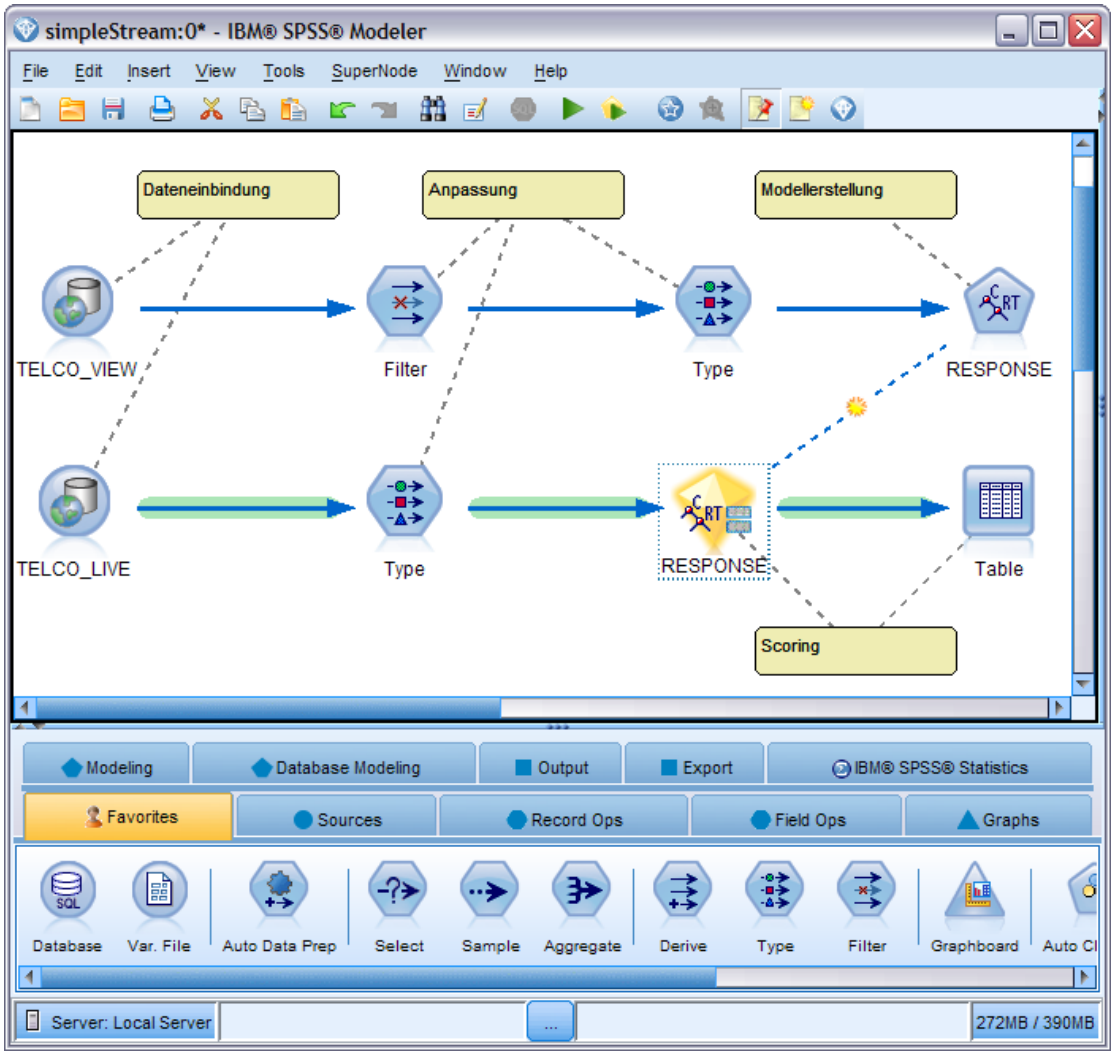

Abbildung 2.4: IBM SPSS Modeler Benutzeroberfläche mit einfachem Stream, mit welchem nach der Datenaufbereitung ein Modell für Kunden im Telekommunikationsbereich erstellt wird. In der zweiten Zeile wird das berechnete Modell dann auf neue Kundendaten angewandt (Scoring) und beispielhaft in einer Tabelle ausgegeben.

SPSS Modeler anzubieten, um bestehende Funktionen zu erweitern oder um neue Modelle zu unterstützen. Die Architektur ist wie der SPSS Modeler ein Client/Server-Modell, sowohl die Client- als auch Server-Komponenten sind in Abbildung 2.5 dargestellt. Im einfachsten Fall wird für eine Erweiterung nur eine Spezifikationsdatei benötigt. Über die Java API sowie der Verwendung eigener Java-Klassen können zudem größere Änderungen an der Benutzeroberfläche durchgeführt werden. Schon mit der Installation von IBM SPSS Modeler kommen viele CLEF-Erweiterungen zum Einsatz, so etwa Modelle wie K-Nearest-Neighbor (KNN), Neuronales Netz, Cox-Regression oder Bayessches Netz. Diese Modelle werden wie jede andere Berechnung auf dem Server ausgeführt, wofür serverseitig Erweiterungen in C/C++ (Shared Libaries) vorhanden sein müssen, welche die Umsetzung steuern beziehungsweise die Berechnung durchführen. Verwaltungsaufgaben können etwa sein, dass der Modeler Server den Client anfragt, welches Modell erstellt werden soll oder das nach der Modellerstellung nicht mehr benötigte Objekte gelöscht werden. [sps12c]

2.4 Data-Mining-Modelle

In IBM SPSS Modeler können auf der Grundlage von Daten neue Zusammenhänge gefunden werden, die dann als Modelle eingesetzt werden, um Prozesse zu optimieren. Im weiteren Verlauf der Arbeit werden einige Modelle verwendet, über welche in diesem Abschnitt ein Überblick gegeben werden soll. Dabei werden zuerst zwei Algorithmen zur Erstellung von Entscheidungsbäumen erläutert, wonach zwei Methoden zur Berechnung von Regelsätzen für eine Entscheidungsanalyse vorgestellt werden.

2.4.1 Entscheidungsbaum

Mit Entscheidungsbäumen kann eine automatische Klassifizierung von Datenobjekten mit Hilfe von Entscheidungsregeln, welche an den Knotenpunkten im Baum überprüft werden, durchgeführt werden. Mit einem fertig erstellten Entscheidungsbaum können demnach neue Daten schnell in bestimmte Klassifikationen eingeordnet werden. Unterschieden werden dabei binäre und nicht-binäre Bäume, bei binären Entscheidungsbäumen erfolgt eine Beschränkung auf zwei Auswahlmöglichkeiten als Ergebnis einer Regel. Entscheidungsbäume finden in vielen Data-Mining-Szenarien Verwendung, da Sachverhalte mit vielen Eigenschaften gut abgebildet werden können, die dann zusätzlich von Menschen intuitiv zu handhaben sind. Die Erstellung von Entscheidungsbäumen kann entweder manuell erfolgen oder über Algorithmen, die einen Entscheidungsbaum induzieren. Dabei gibt es unterschiedliche Methoden wie CHAID von Kass [Kas80] oder CART von Breiman et al. [BFOS84].

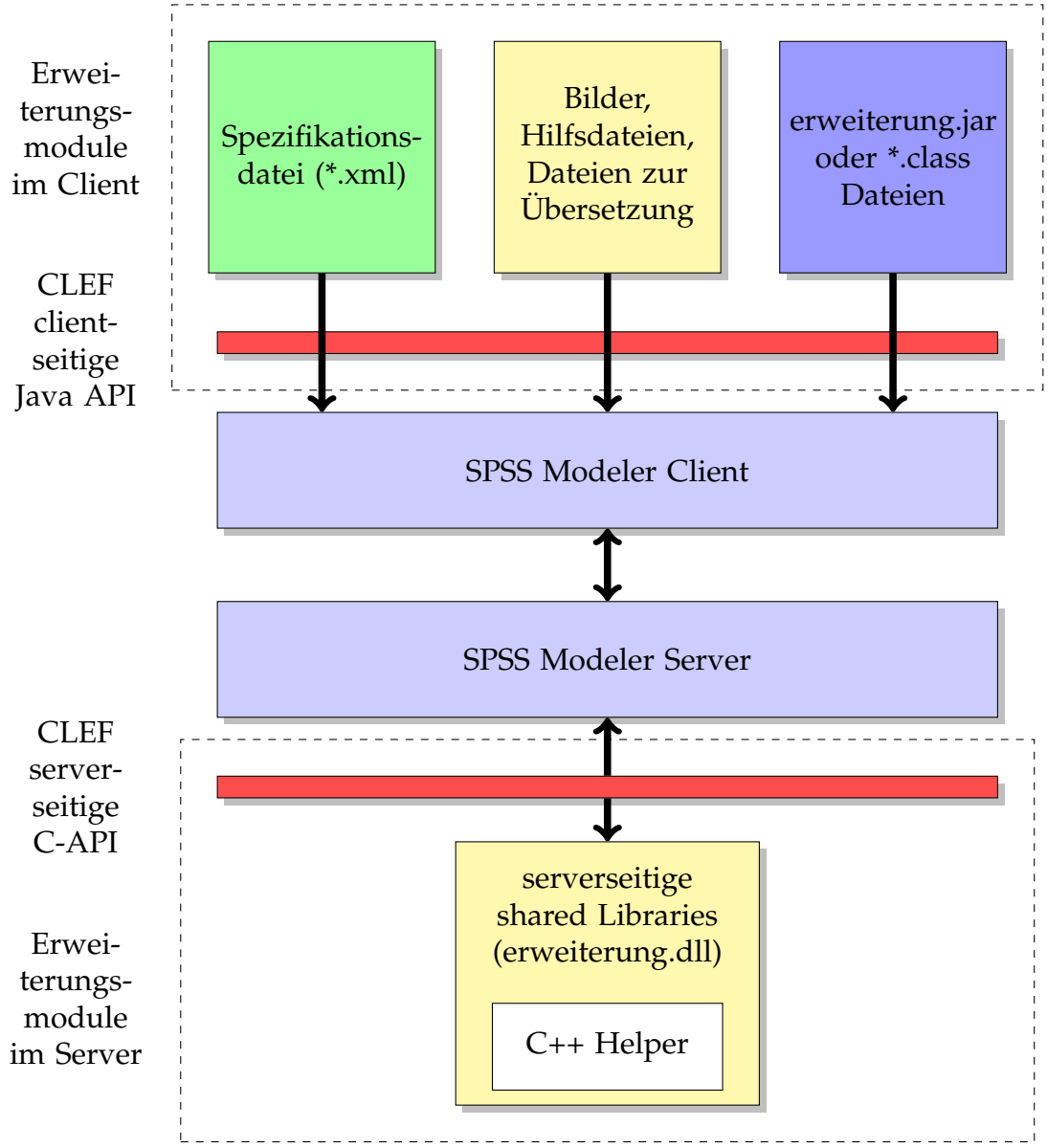

Abbildung 2.5: Client- / Server-Komponenten der CLEF-Architektur: Die clientseitigen Erweiterungsmodule kommunizieren über die Java API mit dem Client, welcher die nötige Kommunikation mit dem Server aufrecht erhält. Dieser wiederum verständigt sich serverseitig mit einer C API, um Node-Funktionen auszuführen. (Quelle: [sps12c])

CHAID ist ein Vertreter zur Erstellung nicht-binärer Bäume und wird oft im Marketing eingesetzt, auf Basis von CART hingegen werden ausschließlich binäre Entscheidungsbäume erstellt. In jedem Falle ist für die Erstellung eines Entscheidungsbaumes eine möglichst große Menge von Trainingsdatensätzen nötig, welche viele Parameter zur Einschätzung des Szenarios beinhaltet. Beispielsweise sollten für ein Kreditrating die Einkommenszahlen, die Anzahl der Kreditkarten und der Erfolg oder Misserfolg über die Rückzahlung vermerkt sein. Der Vorgang zur Erstellung ähnelt sich bei CHAID und CART, der Baum wird jeweils von oben herab rekursiv aufgebaut, dabei wird an jedem Knoten über ein Selektionsmaß das Attribut auswählt, welches die Daten am besten aufteilt. In den darunterliegenden Knoten wird aus den verbliebenen Attributen erneut das statistisch relevanteste gewählt und dadurch die Datenmenge erneut aufgeteilt. Dies erfolgt entweder solange, bis keine Attribute mehr zur Auswahl stehen oder durch die Wahl der maximalen Tiefe des Baumes, nachdem der Algorithmus stoppen soll. [HK00, sps12d]

2.4.1.1 CHAID

CHAID (Chi-squared Automatic Interaction Detectors) prüft alle zu bewertenden Attribute über den χ^2-Test und ermittelt darüber das Attribut mit der höchsten statistischen Signifikanz, welches damit an dieser Stelle den optimalen Parameter zur Aufteilung für den nächsten Knoten darstellt. Eine Stärke von CHAID ist, dass nicht-binäre Bäume erstellt werden können. Die Beliebtheit im Bereich Marketing stammt daher, dass die Aufteilung in mehrere Unterkategorien (etwa 4 Einkommenskategorien) auf einer Ebene des Baumes für die Einteilung in Marktsegmente genutzt wird. [sps12d, HL06]

2.4.1.2 CART

In dem Buch *Classification and Regression Trees (CART)* von Breiman et al. [BFOS84] wird die Generierung von binären Entscheidungsbäume beschrieben. Die Auswahl für die nächste Aufteilung findet durch die Berechnung eines statistischen Maßes, entweder der Gini-Koeffizient oder über den mittleren Informationsgehalt. Die Wahl des Attributes mit dem höchsten Informationsgehalt versucht dabei etwa, die Daten mit hohem Informationsgehalt zuerst einzuordnen, damit danach weniger Information fehlt, um die Klassifizierung abzuschließen. Nach der Erstellung des Baumes können über Pruning (oder Beschneiden) des Baumes unerwünschte Anomalien oder eine zu hohe Anpassung auf die Gegebenheiten der Testdaten (Overfitting) ausgefiltert werden. Diese Aufgabe wird am besten mit Transaktionen durchgeführt, die nicht in den Testdaten vorhanden sind. Der resultierende Baum ist dann oft einfacher als ein nicht beschnittener Baum. [HK00, HL06]

2.4.2 Assoziationsanalyse

Treten bestimmte Datensätze in einer Menge von Transaktionen regelmäßig zusammen auf, etwa beim Kauf von einem Handy *und* einer Handytasche oder bei einer Patientenuntersuchung Symptome wie Halsschmerzen *und* Fieber, lassen sich diese zu Regeln zusammenfassen, die zu neuen Erkenntnissen in Datenbeständen führen können. Die Erstellung dieser Regelsätze auf einer Datenmenge innerhalb festgelegter Grenzen wird als Assoziationsanalyse bezeichnet, Einsatzbereiche sind zum Beispiel das Finden von neuen Zusammenhängen in Datenbeständen oder eine unterstützende Analyse für eine Klassifizierung von Daten. Bei der Suche nach Regeln sind zwei Parameter besonders wichtig. Zum einen der *Support*, der angibt, für welche Prozentzahl der Transaktionen die Regel gilt und zum anderen die *Konfidenz*, die aussagt, bei wieviel Prozent der Käufer die Wahl „Handy“ zu „Handytasche“ führt. Zusätzlich werden in der Regel für eine Analyse der minimale Wert für Support und Konfidenz angegeben, damit zur Berechnung eine untere Grenze für etwaige Regeln feststeht. Die Menge der gefundenen Regeln wird auch als häufige Itemsets bezeichnet. Ein einzelnes Itemset ist also eine Anzahl von Items, die zusammen häufig auftreten. Für die Durchführung der Assoziationsanalyse können verschiedene Algorithmen verwendet werden, wofür im Folgenden zwei Vertreter vorgestellt werden. [HK00]

2.4.2.1 Apriori

Bereits 1994 von Agrawal et al. [AIS93] vorgestellt, findet Apriori in zwei Teilschritten häufige Regelsätze. Zuerst erfolgt die Generierung von Kandidaten, beginnend mit Itemsets der Länge $k = 1$. Demnach werden alle häufigen Items als Kandidaten behandelt, welche die Bedingungen für den minimalen Support erfüllen. Mit *diesen* Itemsets erfolgt die Suche nach Itemsets der Länge $k + 1$, was solange durchgeführt wird, bis keine neuen häufigen Itemsets mehr gefunden werden können. In jedem Unterschritt werden Werte aussortiert, die den minimalen Support nicht erreichen. Ab $k \geq 2$ wird das Wissen über die zurückliegend $(k - 1)$ als häufig befundenen Kandidaten in einem zweigeteilten Subprozess verwendet, um die Kandidaten der folgenden Stufe (k) zu ermitteln.

- *Join* der Kandidaten aus $k - 1$ mit sich selbst.
- *Prune* (Aussortieren) von möglichen Kandidaten, die Teilmengen enthalten, welche nicht in denen aus Schritt $k - 1$ bekannt waren.

Dieser Prozess wird so lange fortgeführt, bis keine neuen Kandidaten gefunden werden können. Mit den berechneten Kandidaten werden alle Transaktionen in der Datenbank nach Kandidaten wie folgt erneut durchsucht. Jede Fundstelle eines jeden Kandidaten in einer beliebigen Teilmenge einer Transaktion wird

gezählt und am Ende wird die Anzahl der Fundstellen pro Kandidat genutzt, um das Support-Kriterium zu überprüfen.
Nach der Generierung der Kandidaten kann der zweite Teilschritt ausgeführt werden. Dies umfasst die Erstellung der tatsächlichen Regeln mit der Prüfung, ob der Mindestwert für die Konfidenz erreicht wird. Da die Konfidenz die Wahrscheinlichkeit widergibt, mit welcher nach Item A auch Item B folgt, kann der Wert über die Bedingte Wahrscheinlichkeit $P(B|A)$ ermittelt werden. [HK00, AIS93]

$$\text{Konfidenz}(A \Rightarrow B) = P(B|A) = \frac{\text{Support}(A \cup B)}{\text{Support}(A)} \tag{2.1}$$

Diese Gleichung 2.1 wird für alle Regeln angewandt, die auf folgende Art und Weise bestimmt werden:

- Berechne für jeden Kandidaten k die nichtleeren Teilmengen t
- Gib für jedes $t \in k$ die Regel „$t \Rightarrow k \setminus t$“ an, wenn gilt:

$$\frac{\text{Support}(k)}{\text{Support}(t)} \geq \text{Minimum Konfidenz} \tag{2.2}$$

Das Ergebnis ist die Menge aller Regelsätze mit den geforderten Eigenschaften auf der gegebenen Datengrundlage. [HK00]

2.4.2.2 FP-growth

Der Ansatz für den *FP-growth-Algorithmus (Frequent-pattern-growth)* ist aus der Erkenntnis heraus entstanden, dass die Generierung von Kandidaten bei Apriori umfangreich ausfallen kann. Wenn bei Apriori eine große Menge an Kandidaten existiert, muss für viele Kandidaten überprüft werden, ob sie Teilmengen enthalten, die nicht im vorherigen Schritt bekannt waren. Das Ziel des FP-growth ist es daher, die Generierung von Kandidaten allgemein zu vermeiden. Im ersten Schritt wird analog zum Apriori das Vorkommen von einzelnen Items in der Datenbank vermerkt, die entstehende Liste L wird absteigend, beginnend mit dem häufigsten Eintrag gespeichert. Darauf aufbauend wird jede Transaktion aus der Datenbasis in einen FP-tree nach folgendem Schema einsortiert. Die einzelnen Posten einer Transaktion werden nach ihrer Position in der Liste L in einen Baum einsortiert, immer beginnend bei der Wurzel, die den Wert „null“ hat. Wenn in einem ersten Schritt A, B und C einsortiert werden sollen, ist A das Kind von „null“, B das Kind von A und schließlich C von B. Jedem Knoten wird die Häufigkeit des Vorkommens angeheftet, an dieser Stelle für jeden Knoten 1. Als nächstes soll die Transaktion mit A, B, D und C einsor-

tiert werden. Da A schon als Kind unter der Wurzel existiert, wird hier nur die Häufigkeit des Vorkommens auf 2 erhöht, was zudem auch für B zutrifft. D ist bisher nicht als Kind von B eingetragen, wird demnach neu angelegt und mit 1 bewertet. Das letzte Item C existiert zwar schon in einem benachbarten Pfad, allerdings wird es unter D neu als Kind erzeugt und das Vorkommen in diesem Pfad mit 1 bewertet. Nachdem Einfügen aller Transaktionen in den Baum werden in der Liste L alle vorkommenden Stellen des jeweiligen Items als Referenz angegeben. Anstatt Kandidaten zur Erstellung von Regeln zu verwenden, erfolgt das Mining jetzt über die Suche von Mustern in dem FP-tree. Beginnend mit dem seltensten Wert s aus der Liste L werden alle Pfade zur Wurzel notiert, die das Support-Kriterium erfüllen. Diese Pfade werden pro Wert in L als neuer, bedingter FP-tree gespeichert und geben Regeln wider, die s enthalten. [HK00]

2.5 Data-Warehouse-Systeme

Zu Beginnn von diesem Abschnitt soll eine Definition des Begriffs nach einem Zitat von Inmon [Inm96] den Grundgedanken von einem Data-Warehouse widerspiegeln:

> A data warehouse is a subject-oriented, integrated, time-variant, nonvolatile collection of data in support of management's decision-making process.

Relevant ist somit einerseits der Begriff *subject-oriented* als Anwendungsbezug auf die jeweilige Umgebung, in der ein Data-Warehouse eingesetzt werden soll. Mit *integrated* wird der Hinweis gegeben, dass verschiedene Datengrundlagen in eine gebündelte und angepasste Datenbank aufgenommen werden. Weiterhin wird mit *time-variant* ausgedrückt, dass Daten über einen längeren Zeitraum vorliegen sollen und somit historische Analysen ermöglicht werden. Letztlich der Ausdruck *nonvolatile*, welcher nahelegt, dass die Daten vom transaktionalen System getrennt werden und somit nach dem Laden auf das Data-Warehouse in der Regel nicht mehr geändert werden [HK00]. Ergänzt um Funktionen zur Datenintegration und -analyse entsteht das *Data-Warehouse-System*, wobei die Vorgehensweise oft ist, die Daten einmal pro Woche aus den Quellsystemen zu importieren und dann darauf viele lesende Zugriffe laufen zu lassen, etwa zur Erstellung von Berichten. Dieser Prozess der Datenbeschaffung wird allgemein auch als ETL-Prozess bezeichnet und findet in einer vom operationalen Geschäft unabhängigen Staging Area (auch Arbeitsbereich) statt, um bestehende Vorgänge nicht zu beeinflussen. Die nachfolgenden drei Schritte beschreiben den ETL-Prozess näher nach [BG08]:

Extract Die Extraktion der geforderten Daten erfolgt aus den Quellsystemen über festgelegte Schnittstellen wie ODBC. An dieser Stelle finden die ersten Überprüfungen statt, ob die Daten korrekt erfasst worden sind. Aufgrund der Komplexität des ETL-Prozesses wird dieser Schritt meist nur periodisch durchgeführt.

Transform Dieser Arbeitschritt beinhaltet die Anpassung der Quelldaten an das Zielformat, beispielsweise mit der Anpassung von Datentypen oder Datumsangaben an die Data-Warehouse-Umgebung. Zudem fällt in diesen Schritt die Überprüfung der Daten auf eventuell fehlende oder doppelt vorhandene Werte, auch können veraltete Datensätze anders behandelt oder potentiell fehlerhafte Daten aussortiert werden.

Load Der abschließende Lade-Prozess überführt die vorbereiteten Datensätze in das Data-Warehouse. Dies kann durch den Aufbau einer historischen Struktur erfolgen, etwa durch eine blockweise Speicherung pro Import (stündlich, täglich). Eine zweite Möglichkeit ist, die alten Daten zu verwerfen und nur die neuen Datensätze zu speichern. [BG08]

Der Aufbau eines Data-Warehouse-Systems dient somit vor allem dem Bearbeiten von analytischen Datenbankanfragen (OLAP, Online Analytical Processing) und unterscheidet sich mit diesem Fokus stark von transaktionalen Systemen (OLTP, Online Transaction Processing). Aus diesem Grund werden in Tabelle 2.1 einige grundsätzliche Unterschiede zwischen beiden Anfragetypen. Resultierend daraus ist für transaktionalen Workload auf einem Datenbanksystem die Shared-Disk-Architektur optimal geeignet. Zum Beispiel kann die Verteilung von Aufgaben besser auf einem Shared-Disk-System gesteuert werden, auch sind bei einem Ausfall von Prozessoren keine Daten unerreichbar und eine parallele Ausführung von Anfragen ist zudem auch einfacher zu administrieren. Wenn bestimmte Daten, die von einem Rechner in einem Shared-Nothing-System verwaltet werden, überlastet ist, gibt es keine Möglichkeit, die Belastung zu verringern. Für analytischen Workload auf einem Data-Warehouse ist der Shared-Nothing-Ansatz jedoch der interessantere, weil ein paralleler und unabhängiger Zugriff auf die Daten hinter einem Rechner stattfinden kann. Bei umfangreichen Anfragen können für die vorhandenen Datensätze auf jedem Teilsystem Aufgaben durchgeführt werden. [Rah94]

Die Anbieter von Data-Warehouse-Systemen erreichen die geforderten Eigenschaften mit dem Einsatz verschiedener Techniken wie spezieller Hardware oder Optimierung der SQL-Ausführung. Auch gibt es verschiedene Schwerpunkte der Hersteller wie Skalierbarkeit, Flexibilität oder Unterschiede in der Geschwindigkeit, neue Daten in das Data-Warehouse zu übernehmen. Demnach gibt es bei der Konzeption von Data-Warehouse-Systemen je nach Zielsetzung und Anforderungen mehr oder weniger geeignete Kandidaten. [BG08]

Vergleich	transaktional	analytisch
Aktion auf Datenbank	vor allem Schreiben, wenig Lesen	vor allem Lesen, Hinzufügen
Struktur der Anfrage	kurz, einfach struk turiert	lang, komplex
Anzahl Anfragen	viele	wenige
Verarbeitete Datensätze pro Abfrage	geringe Anzahl	große Anzahl
Eigenschaften der Daten	zeitaktuell, dynamisch	historisch, stabil
Anwender	viele Sachbearbeiter	wenige Analysten
Antwortzeit	im Sekundenbereich oder weniger	eher Minuten - Stunden

Tabelle 2.1: Unterschiede bei transaktionalen und analytischen Datenbankanfragen, basierend auf [BG08]

2.6 Data-Warehouse-Appliance

Data-Warehouse-Appliances stellen für Kunden eine Komplettlösung dar, die vordefinierte Software und Hardware einsetzt, um die Konzeption zur Erstellung von einem Data-Warehouse zu erleichtern. Im Detail bedeutet das, dass eine solche Appliance ein System aus Hardware und Software zum Data-Warehousing darstellt, um in diesem Gesamtpaket ohne weitere Ergänzungen benutzbar zu sein. Die Hardware umfasst dabei neben Servern auch die kompletten Storage-Kapazitäten, wonach die Preisgestaltung der Data-Warehouse-Appliances erfolgt. Anbieter von Data-Warehouse-Appliances wie Teradata oder IBM setzen bei der technischen Umsetzung auf eine massiv parallele Berechnung (MPP-System) und verwenden dabei einen Shared-Nothing Rechnerverbund [Bal, BBF+12]. Abgerundet werden diese Pakete durch Dienstleistungen etwa bei Inbetriebnahme und Wartung und in diesem Gesamtpaket zudem damit beworben, leichter zu administrieren und schneller einsatzbereit zu sein als bisherige Data-Warehouse-Systeme. [BFAE12]
Als Beispiel für diese Appliances wird im Folgenden der IBM DB2 Analytics Accelerator (IDAA) vorgestellt.

2.6.1 Übersicht IDAA

Anhand von Abbildung 2.6 soll als erster Schritt die Sicht von außen auf ein System z196 mit angeschlossenem IDAA gezeigt werden. Dabei wird deutlich, dass der Beschleuniger über 10 Gbit Netzwerkanschlüsse mit dem Mainframe verbunden ist. Aus Anwendungs- und Benutzersicht ist der DB2 Analytics Accelerator nicht zu erkennen, er ist demnach von extern gesehen vollkommen transparent. Dies bringt unter anderem den Vorteil mit sich, dass der Accelerator in die durch z/OS vorhandenen Sicherheitskonzepte eingebunden wird. Weiterhin muss die Sicherheit der Daten durch die alleinige Anbindung an das System z nur auf diesem einen System gewährleistet werden. werden muss. Die Administration erfolgt über das Eclipse-basierte IDAA Data Studio, welches über Schnittstellen im DB2 auf den IDAA zugreifen kann. Welche Query über den DB2 Analytics Accelerator beschleunigt werden oder ansonsten vom DB2 selbst besser bearbeitet werden, entscheidet das DB2 automatisch nach vorgegebenen Regeln. [BBF+12]

Abbildung 2.6: Übersicht der IDAA-Komponenten: Anwendungen greifen ohne Änderung über das DB2 auf die Daten zu, im Hintergrund werden bestimmte Queries durch Netezza beschleunigt. Die Administration des DB2 Analytics Accelerator erfolgt über das Programm IDAA Data Studio. (Quelle: [BBF+12])

Nach dieser allgemeinen Einführung werden einzelne Komponenten genauer beschrieben, wofür Bezug auf Abbildung 2.7 genommen wird. Zu erkennen ist in der Abbildung der zentrale Punkt für die Ausführung von Queries: DB2 auf z/OS. Von der Anwendung an die DB2 API abgesetzt, wird jede eingehende Datenbankanfrage direkt weitergereicht an den DB2 Optimizer, wo abgeschätzt wird, auf welchem System die Anfrage geringere Bearbeitungskosten erzeugt. Demnach ist der Optimizer für die Verteilung von Queries nach vorgegebenen

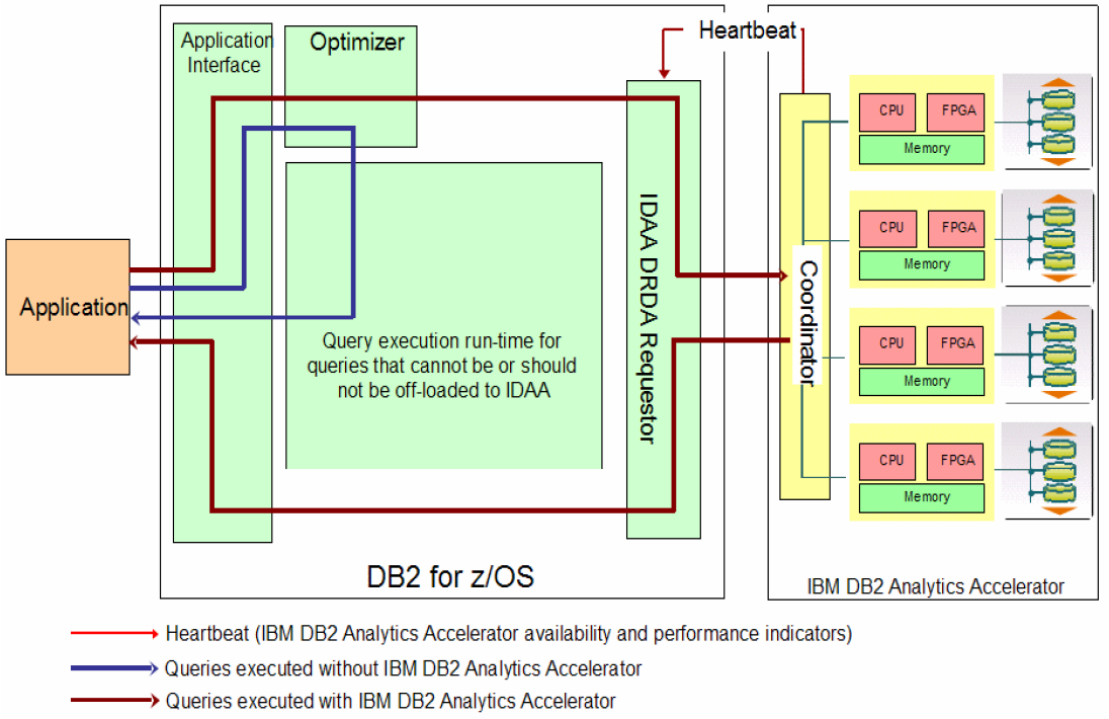

Abbildung 2.7: Weg von SQL Queries durch ein z/OS DB2, dabei trifft der DB2 Optimizer die Entscheidung, ob die Query über den DB2 Analytics Accelerator beschleunigt wird (Quelle: [BBF+12]).

Kriterien verantwortlich und entscheidet, ob die Ausführung vom DB2 selbst übernommen wird, oder ob sie an den DB2 Analytics Accelerator weitergeleitet wird. Dabei werden etwa Voraussetzungen wie das Vorhandensein eines Beschleunigers überprüft und zudem, welche Tabellen bereits vorhanden und damit „accelerated“ sind, um dann mithilfe von Heuristiken festzustellen, ob die Query für IDAA geeignet ist. Komplexe analytische Queries auf Faktentabellen sind ein Beispiel, welches vom Accelerator besonders stark profitieren kann, im Gegensatz dazu sind Anfragen auf einzelne, über Indices optimierte Tabellen weniger geeignet für die massiv parallele Bearbeitung auf dem IDAA und werden daher direkt auf DB2 ausgeführt. Direkt vom DB2 bearbeitete Queries wer-

den nach der Optimierung über die jeweiligen Zugriffspfade ausgeführt und das entsprechende Ergebnis über die DB2 API an die ursprüngliche Anwendung geschickt. Datenbankanfragen, bei denen mit einer Beschleunigung gerechnet wird, werden noch auf der DB2 auf z/OS durch den IDAA DRDA[1] Requestor an den IDAA Server Coordinator (auch SMP Host) weitergeleitet, welcher die einzige Schnittstelle zwischen dem Mainframe und der IDAA Erweiterung selbst darstellt. Insgesamt ist durch diese Aufteilung des Workloads eine Umgebung für gemischten Workload, wie etwa OLTP- und OLAP-Workload, geschaffen. Die weitere Verarbeitung findet dann auf der Erweiterung selbst statt und wird im nächsten Abschnitt näher betrachtet. [BBF+12]

2.6.2 IDAA im Detail

Der Aufbau des DB2 Analytics Accelerator ist in drei Bereiche untergliedert, die in Abbildung 2.8 zu sehen sind. Bei der Hardware wird dabei vor allem auf Technologie der von IBM aquirierten Netezza Corporation gesetzt, was im Folgenden näher erläutert wird. Alle eingehenden Queries werden im grün markierten Bereich *SMP Hosts* in einem ersten Schritt verarbeitet. Die SMP Hosts, aufgrund ihrer Aufgabenstellung auch DB2 Analytics Accelerator Server oder Coordinator genannt, steuern die Query-Verteilung und sind wie bereits in Abschnitt 2.6.1 beschrieben für den Datenaustausch mit dem z/OS DB2 verantwortlich, außerdem wird darüber das System administriert. Die Verteilung der Queries erfolgt auf den in der Abbildung darunterliegenden blauen Bereich mit seinen *Snippet Blades*, von diesen auch Worker Nodes genannten Blades kann ein Accelerator bis zu zwölf Stück besitzen. Jedes dieser Snippet Blades wiederum beherbergt acht CPU Cores sowie acht FPGAs sowie lokalen Hauptspeicher. Desweiteren ist jeder Worker Node exklusiv mit acht Festplatten aus dem grauen Bereich *Disk Enclosures* verbunden, er kann demnach nur auf Daten auf diesem Verbund zugreifen, was dem Cluster-Ansatz *Shared-Nothing* entspricht. Diese unabhängig voneinander arbeitenden Snippet Blades werden im Verbund als *Assymetric Massive Parallel Processing*-System bezeichnet, da die Aufgaben von den SMP Hosts verteilt werden und auf jedem Snippet Blade eigener Hauptspeicher sowie Shared-Nothing Zugriff auf den angeschlossenen Speicher existiert. Diese parallele Verarbeitung ist vor allem für OLAP-typischen Workload mit komplexen Anfragen auf große Datenmengen nützlich. [Rah94, BBF+12]

Als nächstes wird die Verarbeitung von Queries auf dem Snippet Blade beschrieben. Die erste Aufgabe ist es, die komprimierten Daten vom angeschlos-

[1] Distributed Relational Database Architecture, Standard für Interoperabilität bei Datenbanken von The Open Group: `http://collaboration.opengroup.org/dbiop/` (zuletzt aufgerufen am 13.10.2012)

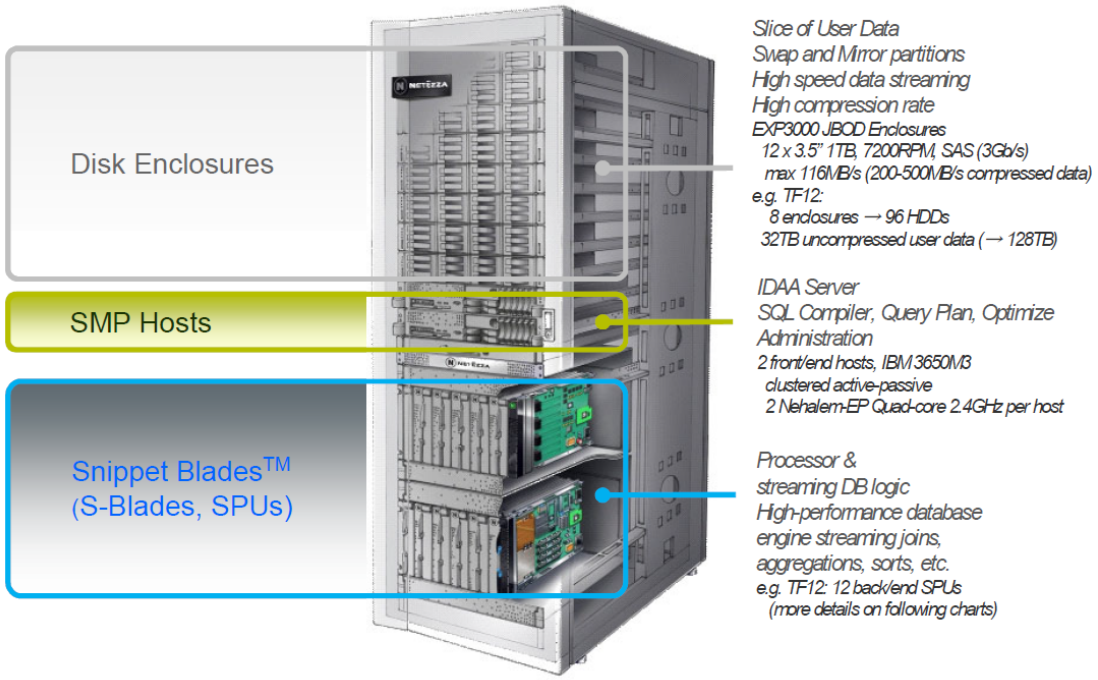

Abbildung 2.8: Technische Grundlagen im IBM DB2 Analytics Accelerator, die SMP Hosts in der Mitte sind für die zentrale Verteilung der Aufgaben auf die darunterliegenden Snippet Blades zuständig. Snippet Blades wiederum erledigen die Arbeit mithilfe FPGA-Hardwarebeschleunigung und sind jeweils exklusiv pro Blade an einen Verbund von acht Festplatten angeschlossen (Quelle: [BBF+12]).

senen Festplattenverbund zu lesen und im Hauptspeicher zu cachen. Ab hier erfolgt die Bearbeitung der Daten in mehreren parallelen Streams, was in Abbildung 2.9 schematisch für einen der acht FPGA pro Snippet Blade dargestellt ist. Der FPGA holt die Daten per Direct Memory Access aus dem Hauptspeicher und dekomprimiert diese. Danach beschränkt der FPGA die Spalten und Zeilen entsprechend der `SELECT` und `WHERE` Angaben. Dekomprimieren und Beschränken der Spalten und Zeilen sind somit spezielle Aufgaben, auf deren Bearbeitung der FPGA konfiguriert ist. Durch die hardwarebasierte Vorverarbeitung im FPGA operiert die CPU auf einer stark reduzierten Datenmenge und erledigt komplexe Aufgaben wie Joins und Aggregationen, die nicht im FPGA implementiert sind. [BBF+12]

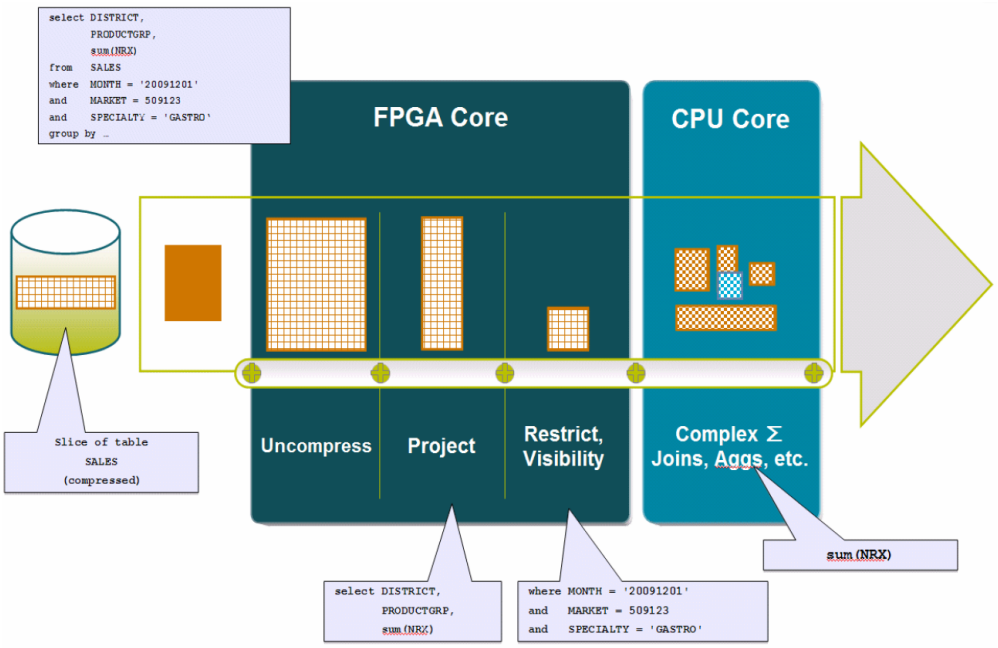

Abbildung 2.9: Query Bearbeitung in einem Snippet Blade, auf jedem der acht Cores wird ein Teil der Daten bearbeitet, dabei übernimmt jeweilig der FPGA das Dekomprimieren und das Beschränken der Daten auf die benötigten Zeilen und Spalten. Danach kann die CPU die überbleibenden komplexen Aufgaben auf reduziertem Ausgangsmaterial durchführen (Quelle: [BBF+12]).

2.6.3 IBM Netezza Analytics

IBM Netezza Analytics (INZA) gibt externen Anwendungen die Möglichkeit, direkt auf der Netezza-Hardware analytische Aufgaben durchzuführen. Mit

diesen auch *In-Database Analytics* genannten Funktionalitäten können etwa Modellierungs- oder Scoringschritte in SAS oder IBM SPSS massiv parallel auf dem DB2 Analytics Accelerator berechnet werden. Zudem bietet INZA eine Entwicklungsumgebung für eigene Erweiterungen an, mit denen Aufgaben wie MapReduce auf Hadoop auf IDAA berechnet werden können. Eine Übersicht der unterstützten Programme und Frameworks, die auf Netezza Analytics zurückgreifen können, ist in Abbildung 2.10 zu sehen. Die Funktionsweise ist in jedem Fall, dass Anwendungen Stored Procedures in DB2 auf z/OS aufrufen, welche den Befehl mitsamt den Parametern weiterleiten an Netezza. Dort wird eine weitere Stored Procedure aufgerufen, die dann die entsprechenden INZA-Befehle ausführt. [Man09]

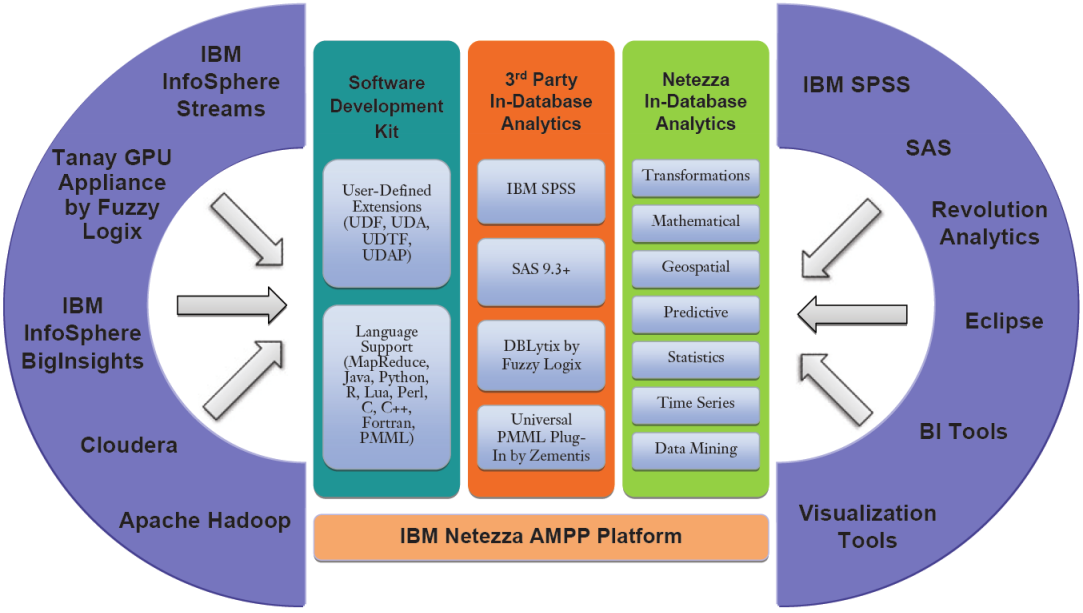

Abbildung 2.10: IBM Netezza Analytics Paket (INZA), verwendet zur Ausführung von analytischen Funktionen direkt auf dem DB2 Analytics Accelerator. (Quelle: [Man09])

2.6.4 Performance IDAA

Für eine Einschätzung der Performance von IDAA wird auf einen Bericht zurückgegriffen, der für eine Beschleunigung eines SAP NetWeaver Business Warehouse erstellt wurde [RM11]. Auf einer Datenbasis von 2 und 18 Millionen Datensätzen wurde dabei getestet, welche Leistungsteigerungen bei typischen Queries wie massiven Aggregationen oder vielen Restriktionen erreicht

werden können. Bei der kleineren Datengrundlage wurde im Schnitt eine Beschleunigung von Faktor 9 erreicht, bei 18 Millionen Datensätzen erreicht der Test eine 73x schnellere Bearbeitung. Aufgrund der hoch parallelen Architektur der IDAA ist zudem zu erwarten, dass bei noch größeren Datenmengen bessere Ergebnisse erzielt werden. Ein zweiter Test wurde mit Hilfe zufällig erzeugter Datenbankanfragen auf einer Basis von 6 Millionen Datensätzen durchgeführt, wobei die Steigerung der Performance im Durchschnitt den Faktor 12 ergeben hat. In einer zweiten Arbeit für ein großes Versicherungsunternehmen beträgt der Geschwindigkeitszuwachs für verschiedene typische Business-Reporting-Anfragen auf bis zu 813 Millionen Datensätzen Faktor 13 bis 1908 [Pes12].

3 Stand der Technik

Nach der Beschreibung von grundlegenden Technolgien und Vorgehensweisen im letzten Kapitel wird darauf aufbauend ausgeführt, wie das Zusammenspiel zwischen den beteiligten Komponenten aussehen kann. Im ersten Abschnitt 3.1 werden aus diesem Grund zwei verschiedene Umgebungen für einen typischen Einsatz von SPSS Modeler in Unternehmen vorgestellt. Die dabei gezeigten Probleme und Schwächen werden im Abschnitt 3.2 als Herausforderung für eine Machbarkeitsstudie gesehen, die untersucht, inwiefern Modellierungsprozesse im IBM SPSS Modeler mit Hilfe eines DB2 Analytics Accelerator beschleunigt werden können. Im Zuge dessen wird der dabei entstandene Prototyp zur Umsetzung beschrieben und auf bestehende Herausforderungen eingegangen. Letztendlich werden im Abschnitt 3.3 zwei praxisnahe Szenarien vorgestellt, die für eine Performance-Messung zum einen auf Basis existierender Technologie und zum anderen unter Verwendung des Prototypen genutzt werden sollen. Im weiteren Verlauf der Arbeit werden diese Szenarien zudem genutzt, um nötige Anpassungen für die Abbildung auf den DB2 Analytics Accelerator zu untersuchen.

3.1 Modellierung auf Basis existierender Technologie

In diesem Abschnitt sollen zwei verschiedene Ansätze präsentiert werden, wie in Unternehmen oder Organisationen OLAP-typische Aufgaben wie Data-Mining durchgeführt werden. Dazu wird im ersten Teil 3.1.1 eine auf System z aufbauende Umgebung gezeigt, in welcher SPSS Modeler unter Linux auf System z direkt auf die per Data Sharing eingebundenen Ressourcen zugreifen kann. Der zweite Teil 3.1.2 stellt dann eine Lösung dar, in der über ETL-Prozesse eine Auslagerung der Daten in ein externes Data-Warehouse-System erfolgt.

3.1.1 SPSS Modeler auf System z

SPSS Modeler läuft in diesem Szenario auf Linux for System z, was in Abbildung 3.1 aufgezeigt wird. Dort ist zu sehen wie der Zugriff auf das DB2

und damit den angeschlossenen Ressourcen von SPSS Modeler aus per HiperSocket ermöglicht wird. Demnach führt SPSS Modeler in diesem Ansatz analytische Queries direkt über DB2 aus. Über Data Sharing erfolgt dabei zum Beispiel Lastbalancierung, welche durch die Coupling Facility (CF) gesteuert wird. In bestimmten Szenarien kann es dazu kommen, dass OLAP-Workload nicht mit der höchsten Priorität bearbeitet wird. Die Beeinflußung und Steuerung der Ressourcen erfolgt dabei zum Beispiel über den z/OS WLM. [BAP+06] Wenn das OLTP-System etwa zusätzliche Ressourcen braucht, werden OLAP-Aufgaben auf „DB2 B“ beschränkt und laufen mit begrenzten Ressourcen. Ein weiteres Szenario kann sein, dass OLAP-Aufgaben allgemein künstlich beschränkt werden, wodurch komplexer analytischer Workload viel Zeit zur Ausführung benötigt. Nach langen Berechnungen werden die Ergebnisse unter Umständen nicht mehr benötigt, die Nutzung der DB2-Rechenzeit muss jedoch troztdem bezahlt werden. Außerdem ist für historische Daten, die nicht mehr geändert werden, die Datenhaltung in einem transaktionalem Datenbanksystem nicht optimal. Dieses Szenario wird im Kapitel 5 für den vergleichenden Performance-Test verwendet.

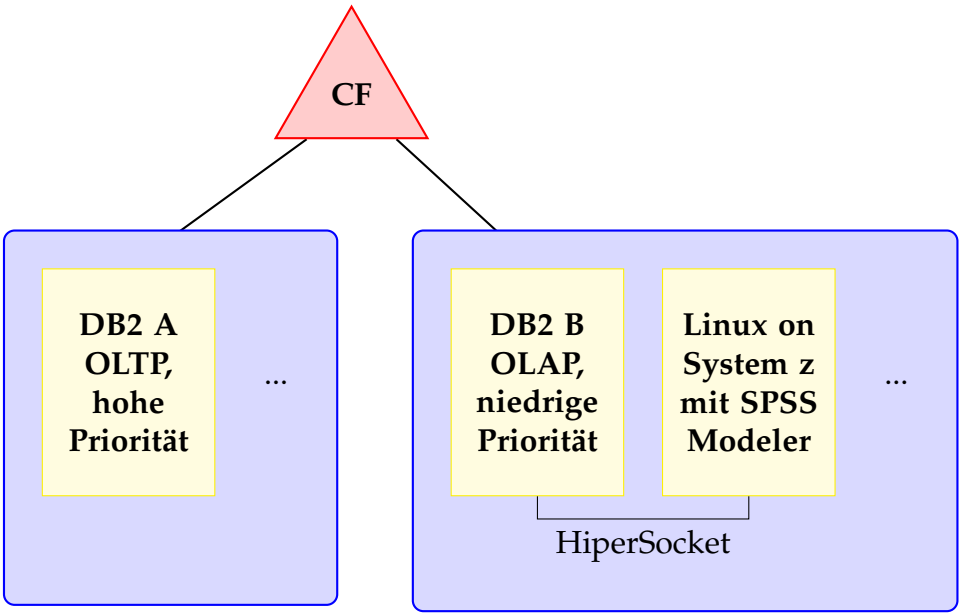

Abbildung 3.1: Modellierung auf System z über SPSS Modeler auf Linux on System z, dabei werden analytische Aufgaben über niedrige Prioritäten bei der Lastbalancierung beschränkt, wenn bestimmte Bedingungen eintreten. [BAP+06]

3.1.2 Externes Data-Warehouse-System

3.2 In diesem zweiten Szenario werden die Daten per ETL-Prozess aus dem bestehenden transaktionalen System z in ein externes Data-Warehouse-System übertragen. Dieses System wird dann zur Ausführung von analytischen Aufgaben verwendet, wobei es den Vorteil hat, dass es nach der Überführung in das Data-Warehouse ohne Beeinflußung durch transaktionalen Workload arbeiten kann. Jedoch gibt es einige Nachteile, die im weiteren erläutert werden. Der ETL-Prozess ist mit seiner Anpassung an das neue Data-Warehouse-System mit seinen Gegebenheiten eine große Herausforderung, der zudem dazu führt, dass der Prozess nur bei Bedarf oder in gewissen Abständen stattfindet. Somit wird oft auf veralteten Daten gearbeitet. Weiterhin ist eine Rückführung von Ergebnissen mit einem erneuten Transformationsprozess verbunden. Durch die Verwendung einer getrennten Data-Warehouse-Lösung ist zudem die Sicherheit und Integrität der Daten mehr gefährdet, da beispielsweise neue Angriffsvektoren für Eindringlinge entstehen, die durch das gewachsene Sicherheitskonzept auf System z vermieden werden können.

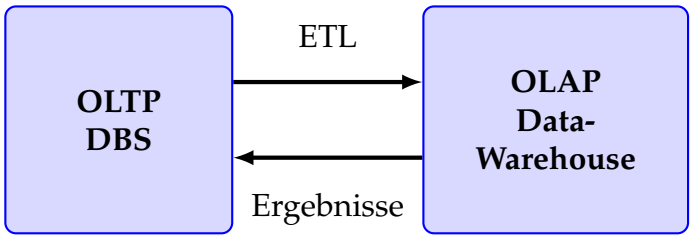

Abbildung 3.2: ETL-Prozess für eine Quelle, wobei zusätzlich eine Rückführung für Ergebnisse in das Datenbanksystem gezeigt wird.

3.2 Machbarkeitsstudie Modellierung auf IDAA

IBM System z ist für seine starke transaktionale Leistung bekannt, für große analytische Workloads muss das System jedoch anderen Anforderungen gerecht werden. Oft werden dafür isolierte Data-Warehouse-Systeme verwendet, die über komplexe ETL-Prozesse mit Daten befüllt werden. Die Verwendung von weiteren Systemen bringt zudem neue Fragen bei der Gewährleistung Datensicherheit und -integrität mit sich. Diese und weitere Kritikpunkte sind in den beiden im letzten Abschnitt 3.1 vorgestellten Szenarien ausführlicher dargestellt. Dieser Abschnitt stellt eine Machbarkeitsstudie des IBM Systems Optimization Competency Centers vor, die darauf abzielt, die oben genannten Kritikpunkte als Herausforderung zu sehen, um SPSS Modeler Workload wie

die Erstellung von Data-Mining-Modellen auf DB2 Analytics Accelerator zu ermöglichen. Für die Beschreibungen in diesem Abschnitt wurde dabei teilweise auf IBM-interne Dokumentationen zurückgegriffen. Die grundsätzliche Vorstellung des Ansatzes erfolgt im Abschnitt 3.2.1 und daran anschließend werden die dafür entwickelten Erweiterungen für die bestehenden Systeme in Abschnitt 3.2.2 präsentiert. Im letzten Abschnitt 3.2.3 werden dann einige Herausforderungen dargestellt, die im Rahmen des neuen Ansatzes auftreten.

3.2.1 Ansatz

Einer der treibenden Gedanken der Machbarkeitsstudie ist, dass der relativ neue IBM DB2 Analytics Accelerator basierend auf seiner Netezza-Technologie sehr gut für OLAP-typische Aufgaben geeignet ist. Bisher fand die Nutzung des Accelerators ausschließlich für einzelne, vom DB2 delegierte Queries statt. Ziel der Studie ist die Evaluierung, ob IDAA für die Erstellung von Data-Mining-Modellen - etwa mit SPSS Modeler - geeignet ist. Gegenstand der Untersuchung ist damit der komplette für die Modellierung nötige Zyklus, beginnend mit der Datenvorbereitung, gefolgt von der Modellerstellung, aber auch ein mögliches Update des Modells bei aktualisierten Daten sowie dem Prozess, dass berechnete Modell zum Scoring im transaktionalen Umfeld zu nutzen. Auf der technischen Seite sind dafür einige prototypische Erweiterungen wie eine angepasste grafische Oberfläche im SPSS Modeler sowie Anpassungen am DB2 und auf dem Accelerator benötigt, diese werden im folgenden Abschnitt 3.2.2 detaillierter ausgeführt. Mithilfe von praxisrelevanten Szenarien soll zudem der mögliche Einsatzzweck gezeigt werden, außerdem werden die gewählten Beispiele in einem Performance-Test vergleichend mit der bisherigen Berechnung und mit dem neuen Ansatz verwendet. Mit der Nutzung des DB2 Analtics Accelerator für komplette analytische Berechnungen kann unter Umständen auf die Einbindung eines externen Data-Warehouse-Systems verzichtet werden, was die Komplexität für analytische Prozesse reduziert. Mögliche Vorteile sind zudem wegfallende ETL-Prozesse und somit auch ein schnellerer Zugriff auf die Daten, desweiteren ist durch die Zentralisierung auf die Plattform System z ein hohes Maß an Sicherheit vorhanden.

3.2.2 Prototypische Erweiterungen

Damit Modelle und analytische Funktionen aus SPSS Modeler heraus direkt auf dem DB2 Analytics Accelerator aufgerufen werden können, müssen Änderungen an mehreren Stellen vorgenommen werden. Diese Änderungen sind von IBM im Zusammenhang mit der Machbarkeitsstudie entwickelt worden. Allgemein muss eine Zugriffsmöglichkeit von SPSS Modeler auf die INZA-

Befehle geschaffen werden, allerdings kann SPSS Modeler nicht direkt mit dem Accelerator kommunzieren. Da das ganze Szenario relativ komplex ist, wird in Abbildung 3.3 eine Übersicht über einen möglichen Aufruf gegeben. Die Änderungen für jeden Teilschritt werden nun beschrieben. Aufbauend auf einem Stream in SPSS Modeler Client wird unter Zurhilfenahme von CLEF-Erweiterungen auf dem SPSS Modeler Server SQL-Code generiert. Anstatt der Berechnung einer analytischen Funktion wie etwa einer Assoziationsanalyse löst die CLEF-Erweiterung den Aufruf der entsprechenden INZA-Funktion aus, die als Stored Procedure verpackt wird. Diese Stored Procedure wird an das DB2-Subsystem weitergeleitet, welches mit dem DB2 Analytics Accelerator verbunden ist. Dort wird der Aufruf samt Parameter von der DB2 API weitergeleitet an den Accelerator, genauer den SMP Host. Auf dem SMP Host innerhalb vom DB2 Analytics Accelerator erfolgt der Aufruf der tatsächlichen Stored Procedure und der folgenden Berechnung auf den einzelnen Snippet Blades (S-Blade). Das resultierende Modell wird dann auf dem DB2 Analytics Accelerator als Netezza-Tabelle gespeichert.

3.2.3 Herausforderungen

Mit der neuesten Version 3.1 des IBM DB2 Analytics Accelerator kann ein automatisiertes Aktualisieren der Daten auf dem Accelerator nach Änderungen auf der DB2-Seite aktiviert werden. Basierend auf den Daten sind Data-Mining-Modelle entstanden, die dementsprechend auch aktualisiert werden müssen, damit die Ergebnisse beispielsweise für Business Analytics Prozesse verwendet werden können. Bei manchen Modellen, wie etwa der Assoziationsanalyse, gibt es Ansätze zum inkrementellen Data-Mining, was bedeutet, dass das Modell nicht komplett neu berechnet, sondern unter Verwendung der neuen Daten aktualisiert wird. Diese Idee wird im Abschnitt 4.3 näher untersucht und damit eine mögliche Optimierung vorgeschlagen.

Eine weitere Herausforderung ist der Umgang mit Zwischenergebnissen, die auf der IDAA-Seite erstellt werden und dann auch auf DB2 gehandhabt werden müssen. Um die temporären Daten nicht auf DB2 kopieren zu müssen, werden auf DB2 stattdessen nur sogenannte Proxy-Tabellen ohne den wirklichen Inhalt angelegt. Sobald auf diese Tabellen zugegriffen wird, erfolgt eine Weiterleitung an die tatsächliche Tabelle auf dem IDAA.

Der DB2 Analytics Accelerator beschleunigt normalerweise SQL-Queries, die vom DB2 Optimizer an den Accelerator weitergeleitet werden. Für analytische Aufgaben wie eine Modellerstellung müssen die entsprechenden mathematischen Funktionen auch auf dem Accelerator implementiert werden. Für die Machbarkeitsstudie kann dabei auf viele bereits implementierte Modelle zurückgegriffen werden, die für eine andere Nutzung mit SPSS Modeler ge-

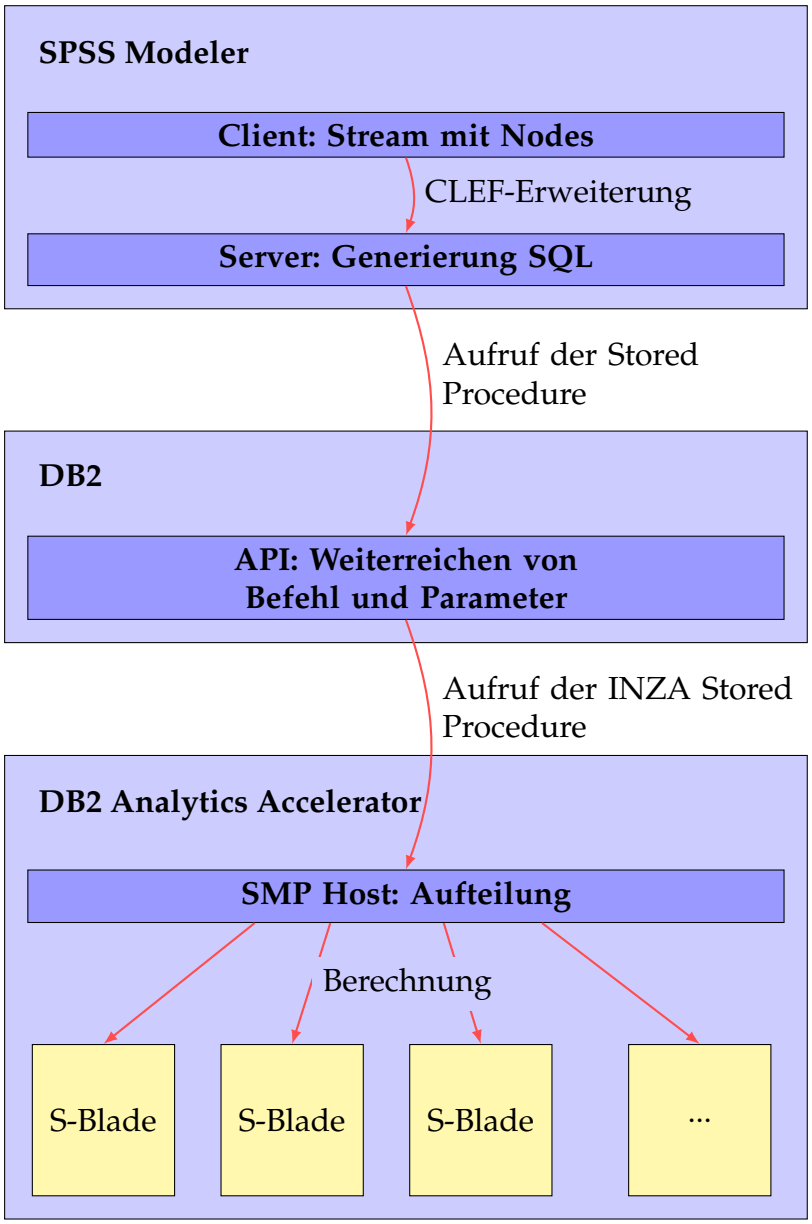

Abbildung 3.3: Zusammenarbeit von SPSS Modeler, dem DB2 Analytics Accelerator und den prototypischen Erweiterungen, um INZA-Befehle auszuführen.

dacht sind [sps11b]. Diese und weitere Funktionen sind im Rahmen des INZA-Paketes vorhanden, wodurch viele Modelle wie Entscheidungs- und Regressionsbäume, Assoziationsanalysen, Cluster-Algorithmen oder Bayessches Netz schon verfügbar sind. Einige Algorithmen fehlen allerdings momentan noch, so gibt es zum Beispiel noch keine Möglichkeit, ein Neuronales Netz aufzubauen.

3.3 Industrie-Blueprints als Anwendungsfall

Um Aussagen über die Machbarkeit des Ansatzes mit IDAA zur Modellberechnung treffen zu können, werden nachfolgend Szenarien beschreiben, wie sie auch in der Wirtschaft zum Einsatz kommen. Die Grundlage für diese Szenarien bilden IBM Industry Models wie das IBM Retail Data Warehouse General Information Manual oder das IBM Telecommunications Data Warehouse General Information Manual, aus welchen praxisnahe Blueprints zur Umsetzung von Geschäftsprozessen entstanden sind [IBM09, IBM11]. Aus diesem Grund wird der IBM SPSS Retail Promotions Blueprint im Abschnitt 3.3.1 beschrieben. Erklärt werden dann die stattfindenden Schritte im Rahmen der Datenvorbereitung und der Modellberechnung, wobei dafür ein Entscheidungsbaum aus SPSS Modeler genutzt wird. Im darauffolgenden Abschnitt 3.3.2 wird der Blueprint Profitability and Customer Retention for Telecommunications verwendet, um eine Assoziationsanalyse inklusive Datenvorbereitung abzubilden. In beiden Fällen werden dabei IBM-interne Dokumentationen verwendet. Außerdem wird der jeweils ursprüngliche Blueprint aufgrund der Komplexität nur in Teilen umgesetzt, um die gewählten Algorithmen isoliert und detailliert zu beschreiben.

3.3.1 Retail Prediction Promotions Blueprint

Eine typische Anwendung im Einzelhandel kann es sein, dass mit Hilfe von Kundendaten Vorhersagen zum künftigen Kaufverhalten der Kunden getätigt werden sollen. Diese Vorhersage wird von einem vorher erstellten mathematischen Modell auf der Grundlage von bisher gesammelten Kundendaten getroffen und kann zur weiteren Nutzung für neue Datensätze vorgehalten werden. Dem Ergebnis entsprechend angepasst wird dem Käufer über das potentiell richtige Medium - etwa Post, E-Mail oder direkt im Shop - die Werbebotschaft mitgeteilt. Je nach Erfolg oder Mißerfolg kann das Modell dann an neue Parameter angepasst und somit verbessert werden.
Der ursprüngliche Blueprint Retail Prediction Promotions beinhaltet drei Szenarien, von welchem im Folgenden allein auf Direct Mail Promotions eingegangen wird.

Online Promotions oder Warenkorbanalyse. Kunden befinden sich in einem Online-Shop eines Unternehmens, dabei sollen entsprechend ihres Verhaltens interessante Angebote angezeigt werden.

Store Promotions Durch Verkaufszahlen nach Aktionen im Geschäft werden vorhergesagte und realisierte Werte verglichen, worüber künftige Aktionen besser geplant werden können.

Direct Mail Promotions Auf Grund der erstellten Profile und dem Verhalten von Kunden wird über mathematische Modelle berechnet, ob verschiedene Werbeträger beim Konsumenten eine Reaktion wie etwa Kauf eines Produktes hervorrufen. Dementsprechend kann die Zustellung von Werbung auf dem potentiell richtigen Medium ausgeliefert werden.

Durch die Begrenzung auf das Szenario Direct Mail Promotions wird die Komplexität des Problems reduziert, jedoch wird dadurch eine Konzentration auf ausgewählte Elemente erreicht, welche dann detailliert untersucht werden. Im Beispiel geht es speziell um das Modell Entscheidungsbaum sowie allgemeine Datenbankoperationen wie Merge, Filter, Distinct und Aggregate. In Abbildung 3.4 ist der Stream zu sehen, der zum einen Daten für die weitere Verwendung vorbereitet und zum anderen das Modell berechnet, mit dem dann weitergearbeitet werden kann. Die ersten beiden Schritte zeigen einen Teil der Datenvorbereitung. Zu Beginn werden dabei die zwei Tabellen *CST_TRANSACTION_SAMPLE* und *TRANSACTION_LOG* als Datengrundlage eingebunden, die neben zwei später verwendeten Tabellen in Abbildung 3.5 gezeigt werden. Die beiden Tabellen werden dann sogleich mit einen Inner Join (Merge) über die Warenkorb-Transaktions-ID *MKT_BSKT_TXN_ID* zusammengeführt werden. Der erste Knoten im zweiten Schritt, *RFM and Sales potential*, kennzeichnet einen Supernode, unter welchem verschiedene im nächsten Absatz erklärte Operationen ausgeführt werden. Anschließende erfolgt eine erneute Zusammenführung der Daten per Inner Join über die *CUST_ID*, diesmal mit historischen Informationen über das Antwortverhalten der verschiedenen Kommunikationskanäle aus der Tabelle *CST_CHANNEL_RESPONSE*. Im übernächsten Absatz wird als dritter Schritt der Supernode *Calculate Channel Propensity* beschrieben, welcher für die Berechnung der Modelle verantwortlich ist. Im vierten Schritt werden die durch die Modellberechnung gewonnenen Daten mit der Tabelle *CST_DEMOGRAPHY* zusammengeführt, in welcher Informationen zur Reaktion von Kunden auf Werbekampagnen abgelegt sind. Abschließend werden die verarbeiteten Informationen in die Tabelle *CST_RFM_CHANNEL_PROPENSITY_DM_DATA* geschrieben. Diese und eine weitere Ausgabetabelle sind in Abbildung 3.6 veranschaulicht. Zusammenfassend kann festgestellt werden, dass der Stream Modelle bereitstellt, mit denen neue Kunden möglichst schnell auf dem für sie passenden Kommunikationsmittel angesprochen werden können.

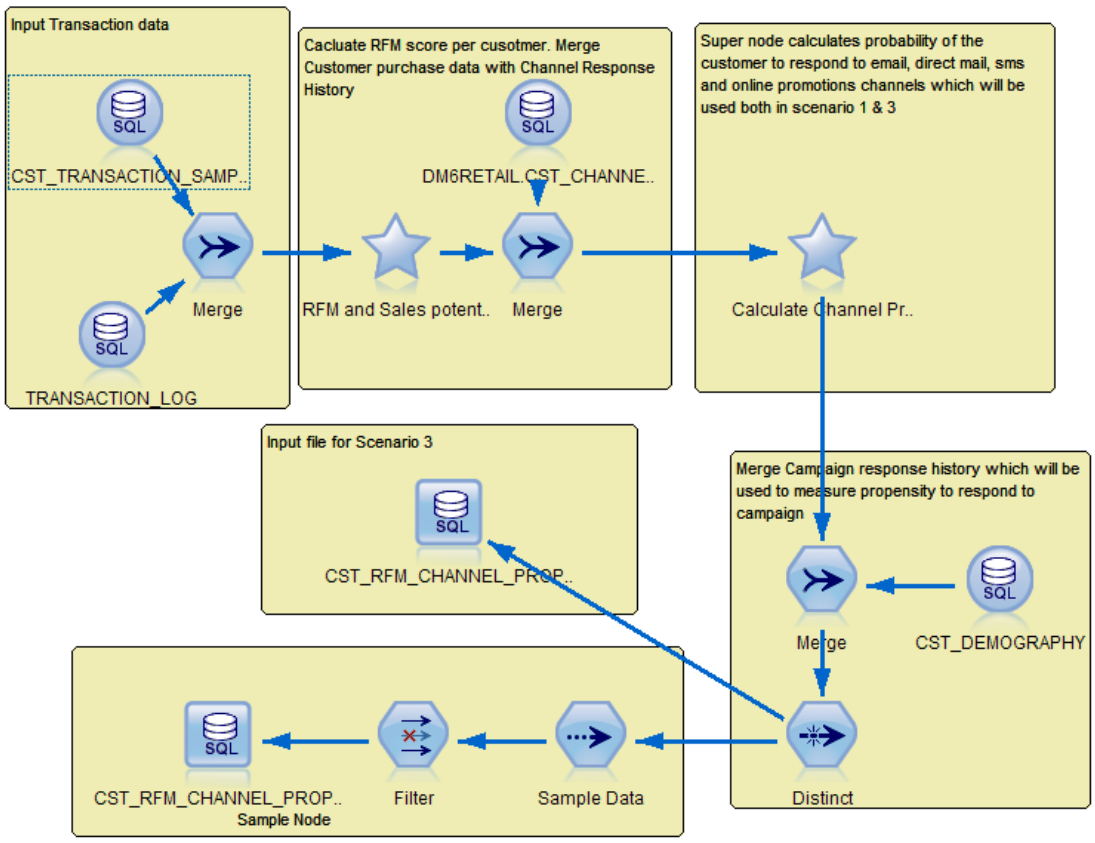

Abbildung 3.4: Dieser Stream aus dem Retail Blueprint zeigt die komplette Vorverarbeitung von Daten und der darauf folgenden Modellberechnung unter dem Supernode *Calculate Channel Propensity*. Diese berechneten Modelle können dann zum Scoring von neuen Daten herangezogen werden. Die letzten Schritte im Stream schreiben neu gewonnene Informationen in Tabellen, die dann etwa als Eingabe für andere Streams oder zur Auswertung der Ergebnisse verwendet werden können.

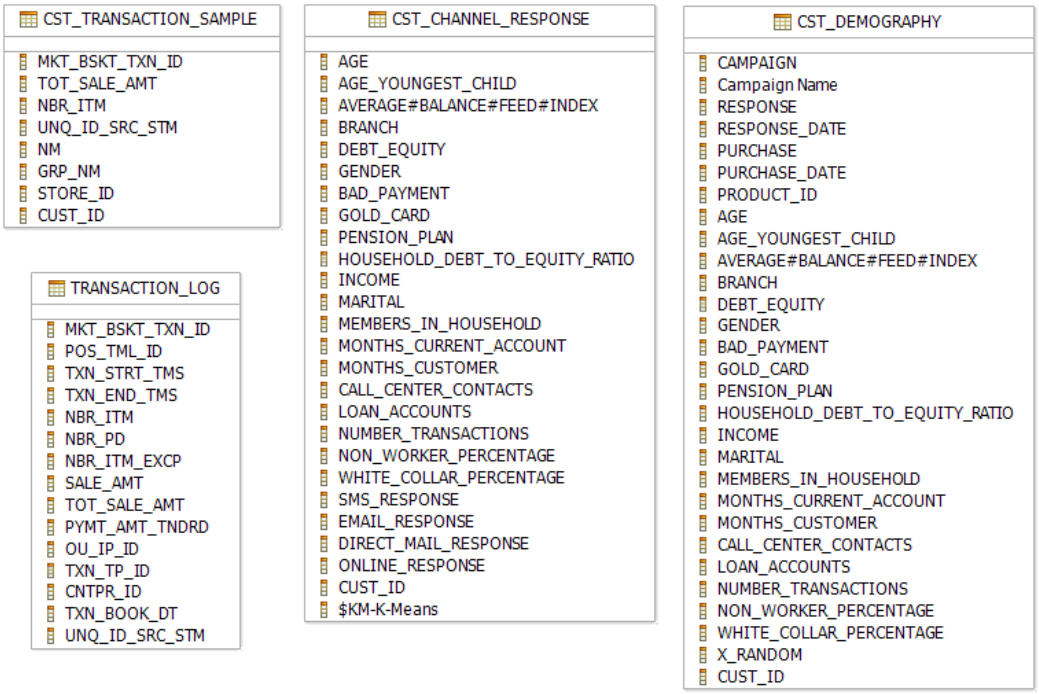

Abbildung 3.5: Input-Tabellen im gewählten Szenario des Retail Prediction Blueprints: Zuerst verwendet werden die beiden Tabellen `CST_TRANSACTION_SAMPLE` und `TRANSACTION_LOG`, gefolgt von der Tabelle `CST_CHANNEL_RESPONSE` nach einigen Operationen. Einige Schritte später folgt als letzte Input-Tabelle `CST_DEMOGRAPHY`.

Abbildung 3.6: Output-Tabellen beim Retail Blueprint: Direkt nach der Modellberechnung werden die Ergebnisse in der Tabelle `CST_CHANNEL_RESPONSE_PROPENSITY` gespeichert. Es folgt eine Verknüpfung mit weiteren Daten, um die Daten abschließend in die Tabelle `CST_RFM_CHANNEL_PROPENSITY_DM_DATA` zu schreiben.

Supernode RFM and Sales potential Der erste Supernode, zu sehen in Abbildung 3.7, berechnet für jeden Kunden den RFM-Score, der sich aus drei Teilen zusammensetzt. *R* steht für *Recency* und gibt an, wann der Kunde das letzte Mal einen Einkauf getätigt hat, *F* steht für *Frequency* und sagt, wie regelmäßig der Kunde zu Besuch ist und *M* schließlich bedeutet *Monetary Value* und verdeutlicht, wie viel Geld der Kunde im Durchschnitt ausgibt. Für jede Kategorie aus *R*, *F* und *M* werden dann entsprechend der vorhandenen Einzelwerte ähnlich wie für Schulnoten diskrete Bereiche gefunden, mit denen eine Einteilung stattfinden kann, zum Beispiel 1 - 5 für niedrige bis hohe Bereiche für den durchschnittlichen Betrag beim Einkauf. Zudem kann den einzelnen Kategorien ein Gewicht zugeteilt werden, um etwa regelmäßig Einkäufe mehr zu belohnen als einmalige teure Anschaffungen. [MH07]

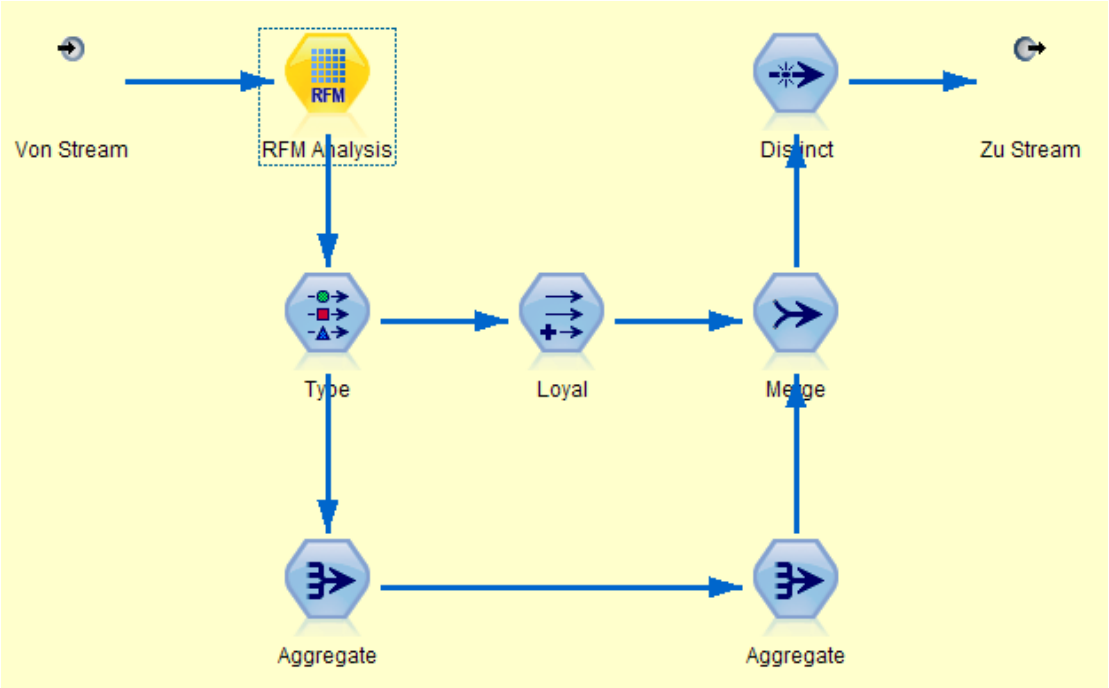

Abbildung 3.7: Supernode RFM and Sales Potential: Zu Beginn findet die eigentliche RFM-Analyse statt, im Nachhinein werden Kunden ab einem bestimmten RFM-Wert als *loyal* bezeichnet und zudem werden die Gesamt-Kaufbeträge in den `Aggregate`-Knoten aufaddiert.

In der Datenvorbereitung werden die Eingabewerte für die RFM-Analyse im Stream fehlerhaft aufbereitet. Eine einfache Lösung wäre an dieser Stelle, einen RFM-Aggregatknoten vorzuschalten, der die Daten für die RFM-Analyse passend aufbereitet. Im folgenden wird erläutert, warum Änderungen stattfinden müssen und zudem wie die Anpassungen für diesen konkreten Fall aussehen.

Die Eingabedaten bestehen aus Transaktionen, die über eine ID (MKT_BSKT_-TXN_ID) gekennzeichnet werden. Jede Transaktion beinhaltet mehrere Tupel (Produkte, die ein Kunde in dieser Transaktion kauft), die einer Transaktion zugeordnet sind. Für diese Tupel sind jeweils die Eingabewerte `TXN_BOOK_-DT` (für Recency), `CUST_ID` (für Frequency) und `TOT_SALE_AMT` (für Monetary Value) verzeichnet. Jedes dieser Tupel wird fälschlicherweise in der Eingabe verwendet, wodurch sich die Ergebnisse in die Richtung der RFM-Scores von Transaktionen mit vielen gekauften Produkten verschiebt. Für eine vergleichende Analyse ist dies nicht relevant, da bei einem Vergleich mit den gleichen Eingabedaten in beiden Szenarien auch die gleiche Verschiebung stattfindet. [sps12e]
Für eine reale Berechnung auf Kundendaten sollten die Eingabedaten vorher über einen Distinct auf der Transaktions-ID angepasst werden, womit die für eine RFM-Analyse uninteressanten einzelnen Produkte aussortiert werden. Dem folgend wird ein RFM-Aggregatknoten vor die RFM-Analyse geschaltet, welcher die Daten korrekt vorbereitet. In Abbildung 3.8 wird dieser Knoten verdeutlicht, auf die drei Eingabefelder „ID“, „Date“ und „Value“ wird im folgenden eingegangen.
Die `CUST_ID` wird dabei als eindeutige „ID“ für die Berechnung der Grenzen von Recency, Frequency und Monetary Value verwendet, da für jeden Kunden nur je ein Wert berechnet werden soll. Als „Date“ wird dann wie in der Abbildung des RFM-Aggregatknotens zu sehen `TXN_BOOK_DT` verwendet, um den Abstand vom *letzten* Einkauf des Kundens bis zum heutigen Tag zu berechnen. Bisher wurde für jede Transaktion das Datum aus `TXN_BOOK_DT` erneut in die Berechnung der Recency aufgenommen, was falsch ist, da an dieser Stelle nur der zuletzt durchgeführte Kauf wichtig ist. Über den RFM-Aggregatknoten wird somit der korrekte Recency-Wert eines jeden Kunden in der passenden Spalte Recency übertragen. Bisher wird die `CUST_ID` direkt als Eingabe für Frequency verwendet, womit der Wert der Kundennummern gleichgesetzt wird mit der Anzahl der Einkäufe, was erneut nicht korrekt ist. Dementsprechend verteilten sich zudem die Bereiche der Frequency auf die vorhandenen Kundennummern, so etwa auf die Werte $0 - 999$. Der Widerspruch wird an dieser Stelle spätestens deutlich, da jede verzeichnete Transaktion mindestens einem Einkauf entspricht, die Anzahl der Einkäufe also nicht 0 sein kann. Im RFM-Aggregatknoten wird demzufolge über das Vorkommen der `CUST_ID` die korrekte Anzahl der Einkäufe bestimmt und als neue Spalte `Frequency` bereitgestellt. Der dritte Wert „Value“ im RFM-Aggregatknoten bereitet den Wert Monetary Value vor, indem er den Kaufbetrag einer einzelnen Transaktion `TOT_-SALE_AMT` zugewiesen bekommt. Die einzelnen Kaufbeträge führen dann in Verbindung mit der `CUST_ID` zur Berechnung des durchschnittlichen Kaufbetrag eines jeden Kunden, was in der neuen Spalte `Monetary` gespeichert wird.

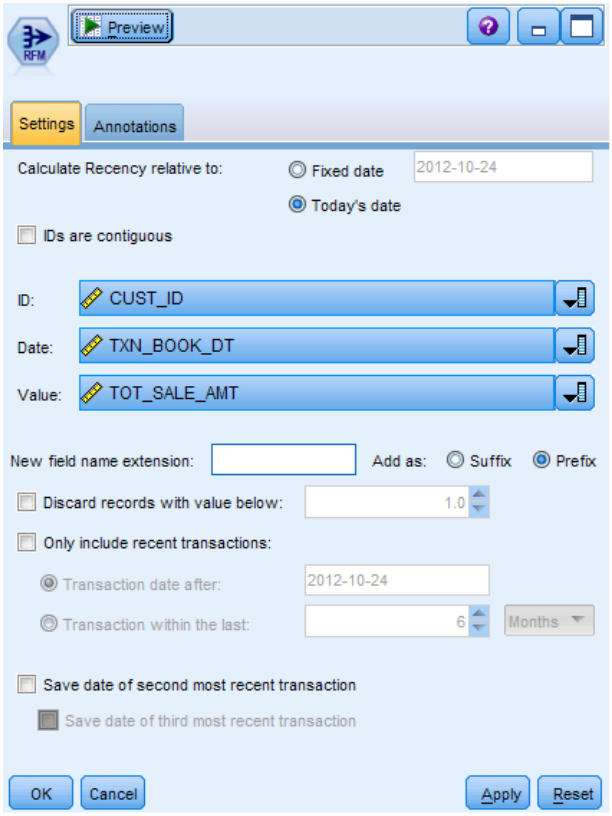

Abbildung 3.8: Parameter für den RFM-Aggregatknoten

Die neuen Spalten `Recency`, `Frequency` und `Monetary` werden dann als Eingabewerte für die eigentliche RFM-Analyse verwendet, wie in Abbildung 3.9 verwendet, womit eine Einordnung aller Kunden in die jeweils passenden Kategorien stattfindet. Danach wird mit W als Wert der Kategorie und G als Gewicht der `RFM Score` für jeden Kunden wie folgt berechnet.

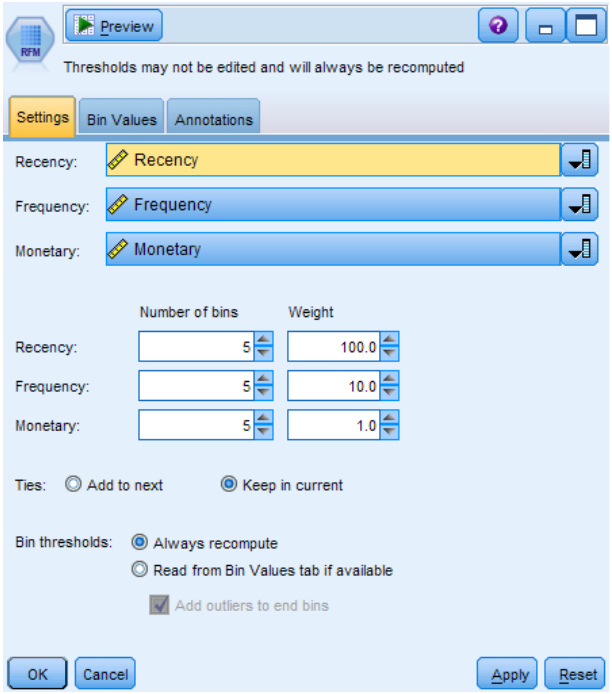

Abbildung 3.9: Parameter für den Knoten RFM-Analyse

$$
\begin{aligned}
RFM &= W_{\text{Recency}} \cdot G_{\text{Recency}} + W_{\text{Frequency}} \cdot G_{\text{Frequency}} + W_{\text{Monetary}} \cdot G_{\text{Monetary}} && (3.1)\\
&= 3 \cdot 100 + 5 \cdot 10 + 2 \cdot 1 && (3.2)\\
&= 352 && (3.3)
\end{aligned}
$$

Mit diesem RFM-Wert können die Klienten verschiedene Klassen eingeteilt werden, etwa um Kunden mit hohem RFM-Wert mehr Rabatt anzubieten als denen mit geringeren Werten. Im Stream folgt ein Typoperator, der zwei Datentypen leicht ändert: *NBR_ITM* und *OU_IP_ID* wechseln von Real zu Integer. Es folgt eine Zweiteilung des Streams. Zum einen findet sich ein Ableitungsknoten *Loyal*, hier wird eine neue Tabellenspalte angelegt in der ein Kunde als treu eingestuft wird, wenn die RFM-Analyse einen bestimmten festgesetzten Wert überschreitet. Zum anderen werden über zwei Aggregierungsknoten die Felder

TOT_SALE_AMT und *TOT_SALE_AMT_Mean* mit den entsprechend aufsummierten Einzelwerten angelegt. Dabei kann angemerkt werden, dass die Aggregierung etwa für `TOT_SALE_AMT` ohne Nutzen ist, da der Durchschnitt von x gleichen Werten wieder x entspricht. Der zudem in diesem Schritt eingefügte `Record_Count` gibt die Anzahl der Produkte pro Transaktion an und wird im nächsten Aggregate-Knoten durch die Gesamtzahl der gekauften Produkte pro Kunde überschrieben. Letztlich erfolgt eine Zusammenführung der Teil-Streams und ein Distinct über die *CUST_ID*, wonach die Daten zum Haupt-Stream zurückgegeben werden.

Supernode Calculate Channel Propensity In diesem Supernode, welcher in Abbildung 3.10 zu sehen ist, findet die tatsächliche Modellierung nach dem vorgegebenen Modell CHAID statt. Zu Beginn wird in je einem Typknoten pro Modell das Ziel der Berechnung angegeben, zum Beispiel das Feld *SMS_RESPONSE* für das gleichnamige Modell. Infolgedessen stoßen die vier blauen CHAID Knoten mit den bisher bekannten Daten - den Trainingsdaten - die Generierung des jeweiligen Modells an. Im Stream wird das berechnete Modell je durch ein CHAID Modell-Nugget angezeigt. Das erstellte Modell-Nugget kann dann in jedem beliebigen Stream zum Scoring verwendet werden. Nach der Modellberechnung werden für drei der vier Teilstreams die doppelten Attribute herausgefiltert, die nur für die Berechnung benötigt wurden. Im letzten Abschnitt vom Supernode wird ein Merge - wieder als Inner Join - in eine gemeinsame Tabelle durchgeführt und danach zum einen in die Tabelle *CST_CHANNEL_RESPONSE* geschrieben und zum anderen an den Haupt-Stream zurückgegeben. [sps12b]

3.3.2 Profitability and Customer Retention for Telecommunications Blueprint

Mit dem zweiten Blueprint wird ein Szenario aus der Telekommunikationsbranche beschrieben, welches sich mit dem Thema Gewinnmaximierung und Kundenbindung beschäftigt. Ein Mittel dafür kann es sein, mit Hilfe von Call Centern Kunden anzurufen, welche in bestimmte Verhaltensmuster fallen. Weiterhin kann es nützlich sein, Kunden mit großem Umsatz über spezielle Aktionen einen hohen Anreiz zu geben, dem Unternehmen treu zu bleiben. Auf diese Art und Weise können weitere Prozesse gefunden werden, die unter Zuhilfenahme von gespeicherten Kundendaten eine wichtige Grundlage zur Optimierung ausmachen. Im Umfang des Blueprints finden sich verschiedene Analyseprozesse, welche Themen wie Kundenabwanderung im Prepaid- und Postpaid-Bereich, den Kundenwert, die Zufriedenheit und Zusammenhänge bei den verwendeten Tarif-Optionen umfassen. Über die Verwendung

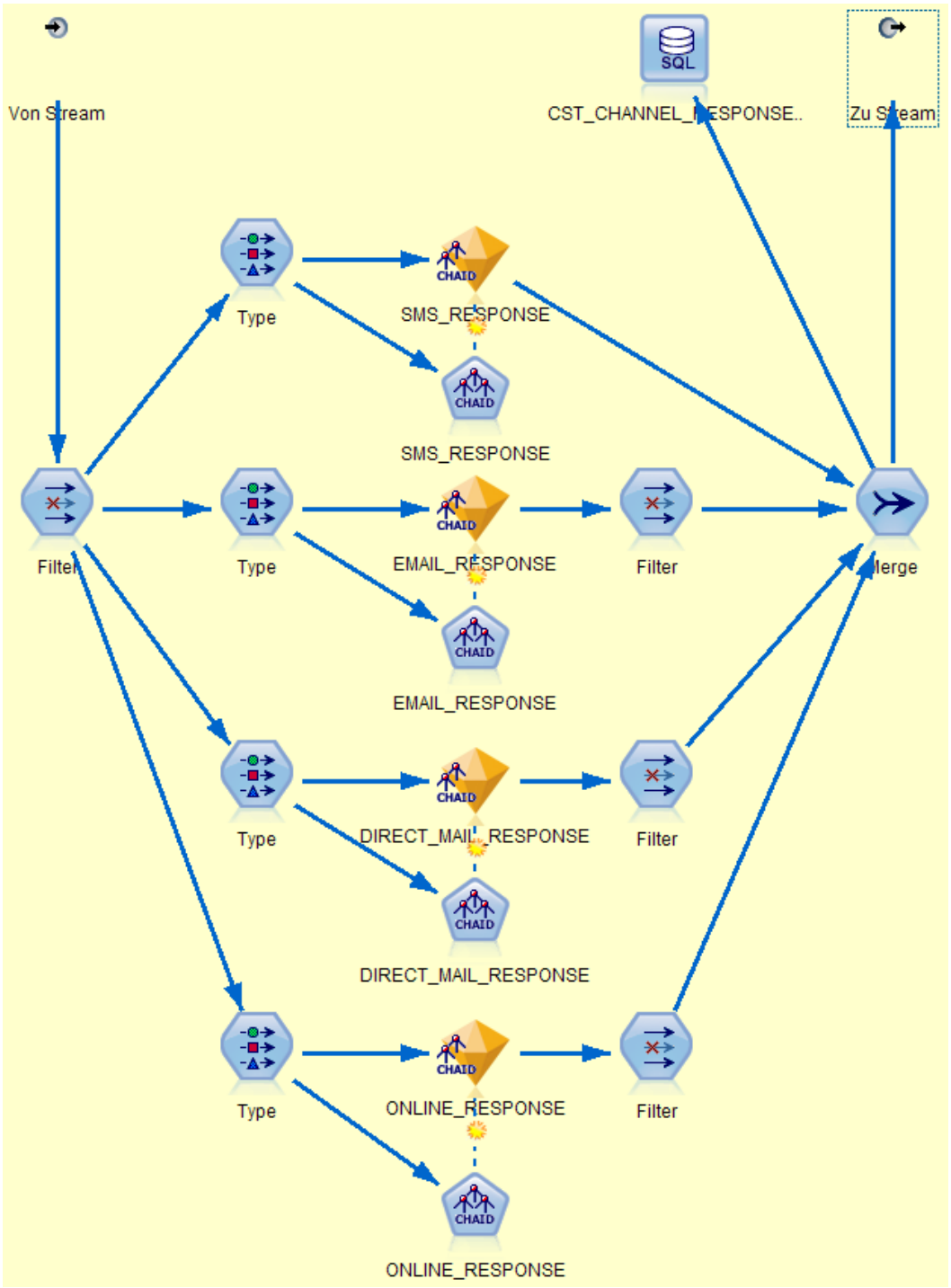

Abbildung 3.10: Supernode Calculate Channel Propensity: Modellberechnung in den CHAID-Knoten pro Kommunikationskanal. Berechnetes „goldenes“ Nugget erstellt dann mit den Eingabedaten die Werte pro Kunde.

von Entscheidungsbaum-, Regressionsmodellen oder der Assoziationsanalyse werden mit Hilfe der gespeicherten Informationen über einzelne Kunden neue Erkenntnisse gewonnen werden, die dann etwa für gezieltes Marketing eingesetzt werden. Im folgenden werden die Zusammenhänge für Tarif-Optionen im Prepaid-Bereich mit einer Assoziationsanalyse beschrieben und weiter untersucht.
Wenn Brot und Butter eingekauft wird, besteht eine 90% Chance, dass auch Milch gekauft wird. Diese Aussage stammt aus einer Arbeit von Agrawal et al. [AIS93] aus dem Jahr 1994, in welcher erstmals das Finden von Assoziationsregeln in großen Datenmengen beschrieben wird. Analog dazu könnte eine Aussage im Telekommunikationsbereich heißen: Wer eine Daten- und SMS-Option gebucht hat, nutzt zu 90% auch eine Auslands-Option. Die 90% geben dabei an, in wie vielen Fällen die Wahl von Daten- und SMS-Option auch zur Entscheidung für die Auslands-Option führt. Die Regelsätze werden über die Analyse der vergangenen Transaktionen gefunden, im hier untersuchten Beispiel werden dafür die Buchungen der gewählten Tarifoptionen aller Kunden auf mögliche Assoziationen untersucht. [AIS93]

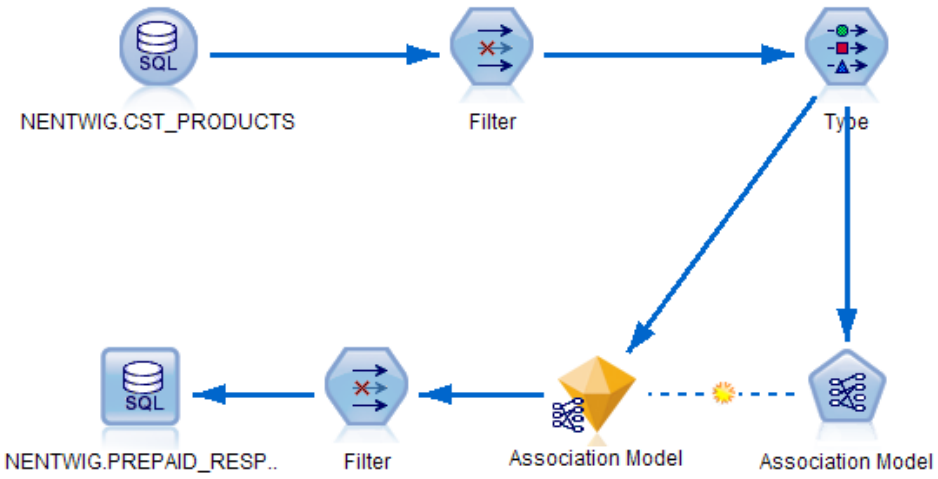

Abbildung 3.11: Stream zur Umsetzung einer Assoziationsanalyse auf Grundlage von kundenbezogenen Tarif-Optionen im Telekommunikationsbereich: Die obere Zeile zeigt die Datenvorbereitung, um in der unteren Zeile das Modell zu berechnen und dieses abschließend auf Kundendaten zu nutzen.

Zu diesem Zweck wird der in Abbildung 3.11 gezeigte SPSS Modeler Stream verwendet. Der Schritt der Datenvorbereitung fällt dabei mit drei Nodes kurz aus, da die benötigten Tarif-Optionen der Kunden wie `SMS PACKAGE` oder `ADVANCED DATA PACKAGE` sowie die Kundennummer in der Tabelle `CST_-`

PRODUCTS vorliegen. Diese Tabelle sowie die einzige Ausgabetabelle sind in Abbildung 3.12 zu sehen. Der zweite Node benennt einige Attribute um und übergibt die Daten an den „Type“-Knoten, der die Rollen für die Modellberechnung festlegt. Außerdem legt dieser Knoten für jede Tarif-Option die den Wertebereich als „Flag“ fest, wodurch nur die Boolschen Werte „Y“ oder „N“ verwendet werden. Dem folgt die Erstellung des Modells im blauen Node „Association Model“ und gleich danach die Anwendung des errechneten Modell-Nugget auf die Testdaten. Die automatisch generierten Spaltennamen werden im nachkommenden Node umbenannt, weil sie sonst zu lang für DB2 sind. Abschließend werden diese Ergebnisse in eine neue Tabelle PREPAID_RESPONSE geschrieben, womit die Aufgabe des Streams erfüllt ist.

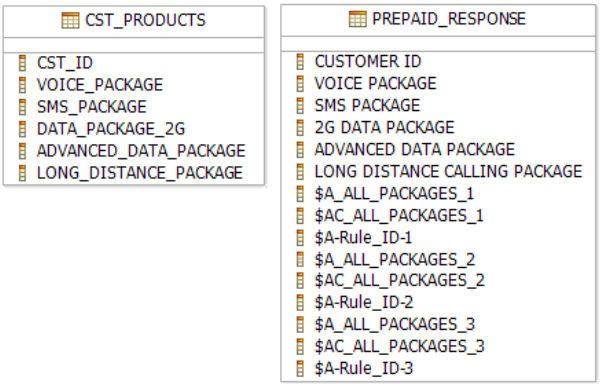

Abbildung 3.12: Ein- und Ausgabetabelle im Telekommunikations-Szenario: In CST_PRODUCTS sind die interessanten Tarif-Optionen abgelegt. Die Ergebnistabelle PREPAID_RESPONSE zeigt zusätzlich zu den gewählten Optionen die mit der Modellberechnung gewonnenen Attribute beginnend mit „$“.

4 Umsetzung

In den Grundlagen im Kapitel 2 wurden Technologien wie SPSS Modeler und der IBM DB2 Analytics Accelerator eingeführt, aufbauend darauf erfolgte im Kapitel 3 eine Einschätzung des Ansatzes, die Modellierung auf SPSS Modeler durch IDAA zu beschleunigen sowie eine Vorstellung von praxisnahen Szenarien aus industrierelevanten Dienstleistungsprozessen. Dieses Vorwissen wird in diesem Kapitel verknüpft, um die gewählten Szenarien aus Einzelhandel und Telekommunikation zum einen auf z/OS DB2 abzubilden und zum anderen auf den DB2 Analytics Accelerator. Bei der Abbildung auf DB2 für z/OS, welche in Abschnitt 4.1 beschrieben wird, wird zudem auf einige aufgetretene Probleme und deren Lösung eingegangen. Wenn die Daten auf z/OS DB2 verfügbar sind, können die Streams unter SPSS Modeler ausgeführt werden. Die weitere Integration der Daten mit dem DB2 Analytics Accelerator wird im Abschnitt 4.2 dargestellt, das Augenmerk richtet sich dann aber vor allem auf die Herausforderung, Algorithmen und Modelle wie die Assoziationsanalyse über Netezza-Algorithmen direkt auf dem Accelerator abzubilden. Damit wird eine Basis für einen Performance-Test sowie weitere Arbeiten geschaffen. So gibt es etwa einige Möglichkeiten zur Optimierung mit dem neuen Ansatz auf IDAA, ein Beispiel für eine schnellere Modellerstellung wird am Ende des Kapitels im Abschnitt 4.3 beschrieben.

4.1 Abbildung auf DB2 für z/OS

Die verwendeten Blueprints setzen als Datenbankanbindung im SPSS Modeler auf DB2 für Linux, UNIX und Windows, was sich an vielen Stellen von DB2 auf z/OS unterscheidet. Aus diesem Grund wird in diesem Abschnitt beschrieben, wie die in Abschnitt 3.3 erläuterten Szenarien auf einem z/OS-System mit DB2 lauffähig gemacht werden. Dabei wird der Abschnitt entsprechend der unterschiedlichen Aufgaben in drei Bereiche aufgeteilt. Da aufgrund der Komplexität der Blueprints nur Ausschnitte aus diesen verwendet werden, erfolgt im ersten Abschnitt 4.1.1 eine Beschreibung, wie die benötigten Daten aus den Blueprints extrahiert werden und welche Probleme dabei aufgetreten sind. Die nächste Aufgabe beinhaltet die Vergrößerung der extrahierten Daten, beschrieben im Abschnitt 4.1.2. Dies wird als Vorbereitung auf den vergleichenden Performance-Test durchgeführt. Abschließend werden die vorbereiteten

Datensätze mit JCL-Skripten auf das DB2 transferiert, was in 4.1.3 zusammengefasst wird.

4.1.1 Extraktion der verwendeten Daten

Im verwendeten Szenario Direct Mail Promotions aus dem Retail Prediction Promotions Blueprint werden vier Tabellen zur Gewinnung der Eingabedaten für die Modellberechnung sowie zwei weitere Tabellen für die Ergebnisse benötigt. Alle Tabellen ergeben sich direkt aus dem SPSS Modeler Stream, der zu dem Szenario gehört und im Abschnitt 3.3.1 beschrieben wird. Für die weitere Verwendung stehen die Tabellen als Dateien mit kommaseparierten Werten in den Anlagen des Blueprints zur Verfügung. Eine wichtige Änderung ist in der Tabelle `TRANSACTION_LOG` nötig, das Feld `TXN_BOOK_DT` hat anfangs das Format „YYYYMMDD“, DB2 benötigt Datumsangaben allerdings im Format „YYYY-MM-DD“. Mit dem Unix-Tool sed kann die Anpassung in einem Arbeitsschritt beispielsweise über das folgende Kommando geschehen (eine Zeile):

```
sed -e
's:\(200[2-4]\)\([0-1][0-9]\)\([0-3][0-9]\):\1-\2-\3:g'
TRANSACTION_LOG > output
```

Das Szenario aus dem Bereich Telekommunikation setzt beim Import der Daten in den SPSS Modeler mit einer Enterprise View auf eine Abstraktionsschicht, daher müssen die Quellen zur Umsetzung manuell im verwendeten SPSS Modeler Stream ermittelt werden. Dabei fällt auf, das viele Tabellenspalten im ursprünglichen Anwendungsfall wie etwa `MARITAL STATUS` oder auch `NUMBER OF CHILDREN` nicht verwendet werden, daher werden diese für die weiteren Messungen aussortiert, womit eine Konzentration auf die Assoziationsanalyse erfolgt. Über zwei Eingabetabellen kann der Zugriff auf die relevanten Information erfolgen, etwa welche Tarif-Optionen ein Kunde gewählt hat und zudem, ob er Prepaid- oder Postpaid-Kunde ist. Ergänzt durch eine Ausgabetabelle sind in diesem Szenario drei Tabellen erfordert. Anders als bei dem Retail Blueprint sind in diesem Anwendungsfall keine weiteren Anpassungen der Daten nötig.

4.1.2 Vergrößerung der Datengrundlage

Die durch den Blueprint bereitgestellte Datenbasis ist oft zu klein, um damit eine praxisnahe Überprüfung der Performance durchzuführen. Die verwendeten Daten im Retail Blueprint sind beispielsweise etwa 15 Megabyte

groß, die im Telekommunikations-Blueprint sogar nur knapp über 600 Kilobyte. Um zudem die durch den DB2 Analytics Accelerator gegebene parallele Architektur gut nutzen zu können, muss der Umfang der Datengrundlage stark vergrößert werden, um eine sinnvolle Performance-Analyse durchführen zu können. Ein dafür geschriebenes kleines Java-Tool besteht aus zwei Klassen, die in Abbildung 4.1 für das Retail-Szenario zu sehen sind. In der Klasse `MorePredictionData` werden Startparameter wie `multiplier` und `maxRecordCount` eingelesen, daraufhin werden die Quelldateien mit den CSV-Datensätzen eingelesen und über Methoden aus der Klasse `DataWriter` vervielfältigt. Dabei werden die Primärschlüssel der Tabellen so angepasst, dass deren referentielle Integrität gewahrt bleibt. Bei dem Szenario aus dem

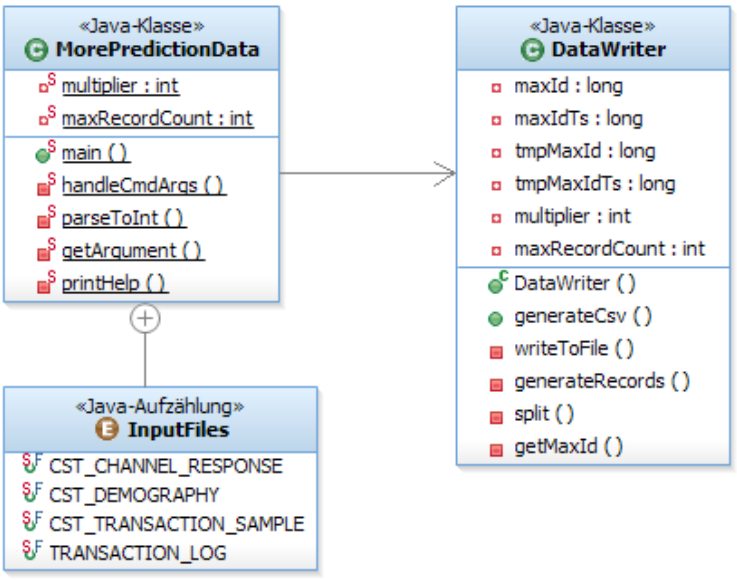

Abbildung 4.1: Klassendiagramm für Programm zur Vervielfältigung, `MorePredictionData` stellt einen allgemeinen Rahmen dar und ruft pro Quelldatei die benötigten Methoden aus der instanziierten Klasse `DataWriter` auf, die zur Vervielfältigung benötigt werden. (Verwendete Software: IBM Rational Software Architect)

Retail Blueprint wird dabei in den Tabellen `CST_DEMOGRAPHY` und `CST_CHANNEL_RESPONSE` die `CUST_ID` angepasst, zudem in `TRANSACTION_LOG` die `MKT_BSKT_TXN_ID` und in der Tabelle `CST_TRANSACTION_SAMPLE` werden beide IDs angeglichen. Wenn die Anzahl der Zeilen in der Zieldatei zu groß wird, werden automatisch durchnummerierte neue Dateien erstellt. Über Startparameter kann angegeben werden, um welchen Faktor die Datenbasis vergrößert werden soll und wie viele Zeilen pro angelegter Datei nicht überschritten

werden dürfen. Nach erfolgreicher Vervielfältigung der Ausgangsdaten können die Daten auf das DB2 importiert werden.

4.1.3 Import auf DB2 für z/OS

Um die Daten auf das DB2 für z/OS importieren zu können, müssen verschiedene vorbereitende Schritte auf dem Mainframe durchgeführt werden, die nachfolgend aufgezählt und dann detailliert erklärt werden.

- Anlegen von benötigten Datasets
- Upload der Datenbasis per FTP auf den Mainframe
- Datenbankobjekte anlegen
- Laden der CSV-Dateien in die Datenbank

Die meisten Teilschritte benötigen Zugriff auf das ISPF-Subsystem (Interactive System Productivity Facility) des verwendeten Mainframes, in dieser Arbeit wird dafür der 3270-Client von IBM (Personal Communications) verwendet. Ein Screenshot vom ISPF ist in Abbildung 4.2 zu sehen, dieser zeigt die Standardansicht nach der Anmeldung. Dabei sind alle verfügbaren Subsysteme zeilenweise aufgelistet und können über die weißen Anfangsbuchstaben gestartet werden. Für die weiteren Arbeiten werden drei Systeme benötigt, die nachkommend kurz beschschrieben werden.

PDF *ISPF/Program Development Facility*, etwa zum Anlegen und Bearbeiten von Datasets, zudem aber auch zum Submit von JCL-Skripts.

SDSF *System Display and Search Facility* zur Prüfung, ob Jobs beendet worden sind oder nicht und zudem, um in den angelegten Logs nach Fehlern beim Ausführen von Skripten zu suchen.

DB2 Zugriff auf das *DB2-Subsystem* für administrative Aufgaben wie Änderung von Rechten, dem direkten Ausführen von SQL Statements oder zur Prüfung, ob Tabellen und deren Inhalten angelegt worden sind.

Der erste Teilschritt betrifft das Anlegen von Datasets auf z/OS, was über das PDF-Subsystem durchgeführt werden kann. In diesem Fall wurden die Datasets `NENTWIG.DATA` und `NENTWIG.JCL.LOAD` angelegt, zum einen für die vorbereiteten CSV-Dateien und zum anderen für die benötigten JCL-Skripte zum Erstellen und Laden der Tabellen. Im nächsten Abschnitt können dann die Daten vom Quellrechner per FTP-Verbindung auf den Mainframe gespielt werden, als Übertragsungsart muss dabei ASCII verwendet werden, was allerdings oft die Voreinstellung ist. In jedem Fall werden die CSV-Dateien per FTP übertragen, aber auch die JCL-Skripte können so auf einem beliebigen Rechner geschrieben werden. Dies bringt den Vorteil mit sich, das der in ISPF integrierte Texteditor mitsamt seiner gewöhnungsbedürftigen Bedienung nur zum Absenden der Skripte verwendet werden muss. Der dritte Schritt beschäftigt sich mit

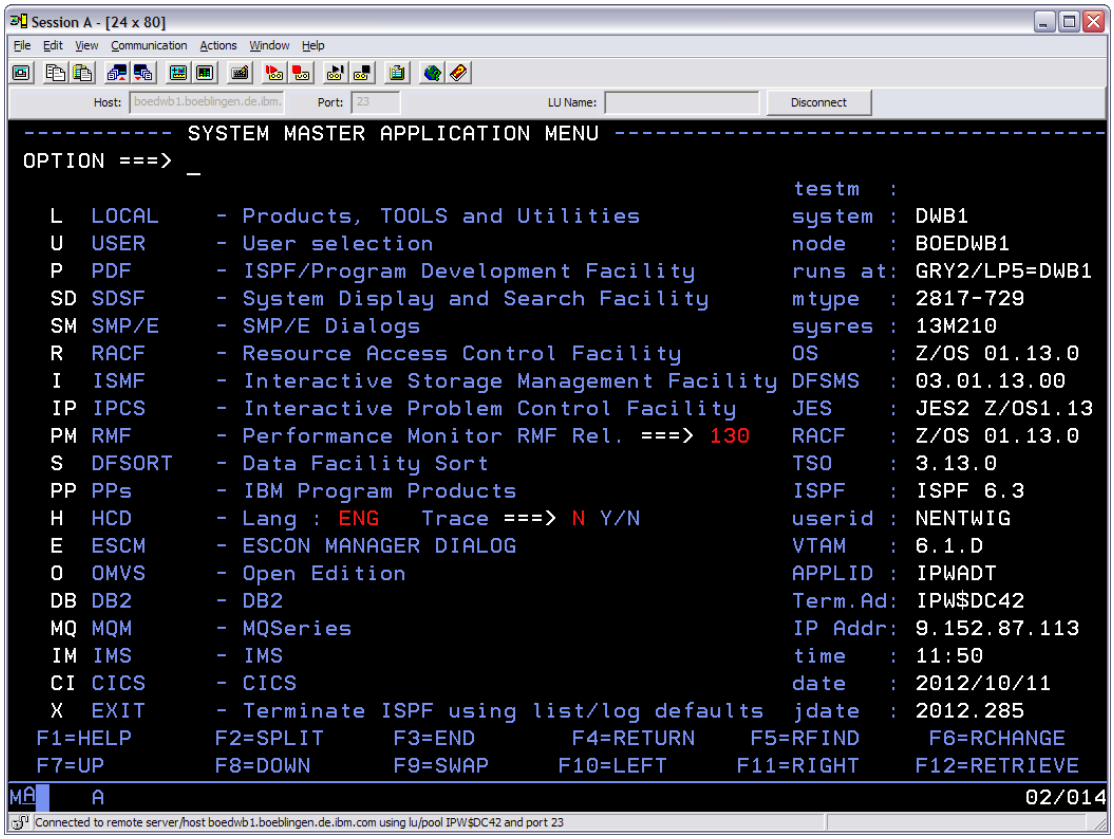

Abbildung 4.2: z/OS ISPF-Subsystem Ansicht beim Start, weiterführende Subsysteme wie PDF, DB2 oder CICS können über den Aufruf der weißen Kürzel zu Beginn einer Zeile gestartet werden.

dem Anlegen der benötigten Datenbankobjekte auf DB2 für z/OS, die wichtigste Voraussetzung dafür sind die benötigten Rechte, um auf eine Storage Group zugreifen zu können oder um diese zu erstellen. Weiterhin braucht der Benutzer Rechte für das Anlegen von Datenbanken und Tablespaces auf dem entsprechenden System. Im DB2-Subsystem selbst können Rechte über `Execute SQL Statements` → `Grant/revoke privileges on objects` geändert werden. Sind die Rechte geklärt, können kleinere SQL Statements direkt über `Run or Explain SQL statements` ausgeführt werden, wie hier beispielhaft für Storage Group, Database und einen Tablespace für den Retail Blueprint:

```
CREATE STOGROUP SGRETAIL
    VOLUMES("*")
    VCAT DBNI;
COMMIT;

CREATE DATABASE RETAIL
    BUFFERPOOL BP2
    INDEXBP BP1
    STOGROUP SGRETAIL
    CCSID UNICODE;
COMMIT;

CREATE TABLESPACE TSTRANLO
    IN RETAIL
    USING STOGROUP SGRETAIL
    CCSID UNICODE
    COMPRESS YES;
COMMIT;
```

Damit sind alle Voraussetzungen zum Anlegen der benötigten Tabellen erfüllt und der nächste Schritt kann begonnen werden. An dieser Stelle werden JCL-Skripte (Job Control Language) verwendet, um Aufgaben über das JES (Job Entry Subsystem) an z/OS zu delegieren. Dort wird die geforderte Aufgabe ausgeführt, wonach das JES erneut genutzt wird, um Ausgabeinformationen für den Nutzer aufzubereiten. Jede Aufgabe wird über ein `JOB`-Statement benannt und kann verschiedene `EXEC`-Statements besitzen, die Programme ausführen. Über `DD`-Statements werden zudem Ein- und Ausgabedaten spezifiziert. Normalerweise beginnen alle JCL-Statements mit `//`, eine detaillierte Übersicht zur Verwendung von JCL ist im entsprechenden User's Guide [zos01] zu finden. Um eine der benötigten Tabellen zu erstellen, wird folgendes JCL-Skript verwendet:

```
//NENTWIGC JOB 'Add io table',MSGCLASS=H,
```

```
//          CLASS=S,NOTIFY=NENTWIG,TIME=1440
//JOBLIB    DD DSN=SYS1.DSN.V910.SDSNLOAD,DISP=SHR
//          DD DSN=SYS1.DSN.SNI1.SDSNEXIT,DISP=SHR
//STEP01    EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN  DD *
DSN SYSTEM(DBNI)
RUN PROGRAM(DSNTEP2) PLAN(DSNTEP91)
END
//SYSIN DD *
CREATE TABLE NENTWIG.TRANSACTION_LOG
(
    MKT_BSKT_TXN_ID INT,
    POS_TML_ID FLOAT,
    TXN_STRT_TMS VARCHAR(40),
    TXN_BOOK_DT DATE
) IN RETAIL.TSTRANLO CCSID UNICODE;
```

Interessant ist hier zum Beispiel der Schritt `STEP01`, in welchem das Mainframe-Utility `IKJEFT01` ausgeführt wird, welches die Ausführung von TSO-Befehlen (Time Sharing Option) im Batch Mode gestattet. Bei der Bedienung von z/OS wird normalerweise ISPF verwendet, welches über Dialoge und Menüeinträge komfortabler zu bedienen ist als die TSO-Umgebung, bei der Befehle über eine Kommandozeile abgegeben werden. Im obigen Falle wird durch den Start von `IKJEFT01` die Ausführung des Programms `DSNTEP2` ermöglicht. `DSNTEP2` wiederum ist ein DB2-Utility, mit welchem dynamische SQL Statements ausgeführt werden können. Das Skript wird im Texteditor im ISPF über `sub` (Submit) abgeschickt und verarbeitet. In diesem Falle wird die Tabelle `NENTWIG.TRANSACTION_LOG` mit den gegebenen Parametern angelegt und die Vorbereitungen sind abgeschlossen. [db210]
Der abschließende Schritt importiert dann die auf bereits auf z/OS gehaltenen Daten in die erstellten Tabellen, wofür erneut ein JCL-Skript verwendet wird, welches nach diesem Abschnitt abgebildet ist. Um die Übersichtlichkeit zu erhöhen, erfolgt eine Beschränkung auf die wichtigsten Zeilen des Skriptes, so etwa wird auf den Beginnn verzichtet, der im letzten Skript sehr ähnlich ist. Das `EXEC`-Statement in diesem Skript führt das `DSNUTILB`-Utility aus, mit welchem es ermöglicht wird, DB2-Utilities zu starten - in diesem Falle `LOAD` zum Befüllen von Tabellen. Die mit `DATA` beginnende Zeile gibt an, welches Dataset als Eingabe verwendet werden soll. Bei der Ausführung des `LOAD`-Utilities werden einige Parameter übergeben, so etwa `REPLACE`, welches den Inhalt im

Tablespace vor dem Laden der neuen Daten komplett löscht oder ENFORCE NO, womit beim Laden der Daten keine Prüfung auf referentielle Integrität erfolgt. [db210]

```
[...]
//LOAD     EXEC PGM=DSNUTILB,PARM=DBNI
[...]
//DATA     DD  DSN=NENTWIG.DATA.TRANLOGD,DISP=SHR
  LOAD DATA INDDN DATA
    REPLACE
    LOG NO
    ENFORCE NO
    FORMAT DELIMITED
    INTO TABLE NENTWIG.TRANSACTION_LOG (
        MKT_BSKT_TXN_ID INT,
        POS_TML_ID FLOAT,
        TXN_STRT_TMS VARCHAR(40),
        TXN_BOOK_DT DATE
   )
```

Wenn alle Tabellen erstellt und geladen sind, kann der entsprechende Stream in SPSS Modeler ausgeführt werden. Außerdem ist damit die Grundlage für den Offload auf den DB2 Analytics Accelerator geschaffen, die im folgenden Abschnitt 4.2 beschrieben wird.
Bei der Durchführung dieser Teilschritte sind einige Fehler aufgetreten. Die Ursachen dafür können im Nachhinein unter Umständen als trivial erscheinen, sind aber teilweise der Portierung der Blueprints von DB2 für Linux, UNIX und Windows auf DB2 für z/OS geschuldet. Ein Beispiel dafür ist, dass die SQL Statements im Datenbankschema nicht die wichtige Großschreibung aller Schlüsselwörter beachten, in DB2 führt die Kleinschreibung allerdings dazu, dass die entsprechenden Tabellen nicht erstellt werden. Bei den Datentypen muss vor allem beachtet werden, dass in DB2 auf z/OS kein DOUBLE, sondern FLOAT oder DECIMAL verwendet wird. Eine weitere Fehlerquelle wurde bereits im Abschnitt 4.1.1 beschrieben. DB2 für z/OS akzeptiert als Datumsformat nur „YYYYMMDD“, hier ist DB2 für Linux, UNIX und Windows flexibler. In dem Skript, in dem die Daten in die Tabellen geladen werden, müssen zudem unbedingt alle Attribute der Tabelle erneut angegeben werden, ansonsten schlägt das Laden fehl. Jede Tabelle benötigt zudem ihren eigenen Tablespace, was nach vielen Variationen bei der Wahl der Parameter im LOAD-Prozess festgestellt wurde. Mit dieser Anpassung sind weiterhin folgende Fehler weggefallen:

Verschiedene Unicode Fehler wie etwa `CHAR CONVERSION FROM CCSID 65534 to 1208 FAILED`. Dabei habe die Feststellung gemacht werden, dass beim Laden von Daten in Unicode-Tabellen ohne Angabe von Encoding-Parametern eine Konvertierung nach Unicode stattfindet [db212].

Tablespace im Status Recovery Pending und damit kein Zugriff auf die Daten möglich, dies ist beispielsweise bei mehreren `LOAD` nacheinander auf dem gleichen Tablespace, aber auch bei Unicode-Fehlern, aufgetreten. In der `DB2 system administration` kann dies unter `Display/terminate utilities` überprüft werden, wofür auf der Kommandozeile folgender Befehl eingegeben wird: `-DIS DB(RETAIL) SPACENAM(TSTRANLO) LIMIT(*)`. Der Einfachheit halber wurde beim Status `RECP` (Recovery Pending) der Tablespace neu angelegt.

Nach dem „Submit“ eines JCL-Skriptes wird dieses abgearbeitet und bei kleineren Aufgaben erfolgt innerhalb weniger Sekunden eine Rückmeldung über einen Fehlercode, der Aufschluß über den Erfolg oder Mißerfolg bei der Ausführung gibt, ein Beispiel dafür ist folgende Meldung:

```
19.28.12 JOB08815 $HASP165 NENTWIGL ENDED AT BOEDWB1
MAXCC=0004 CN(INTERNAL)
```

Interessant ist der Wert hinter `MAXCC`, bei 0 ist alles erfolgreich verarbeitet, bei 4 sind Warnungen aufgetreten und bei 8, 12 oder 16 sind Fehler aufgetreten, die eine erfolgreiche Ausführung verhindern. In einem der Fehlerfälle wird das SDSF (System Display and Search Facility) als zentralen Anlaufpunkt für die Job-Logs verwendet, um nach der Ursache des Fehlers zu suchen. Desweiteren hilft bei manchen Fehlern die eben erwähnte „DB2 system administration“, meist ist jedoch das SDSF nützlicher. Die Arbeit mit den Logs ist jedoch sehr zeitaufwändig, da es nahezu keine Aufbereitung der Daten gibt und die Logs unter Umständen sehr groß werden können. Ein Ausschnitt aus einem Log von einem JCL-Skript zum Laden von Daten ist in Abbildung 4.3 zu sehen, dieser zeigt einen JCL Fehler, durch welchen die Ausführung des Skriptes verhindert wird.

4.2 Integration in IDAA

Die Integration in einen angeschlossenen DB2 Analytics Accelerator beinhaltet grunsätzlich zwei Schritte. Zum einen das Übertragen der Daten auf den Accelerator, und zum anderen die Untersuchung der verwendeten Algorithmen daraufhin, welche Anpassungen auf dem Accelerator zur Berechnung mit IBM

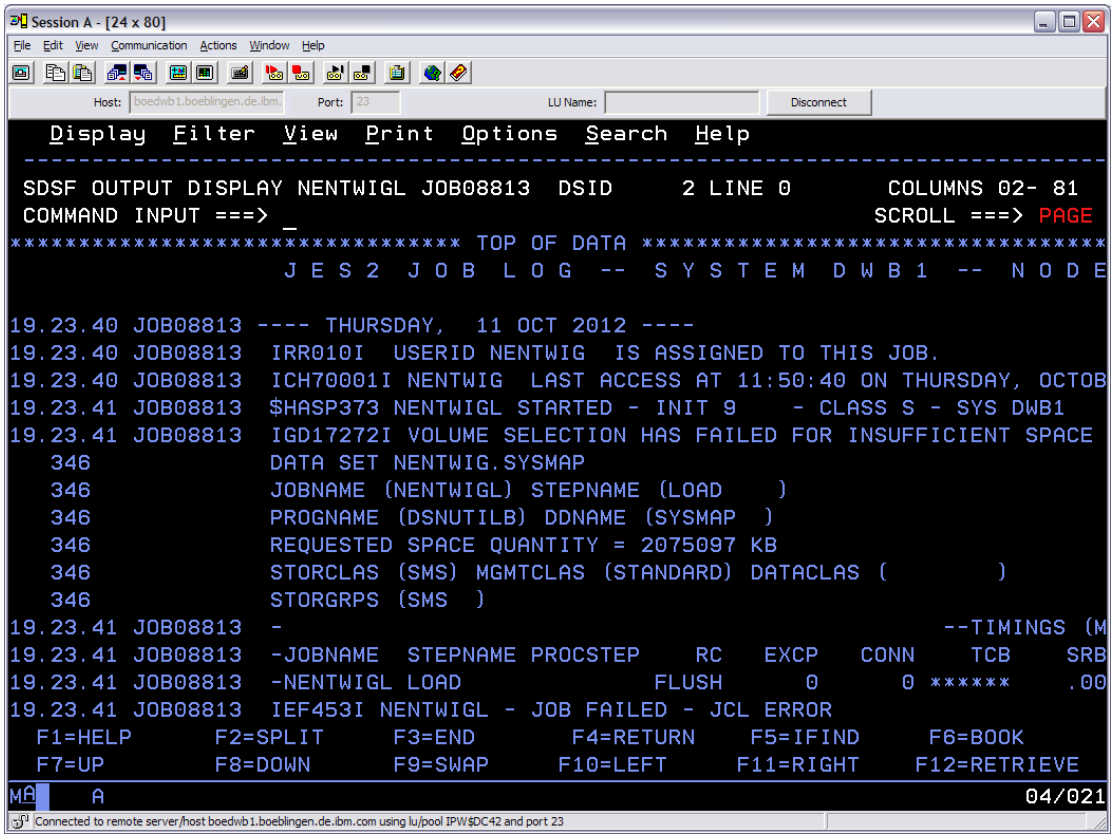

Abbildung 4.3: SDSF-Subsystem, zu sehen ist ein Ausschnitt aus einem Log für einen Job `NENTWIGL`, der ein `LOAD` ausführen soll. In diesem Beispiel ist der Fehler leicht zu finden, da es ein zeitiger und schwerer Fehler in der fünften Zeile ist: `VOLUME SELECTION HAS FAILED FOR INSUFFICIENT SPACE`

Netezza Analytics (INZA) nötig sind. Der Transfer der Daten auf den Accelerator wird im Abschnitt 4.2.1 beschrieben, die wesentliche Aufgabe ist jedoch die Überprüfung und Anpassung der Algorithmen im Abschnitt 4.2.2 mit Fokus auf RFM-Analyse und die Assoziationsanalyse.

4.2.1 Offload auf IDAA

Die wichtigste Grundlage für einen Offload von Tabellen auf das angeschlossene IDAA ist, dass die Daten bereits auf dem DB2 verfügbar sind. Mit der Abbildung der Szenarien auf das DB2 wie in Abschnitt 4.1 beschrieben ist die Vorbereitung abgeschlossen. Mit dem auf Eclipse aufbauenden IBM DB2 Analytics Accelerator Studio können mit dem System verbundene Beschleuniger verwaltet werden, dazu gehört etwa das initiale Laden der Daten oder die Vergabe von Distribution Keys. Ein Beispiel für den Einsatz von Distribution Keys

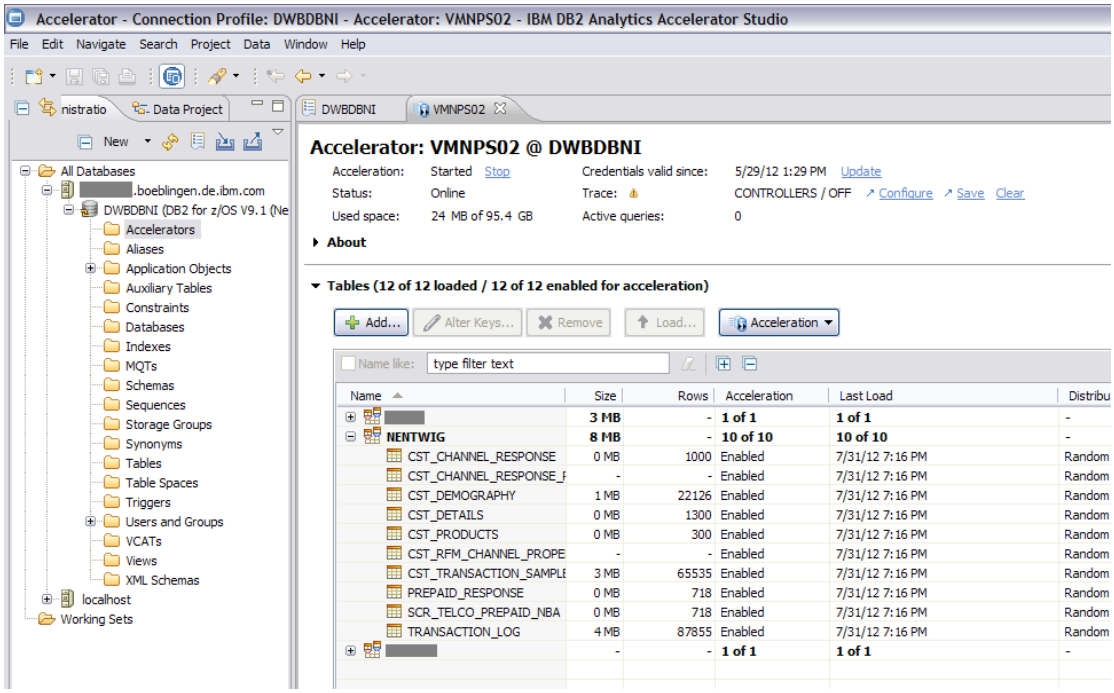

Abbildung 4.4: IBM DB2 Analytics Accelerator Studio, welches die beschleunigten Tabellen auf dem Accelerator VMNPS02 vom Nutzer NENTWIG anzeigt. Das IDAA Studio bietet Möglichkeiten zur Verwaltung von mehreren IDAA-Systemen sowie Optimierungen etwa bei der Verteilung der Tabelleninhalte über Distribution Keys.

kann sein, dass ein bestimmtes, möglichst gleich verteiltes Attribut einer Faktentabelle gewählt wird, wodurch eine ähnliche Verteilung der Daten auf den

Snippet Blades erreicht werden kann. In Abbildung 4.4 wird das DB2 Analytics Accelerator Studio verwendet, um die beschleunigten Tabellen auf dem DB2-Subsystem `DWBDBNI` anzuzeigen. Standardmäßig werden die beschleunigten Tabellen pro Benutzer angezeigt, in diesem Falle werden alle Tabellen angezeigt, die für den Benutzer `NENTWIG` existieren und in den beiden bearbeiteten Szenarien benötigt werden. [ida11]

4.2.2 Anpassung von Algorithmen

In disem Abschnitt werden die Änderungen beschrieben, um den Algorithmus RFM-Analyse sowie das Modell der Assoziationsanalyse auf dem DB2 Analytics Accelerator durchführen zu können. Dabei wird eine Kombination aus SQL-Statements und INZA-Befehlen verwendet, etwa um die Datensätze für die benötigte INZA-Befehlsstruktur vorzubereiten und dann das Modell zu berechnen. Die INZA-Befehle stammen dabei aus dem IBM SPSS In-Database Analytics Paket [sps12a] und werden wie die SQL-Statements auf Netezza über `nzsql` (IBM Netezza SQL Terminal) ausgeführt.

4.2.2.1 RFM-Analyse

Ein Baustein des Retail Prediction Blueprint ist die RFM-Analyse, die im Abschnitt 3.3.1 beschrieben wird. Alle nötigen Anpassungen, um die Berechnung der RFM-Analyse auf dem DB2 Analytics Accelerator durchführen zu können, werden nachfolgenden beschrieben. An dieser Stelle wird der Algorithmus inklusive der Datenvorbereitung beschrieben. Diese Anpassung ist nötig, damit die Daten der Analyse im richtigen Format zugeführt werden. In SPSS Modeler entspricht die Vorbereitung dem RFM-Aggregatknoten. Zudem müsste für einen echten Einsatz des Szenarios jede Transaktion einzigartig sein, nur dann kann die `CUST_ID` als identifizierendes Merkmal (etwa für die Frequency) herangezogen werden. Dies ist über einen einfachen `DISTINCT` möglich. Prinzipiell müssen danach bei der RFM-Analyse für jeden der Bereiche Recency, Frequency und Monetary Value zwei Teilschritte bearbeitet werden, um eine Diskretisierung der Werte vorzunehmen: Das Festlegen der Grenzen für eine wählbare Anzahl diskreter Wertebereiche und danach das Zuordnen der einzelnen Datensätze in diese Bereiche.
Um den ersten Schritt abarbeiten zu können, wird eine Tabelle mit den Eingabewerten angelegt, dabei ist die Besonderheit zu beachten, dass die Diskretisierungsalgorithmen unter Netezza nicht mit dem Datentyp Datum umgehen kann, es werden nur numerische Werte zugelassen. Im Vergleich dazu kann SPSS Modeler auch mit Datumsangaben richtig diskretisieren. Um die Tabellenspalte `TXN_BOOK_DT` als Eingabe verwenden zu können, wird eine zusätz-

liche Spalte RECENCY_DAYS angelegt. Mit „-“ als Operator bei zwei Datumsangaben beträgt das Ergebnis den Abstand der Datumswerte in Tagen, was als INT gespeichert werden kann. Als erster Operand wird dann ein festes Datum ausgewählt, welches zeitlich vor jedem Datum in den Quelldaten vorkommt. Dies ist wichtig, weil damit die Anzahl der Tage mit der Aktualität des Datums zunimmt. Dadurch werden die Werte beim Einordnen in die berechneten Grenzen nicht falsch herum einsortiert. Mit diesem wird die vorbereitete Tabelle SOURCE wie folgt befüllt:

```
INSERT INTO SOURCE
SELECT
    :TRANLOGL.TXN_BOOK_DT AS RECENCY,
    '1990-01-01' - :TRANLOGL.TXN_BOOK_DT AS RECENCY_DAYS,
    :TRANSAML.CUST_ID AS CUST_ID,
    :TRANLOGL.TOT_SALE_AMT AS MONETARY
FROM
    :TRANSAML
INNER JOIN
    :TRANLOGL
ON (
    :TRANSAML.MKT_BSKT_TXN_ID=
    :TRANLOGL.MKT_BSKT_TXN_ID
);
```

Recency und Monetary Value sind dadurch vorbereitet, für die Spalte Frequency wird die Anzahl der Käufe von jedem Kunden verwendet, was über ein Update durchgeführt wird.

```
UPDATE SOURCE SET FREQUENCY = FREQ
    FROM (
        SELECT
            :TRANSAML.CUST_ID AS CUST_ID,
            COUNT(*) AS FREQ
        FROM
            :TRANSAML
        GROUP BY CUST_ID
    ) AS TEMP
    WHERE TEMP.CUST_ID = SOURCE.CUST_ID;
```

Damit können die Grenzen für die Diskretisierung berechnet werden, wofür auf Netezza verschiedene Möglichkeiten zur Auswahl stehen: EWDISC verwendet „Equal Width“ also gleiche Breite der Klassen, wohingegen EFDISC „Equal

Frequency“ beachtet, also bei der Berechnung der Grenzen auch die Häufigkeit der einzelnen Werte mit in Betracht zieht [sps11a]. Dieses zweite Verfahren wird im Beispiel für die Daten aus der Tabellenspalte `TOT_SALE_AMT` verwendet, um der Struktur der Daten gerecht zu werden. Detailliert zu sehen ist dieses Histogramm in der Abbildung 4.5 mit einer Beschränkung auf maximal 300 Kunden auf der Ordinate erkennen lässt, dass sich die Kaufbeträge vor allem im unteren Drittel aufteilen. Speziell ist zu erkennen, dass die Kaufbeträge im Bereich 0-500 an die obere Grenze stoßen. Die Anzahl der Kunden erreicht in diesem Bereich pro Balken teilweise mehr als 60000, `EFDISC` findet auf diesen Daten Grenzen für den Kaufbetrag, womit jedem Bereich eine etwa gleiche Anzahl von Kunden zugeordnet werden kann. Damit kann `EFDISC` unter Angabe der Parameter `incolumn` für die zu evaluierenden Spalten sowie `bins` für die Anzahl der Klassen ausgeführt werden.

```
CALL nza..EFDISC('outtable=RFM_BOUNDS,
    intable=SOURCE,
    incolumn=RECENCY_DAYS;FREQUENCY;MONETARY,
    bins=5
');
```

Für den Monetary Value sehen die Grenzen wie folgt aus.

$$
\begin{aligned}
\text{Monetary Value} &< 11{,}83 && \rightarrow \text{Kategorie 1}\\
11{,}83 \leq \text{Monetary Value} &< 21{,}86 && \rightarrow \text{Kategorie 2}\\
21{,}86 \leq \text{Monetary Value} &< 37{,}1 && \rightarrow \text{Kategorie 3}\\
37{,}1 \leq \text{Monetary Value} &< 69{,}15 && \rightarrow \text{Kategorie 4}\\
\text{Monetary Value} &\geq 69{,}15 && \rightarrow \text{Kategorie 5}
\end{aligned}
$$

Damit kann die Einteilung der Ausgangswerte in die entsprechenden Kategorien durchgeführt werden, wofür der folgende INZA-Aufruf verwendet wird.

```
CALL nza..APPLY_DISC('outtable=RFM,
                     intable=SOURCE,
                     btable=RFM_BOUNDS,
                     replace=false
');
```

Die resultierende Tabelle `RFM` beinhaltet drei neue Spalten mit den entsprechenden Klassifikationen, für welche in 4.1 drei Zeilen auszugsweise angegeben werden.

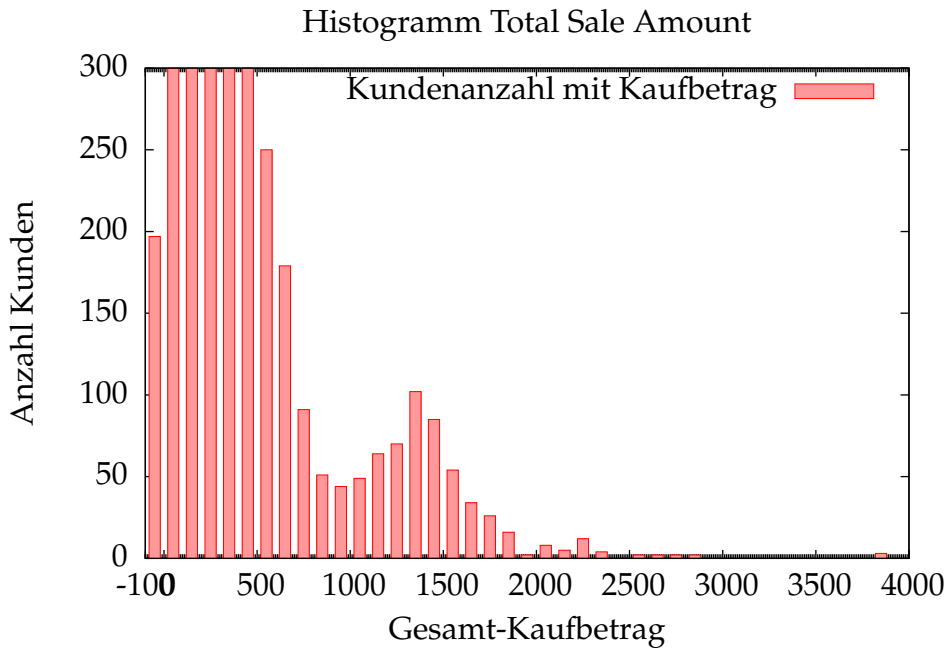

Abbildung 4.5: Die Verteilung der Kaufbeträge ist vor allem auf den Bereich bis etwa 500 Dollar beschränkt, die dargestellten Balken erreichen dort die obere Grenze des Histogramms und der zweite Balken (0 bis 100) wäre mit einer Kundenanzahl von über 60000 viel höher. Um in dem restlichen Bereich überhaupt Datensätze zu erkennen, wurde die Begrenzung auf 300 Kunden gewählt.

Recency Score	Frequency Score	Monetary Score
1	4	1
3	2	1
5	4	5
...	...	...

Tabelle 4.1: Beispielwerte für die Klassifizierung bei der RFM-Analyse, Werte rangieren im Bereich 1-5, wobei 5 die profitableren Kunden widerspiegelt.

Die letzte Aufgabe ist es dann, den kompletten `RFM Score` zu ermitteln, was über eine einfache Addition der gewichteten Einzelwerte vollzogen wird, womit die RFM-Analyse abgeschlossen wird.

```
UPDATE RFM SET 'RFM SCORE' = DISC_RECENCY_DAYS * 100
                           + FREQUENCY * 10
                           + MONETARY;
```

4.2.2.2 Assoziationsanalyse

Aus dem gewählten Szenario des Telecommunications Blueprint ist die Assoziationsanalyse der interessante Schritt, da die restlichen Schritte über einfaches SQL erledigt werden können. Die Assoziationsanalyse wird unter SPSS Modeler mit dem Apriori-Algorithmus umgesetzt, auf Netezza allerdings mit dem FP-growth-Algorithmus. Für die Verwendung von FP-growth müssen die Daten aufbereitet werden, was im nächsten Beispiel erklärt wird. In der Tabelle `CST_PRODUCTS` sind die Daten so angeordnet wie in Tabelle 4.2, jeder Kundennummer `CUSTID` folgen die jeweiligen Tarif-Optionen über eine einfache Zuordnung „Y“ / „N“ für gebucht / nicht gebucht.
Für den Algorithmus `ARULE` zur Erstellung der Assoziationsregeln auf Netezza benötigen die Daten allerdings die Darstellung wie in Tabelle 4.3. Dabei wird jede einzelne Option mit „Y“ auf eine eigene Zeile geschrieben, da die gewählten Tarif-Optionen zu den Assoziationen führen sollen.
Die Umsetzung dieser Aufgabe wird auf Netezza mit dem folgenden SQL-Statement, wiederholt für die fehlenden Optionen, in die Tabelle `ITEMS` geschrieben. `:CSTPROD` ist dabei eine Variable, in `nzsql` definiert über `set CSTPROD '"CST_PRODUCTS_ID_38"'`, womit auf die entsprechende IDAA-Tabelle von `CST_PRODUCTS` verwiesen wird.

CUSTID	VOICE	SMS	DATA	ADV DATA	DISTANCE CALLING
1001	Y	Y	Y	Y	Y
1002	N	Y	Y	N	N
1003	Y	Y	N	Y	N
1004	Y	Y	Y	N	N
1005	Y	N	Y	N	N
1006	Y	Y	N	Y	Y
1007	Y	Y	Y	N	Y
1008	N	N	N	N	N
1009	Y	Y	Y	N	N
1010	Y	Y	N	N	N

Tabelle 4.2: Die Auflistung der Tarif-Optionen für Kunden erfolgt über die einfache Angabe von „Y“ / „N“ in der jeweiligen Spalte.

ITEMS	
TID	ITEM
1001	VOICE
1001	SMS
1001	DATA
1001	ADV DATA
1001	DISTANCE CALLING
1002	SMS
1002	DATA
1003	VOICE
1003	SMS
1003	ADV DATA
...	...

Tabelle 4.3: Jede aktive Tarif-Option wird als Vorbereitung für FP-growth auf Netezza einzeln mit der dazugehörigen Kundennummer auf einer Zeile der Tabelle `ITEMS` gespeichert.

```
INSERT INTO ITEMS (
    TID, ITEM
)
SELECT
    CST_ID, 'VOICE'
FROM :CSTPROD
```

```
    WHERE VOICE_PACKAGE = 'Y';
```

Wenn alle Tarif-Optionen nach dem obigen Schema verarbeitet sind, kann das Modell mit den nötigen Parametern erstellt werden. In diesem Falle wurde als minimaler Support einer Regel 5% gewählt, dies ist der Mindestprozentsatz aller Transaktionen, in dem die Regel vorkommen muss. Zudem ist die Konfidenz mit 50% festgelegt, das heißt, in mindestens 50% der Fälle führt die Wahl der ersten Option auch zu der damit assoziierten Tarif-Option. Die `TID` spiegelt die Kundennummer wieder, womit die verschiedenen Optionen zugeordnet werden können. Der Aufruf zur Modellerstellung lautet wie folgt und wird mit `TELCO` benannt. [sps11a]

```
CALL nza..ARULE('intable=ITEMS,
    tid=TID,
    model=TELCO,
    support=5,
    confidence=0.5
');
```

Die berechneten Regelsätze können dann über den Befehl

```
CALL nza..PRINT_ARULE('model=TELCO');
```

angezeigt werden, was für jede der 51 Regeln in dem folgenden Format geschieht:

```
GRP=| {2G data package,SMS package} -> {Voice package}
[supp=0.30201342281879, conf=0.78947368421053, ...]
```

Dabei sind die ersten Tarif-Optionen die Bedingungen, die mit dem Wert `conf` (Konfidenz) zum resultierenden `Voice package` führen. Diese Regel ist zudem in rund 30,2% (`supp`-Wert) aller Transaktionen zu finden. Die Auswertung sowie ein Vergleich mit der entsprechenden Berechnung auf SPSS Modeler findet im anschließenden Kapitel 5 statt.

4.3 Optimierung

Bereits bei der Beschreibung des neuen Prototypen im Abschnitt 3.2 sind einige Herausforderungen benannt, die mit der erweiterten Nutzung des DB2 Analytics Accelerator etwa zur Modellerstellung im Data-Mining entstehen. Eine dieser Aufgaben betrifft die Aktualisierung der berechneten Modelle, wenn die

Datenbasis erneuert wird. Das Szenario zum Update der Daten kann dabei wie folgt aussehen. Die im DB2 vorgehaltenen Daten werden zu einem bestimmten Zeitpunkt inital auf den IDAA geladen. Ab diesem Zeitpunkt können die Daten zum Data-Mining verwendet werden, so auch zur Erstellung von Modellen für Predictive Analytics. Bei einem Update der Datengrundlage auf dem IDAA, etwa alle Transaktionen des letzten Tages, werden nur diese neuen Informationen auf den Accelerator geladen. Anknüpfend erfolgt mithilfe der neuen Daten und der vorher vorhandenen Datenbasis eine Neuberechnung der abhängigen Modelle, um die Vorhersagen auf einen neuen Stand zu bringen. Dabei ist zudem zu beachten, dass die Verfahren möglichst gut parallelisierbar sein sollen, damit sie die vorhandene Hardware in dem DB2 Analytics Accelerator ausnutzen. Im Laufe der Untersuchungen hat sich gezeigt, dass nicht jeder Algorithmus, der für inkrementell aufbauende Datenbestände geeignet ist, auch gut parallelisierbar ist. Auf der anderen Seite gibt es Algorithmen wie FDM, die parallelisierbar sind, dafür nicht mit inkrementellen Datenänderungen umgehen können [CHN+96]. Im folgenden wird daher neben FP-growth auf die Eignung von zwei weiteren Ansätzen eingegangen.

FP-growth Die Erstellung von Assoziationsregeln findet in der Praxis oft statt, beispielsweise im Rahmen von Warenkorbanalysen zur Empfehlung von Produkten. Auch bei einer Untersuchung in [WKRQ+07], welche Data-Mining-Algorithmen oft in Forschung und Praxis verwendet worden sind, erreichte die Kategorie Assoziationsanalyse mit Vertretern wie Apriori und FP-growth den vierten Platz. Im folgenden wird für die Assoziationsanalyse untersucht, inwiefern die Erstellung von Modellen mithilfe von inkrementellen Algorithmen optimiert werden kann. Auf dem DB2 Analyitics Accelerator wird der Algorithmus FP-growth verwendet, um Assoziationsmodelle zu bauen [sps11a]. Der im Abschnitt 2.4.2.2 beschriebene FP-growth eignet sich nicht direkt für inkrementelles Mining von Assoziationsregeln, der Grund dafür liegt bei der Erstellung des FP-tree. In diesem dürfen dann keine Items gelöscht werden, die den minimalen Support nicht erreichen. Demnach werden an dieser Stelle alle Items gespeichert, auch die, die etwa nur einen Support von 1 haben. Eine Aussortierung dieser Itemsets erfolgt dann erst später bei der tatsächlichen Generierung der Regeln. Neue Transaktionen können auf diese Art und Weise eingefügt werden und für bisher nicht häufige Items wird der Support erhöht. Der FP-tree wird dann nur neu erstellt, wenn die bisher nicht häufigen Itemsets eine bestimmte Schwelle überschreiten. Davor werden die häufigen Itemsets samt ihrer Supportwerte aktualisiert und die bisher nicht häufigen zur Regelerstellung ignoriert. Bei diesem Ansatz kann die Größe des temporären FP-tree schnell zu groß werden, zudem erfolgt bei Überschreitung der Schwelle für nicht häufige Items eine Neuberechnung. [HPY00]

Compressed and Arranged Transaction Sequences Tree (CATS) Eine Erweiterung für den FP-growth-Algorithmus ist durch Cheung et al. [CZ03] entstanden. Dabei wird anstatt der Datenstruktur FP-tree der CATS Tree erstellt, der eine komprimierte Abbildung aller Transaktionen darstellt. Jede einzelne Transaktion wird dafür in den Baum ähnlich wie bei FP-growth eingefügt. Dabei werden einzelne Knoten bei Bedarf in Richtung Wurzel verschoben, um eine bessere Kompression zu erreichen. Dies erfolgt nur, wenn das Vorkommen des zu verschiebenden Knotens mindestens so häufig ist wie die Häufigkeit aller auf dem Weg vorkommenden Knoten vorgibt. Kindknoten haben wie bei FP-growth mindestens die Häufigkeit des darüberliegenden Knotens. Zudem werden die Kinder auf einer Ebene nach der Häufigkeit sortiert, damit eine schnellere Einordnung neuer Transaktionen stattfinden kann. Die Besonderheit ist demnach zum einen eine optimierte Einordnung der einzelnen Knoten für eine bessere Komprimierung und zum anderen die Abbildung aller Transaktionen. Damit besteht die Möglichkeit, inkrementelle Updates auf der Datenbasis durchzuführen und nur die neuen Transaktionen in den CATS Tree aufzunehmen. Desweiteren können bereits eingefügte Transaktionen aus dem Baum entfernt werden, etwa wenn alte Transaktionen nicht mehr beachtet werden sollen. Mit diesem CATS Tree als Grundlage können die Regelsätze mit dem Algorithmus FELINE berechnet werden. Die Regeln werden demnach immer nach dem aktuellen Stand des CATS Tree erstellt, ohne dass die Datenbank komplett durchsucht werden muss. [CZ03]
Bei einer parallelen Ausführung mit dem IBM DB2 Analytics Accelerator kann auf jedem Snippet Blade die vorhandene Datenmenge in CATS Teilbäume eingeteilt werden. Diese Teilbäume können zusammengefasst werden und geben eine komplette Repräsentation des Datenbestandes wider. Insgesamt kann es beim Aufbau von großen Bäumen mit CATS oder FP-growth zu Hauptspeicherengpässen kommen [LLC05].

Sliding Window Filtering (SWF) Eine interessante Alternative ist ein Algorithmus von Lee et al. [LLC05], bei welchem die Datenmenge in Partitionen eingeteilt wird, die zum Beispiel Zeitschritten entsprechen. Eine Partition für jeden Monat im Jahr kann dann etwa alle Transaktionen für einen Monat abbilden. Zu Beginn eines neuen Monats wird die älteste Partition entfernt und der neueste Monat zu den Partitionen hinzugefügt. Wenn die Daten von einem Monat dann zum Beispiel auf je ein Snippet Blade des DB2 Analytics Accelerators übertragen wird, können jeweils lokal die Kandidaten für die Assoziationsregeln sowie die lokalen Parameter wie Support bestimmt werden. Dabei fließen die Ergebnisse des jeweils vorherigen Zeitschrittes mit in die Berechnung ein, wodurch sich eine Abhängigkeit für die parallele Berechnung ergibt. Die zuerst unabhängig durchgeführte lokale Berechnung kann jedoch durchgeführt wer-

den. Der Algorithmus wird als Sliding Window Filtering bezeichnet, wobei die Einflußnahme älterer Datensätze in den jeweils aktuellen Zeitrahmen einfließt, die aktuellen Partitionen in der Berechnung allerdings einen deutlich größeren Einfluß haben. Insgesamt arbeitet SWF bei vielen neuen Daten sowie einer allgemein großen Datenbasis besser. Ähnlich wie bei Apriori erfolgt in jeder Partition die Generierung von Kandidaten, allerdings mit einigen Optimierungen bei der Einschränkung der Kandidatenwahl. [LLC05]

5 Evaluation

In diesem Kapitel erfolgt zu Beginn eine Beschreibung der verwendeten Testumgebung sowie der Besonderheiten. Dem folgend werden im Abschnitt 5.2 Ergebnisse präsentiert, die sich bei der Umsetzung der Algorithmen von IBM SPSS Modeler auf den DB2 Analytics Accelerator gezeigt haben. Im Abschnitt 5.3 wird dann dargestellt, welche Algorithmen in einer realen Testumgebung zum Einsatz gekommen sind.

5.1 Setup

Die im Kapitel Umsetzung beschriebenen und angepassten Algorithmen sind auf einem IDAA-Simulator entstanden. Damit konnten Arbeitsschritte getestet und überprüft werden. Ein nächster Schritt ist die Verwendung von Algorithmen auf einem DB2 Analytics Accelerator mit INZA. Diese Aufgabe wurde bei IBM von Oliver Benke durchgeführt, da Studenten nicht auf die entsprechende Hardware zugreifen durften. Aufgrund dieser Tatsache werden in den folgenden Ergebnissen Algorithmen (etwa die Assoziationsanalyse) vorgestellt, die danach nicht für Messungen auf den Testsystemen eingesetzt wurden. Zudem gibt es aus diesem Grund für den Entscheidungsbaum nur eine Beschreibung im Rahmen der Messungen, nicht jedoch bei den Ergebnissen. Als Hardware für die Performancemessung mit SPSS Modeler werden dabei vier IFLs[2] auf einem System z196 verwendet. Der SPSS Modeler Server wird dementsprechend auf einem Linux for System z eingesetzt. Vergleichend dazu findet der zweite Benchmark auf einem DB2 Analytics Accelerator mit zwölf Snippet Blades statt. Dabei wird das INZA-Paket zur Ausführung von analytischen Funktionen verwendet. In jedem Abschnitt erfolgt eine kurze Beschreibung der jeweils verwendeten Algorithmen.

5.2 Ergebnisse

Verglichen werden in diesem Abschnitt die Ergebnisse, die bei der Umsetzung der Algorithmen Assoziationsanalyse und RFM-Analyse jeweils bei der Imple-

[2] Integrated Facility for Linux: Spezieller Prozessor für die Verwendung von Linux.

mentierung auf SPSS Modeler und INZA auf dem DB2 Analytics Accelerator gewonnen werden konnten.
Dabei wird mit der *Assoziationsanalyse* aus dem Telekommunikations-Blueprint begonnen. Als Datengrundlage wurden auf beiden Plattformen 300 Kundendaten auf Assoziationen untersucht. Sowohl der Apriori-Algorithmus unter SPSS Modeler als auch der FP-growth-Algorithmus im INZA-Paket finden bei gleichen Support- und Konfidenzparametern je 51 Regeln. Auszugsweise werden in Tabelle 5.1 vier der Regeln detailliert verglichen. Dazu sind im oberen Bereich der Tabelle die Tarif-Optionen als *Bedingung* aufgelistet, die mit der Wahrscheinlichkeit der Konfidenz zur Wahl der *Konsequenz* führen. Bei der Gegenüberstellung der Konfidenzwerte fällt auf, dass die Werte unter SPSS Modeler und auf dem DB2 Analytics Accelerator gleich sind. Anders sieht dies bei den Supportwerten aus, hier gibt es bei den Beispielwerten Abweichungen von bis zu $0{,}25\%$. Im Vergleich mit weiteren Regeln kann zudem erkannt werden, dass der Unterschied der Supportwerte zunimmt, umso weniger Bedingungen zu einer Konsequenz führen.

Bedingung	2G DATA ADV DATA	2G DATA DISTANCE SMS VOICE	DISTANCE	DISTANCE ADV DATA SMS
Konsequenz	VOICE	ADV DATA	SMS	VOICE
Konfidenz Nz	84,62%	53,49%	75,33%	86,21%
Konfidenz SPSS	84,62%	53,49%	75,33%	86,21%
Support Nz	22,15%	7,72%	37,92%	16,78%
Support SPSS	22,0%	7,67%	37,67%	16,67%
Differenz	0,15%	0,05%	0,25%	0,11%

Tabelle 5.1: Vergleich von Ergebnissen für vier ausgewählte Assoziationsregeln auf SPSS Modeler respektive IDAA. Bei den Konfidenzwerten sind keine Unterschiede zu erkennen, die Supportwerte variieren und sind auf Netezza/IDAA im Allgemeinen höher.

Die *RFM-Analyse* ist ein Teil des beschriebenen Szenarios aus dem Retail Blueprint und wurde für die Messungen auf dem DB2 Analytics Accelerator mit Hilfe von INZA-Befehlen und SQL-Statements umgesetzt. In 5.2 lassen sich die Ergebnisse für Recency und Monetary Value vergleichen, zu erkennen ist dabei, dass die Werte unter SPSS Modeler generell höher sind als das entsprechende IDAA Pendant. Allerdings scheinen die Unterschiede zu groß, um sie etwa auf Rundungsfehler unter SPSS Modeler zurückzuführen. Eventuell ist die ge-

wählte `EFDISC`-Verteilung unter INZA nicht die, die auch unter SPSS Modeler verwendet wird.

	Monetary Value		Recency	
	IDAA	SPSS	IDAA	SPSS
Grenze 1	11,83	14,52	2002-02-28	2002-03-01
Grenze 2	21,86	25,28	2002-07-25	2002-07-27
Grenze 3	37,10	42,32	2003-01-30	2003-01-31
Grenze 4	69,15	82,84	2003-06-13	2003-06-15

Tabelle 5.2: Vergleich der Grenzen bei der RFM-Analyse mit SPSS Modeler respektive IDAA. Die Frequency ist nicht in dem Vergleich, weil die Daten durch die falsche Datenvorbereitung in SPSS Modeler nicht nutzbar waren.

Eine Ursache für *unterschiedliche Ergebnisse* im Allgemeinen können zum einen die verschiedenen Algorithmen ansich sein, eine weitere kann mit den verwendeten Datentypen in SPSS Modeler und Netezza zusammenhängen. So werden in SPSS Modeler zur Berechnung der Modelle bei numerischen Werten Ganzzahlen (Integer) oder Gleitkommazahlen (Real) verwendet, von z/OS als Decimal kommende Werte werden demnach umgewandelt in die entsprechende Gleitkommazahl. Durch diese Konvertierung und die weitere Verwendung dieser Zahlen für jedwede Modellerstellung kommen Ungenauigkeiten zustande, sodass die Nachkommastellen in SPSS Modeler nicht immer exakt sind. Auf Netezza berechnete Werte haben diese Probleme nicht, da hier Decimal als Datentyp mit festen Nachkommastellen unterstützt wird.
Für den *Entscheidungsbaum* wurde keine Umsetzung von SPSS Modeler auf den DB2 Analytics Accelerator durchgeführt. Dies hat die Ursache, dass unter SPSS Modeler der CHAID Algorithmus und somit ein nicht-binärer Baum zum Einsatz kommt. Im Gegensatz dazu ist unter INZA ein binärer Entscheidungsbaum nach CART implementiert. Damit sind die Ergebnisse nicht direkt vergleichbar. Allerdings können nicht-binäre Bäume immer in binäre Bäume umgewandelt werden, wie in [Dix10] beschrieben wird.

5.3 Verwendung von Algorithmen für Messungen

Die Verwendung von ausgewählten Algorithmen im Rahmen von ersten Performancemessungen findet auf einem System statt, welches von Kunden bisher nicht eingesetzt werden kann. Die von Oliver Benke durchgeführten Messungen gehören zu einer Machbarkeitsstudie, aufgrund dessen können an dieser Stelle keine Messwerte preisgegeben werden. Aus den in 3.3 beschriebe-

nen Szenarien sind drei Teilaufgaben für vergleichende Messungen ausgewählt wurden, welche nachfolgende erläutert werden.

Enscheidungsbaum Abbildung 5.1 zeigt eine Anwendung zur Berechnung von einem Entscheidungsbaum mit einfacher Datenvorbereitung in SPSS Modeler. Wie in 2.4.1 beschrieben, werden Entscheidungsbäume im Data-Mining oft eingesetzt und sind daher an dieser Stelle für eine erste Performancemessung geeignet. Dabei werden in den Eingabedatensätzen trennende Attribute gesucht, die für eine Einordnung zur korrekten `STORE_ID` hilfreich sind. Die zugehörige INZA-Operation auf IDAA lautet:

```
CALL SPSS.DECTREE('intable=CST\_TRANSACTION\_SAMPLE,
    coldeftype=nom, incolumn=unq_id_src_stm:cont;
    nm:nom; tot_sale_amt:cont; nbr_itm:cont;
    grp_nm:nom; cust_id:nom, target=store_id,
    id=unique_id, model=TEST, eval=gini,
    minimprove=0.01, minsplit=50, maxdepth=3');
```

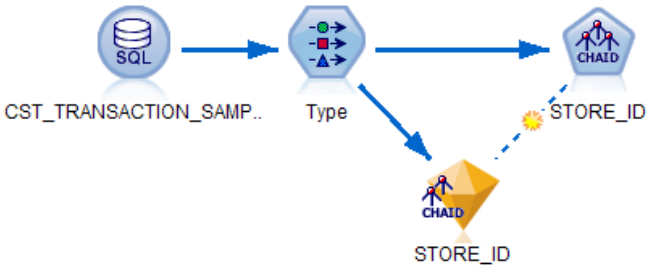

Abbildung 5.1: Stream SPSS Modeler Berechnung Entscheidungsbaum

Inner Join und Random Sample Die nächste Abbildung 5.2 zeigt den SPSS Modeler Stream einer typischen analytischen Aufgabe, die etwa auch im Szenario 3.3.1 vorkommt. Der Stream erstellt einen Inner Join zwischen zwei Datenbanktabellen, danach werden aus dem Ergebnis zufällig 1% der Datensätze ausgewählt und in die Ausgabedatenbank geschrieben. Die folgenden Zeilen führen die gleiche Aufgabe auf Netezza aus:

```
CREATE TABLE RETAIL_TEMP AS
    SELECT B.MKT_BSK_TXN_ID, A.NM, A.GRP_NM,
    A.STORE_ID, A.CUST_ID
    FROM "CST_TRANSACTION_SAMPLE-ID_328" a
```

```
        INNER JOIN "TRANSACTION_LOG-ID_330" B
        ON A.MKT_BSKT_TXN_ID = B.MKT_BSKT_TXN_ID;
CALL nza..RANDOM_SAMPLE('intable=RETAIL_TEMP1,
    fraction=0.01, outtable=SAMPLE');
```

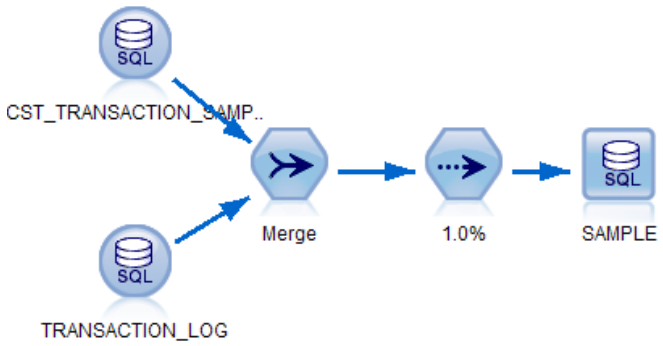

Abbildung 5.2: Stream SPSS Modeler Inner Join und Random Sample

RFM-Analyse Abbildung 5.3 zeigt einen Stream in SPSS Modeler zur Berechnung von RFM-Werten, für IDAA ist die Vorgehensweise bereits im Abschnitt 4.2.2.2 beschrieben.

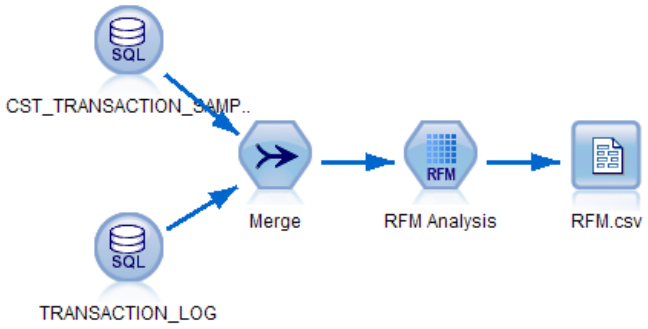

Abbildung 5.3: Stream SPSS Modeler RFM-Analyse

Als Fazit kann festgehalten werden, dass alle getesteten Algorithmen erfolgreich ausgeführt werden konnten. Die bisherigen Performancemessungen sehen zudem vielversprechend aus.

6 Zusammenfassung

Ziel dieser Master Thesis war unter anderem eine Beschreibung, wie die IBM Data-Warehouse-Appliance DB2 Analytics Accelerator verwendet werden kann, um Data-Mining-Prozesse in SPSS Modeler zu beschleunigen. Dafür wurden die prototypischen Anpassungen an den eingesetzten Komponenten beschrieben und zudem einige Probleme und Herausforderungen für eine weitere Entwicklung geschildert.

Darauf aufbauend wird anhand von zwei praxisnahen Szenarien je ein Geschäftsprozess als SPSS Modeler Stream umgesetzt und jeweils unter Anwendung von Data-Mining-Verfahren Modelle erschaffen, die in weiteren Prozessen wie Scoring zum Einsatz kommen können. Diese beiden Szenarien werden angepasst, damit sie in einer z/OS DB2 Umgebung genutzt werden können. An dieser Stelle hat sich gezeigt, dass DB2 auf z/OS einige Anpassungen an den Datensätzen erfordert. Im weiteren Verlauf erfolgt eine Abbildung ausgewählter Algorithmen auf den DB2 Analytics Accelerator. Im Rahmen des IBM Netezza Analytics Paketes (INZA) sind bereits viele Berechnungen direkt auf der Netezza-Datenbank möglich. Ein Vorteil von INZA im Gegensatz zu SPSS Modeler ist an dieser Stelle etwa, dass bei Berechnungen Dezimalzahlen mit exakten Nachkommastellen unterstützt werden. Jedoch sind einige Modelle wie Neuronale Netze bisher nicht für INZA verfügbar, bei anderen wie dem Entscheidungsbaum gibt es bisher nur eine Implementierung, womit die Auswahl beschränkt ist. Die auf SPSS Modeler verfügbare RFM-Analyse fehlte bisher im INZA-Paket. Eine Umsetzung der RFM-Analyse auf dem Accelerator sowie der ausführlichen Beschreibung der einzelnen Schritte erfolgt mit Hilfe bestehender INZA-Befehle und SQL-Statements zur Datenvorbereitung.

Vergleichend wurden dann SPSS Modeler Ergebnisse mit denen auf dem DB2 Analytics Accelerator gegenübergestellt, dies geschieht für die Assoziationsanalyse und die RFM-Analyse. Von nahezu identischen Ergebnissen bei der Assoziationsanalyse über erkennbare Unterschiede der Ergebnisse bei der RFM-Analyse hin zu nötigen Anpassungen für den Entscheidungsbaum gibt es eine breite Spanne. Insgesamt kann jedoch festgestellt werden, dass alle untersuchten Algorithmen mit INZA ausgeführt werden können und die bisherigen Performancewerte vielversprechend aussehen. Neben den Vergleichswerten der Algorithmen können die beschriebenen Szenarien zudem für weitere Aufgaben eingesetzt werden.

Abkürzungsverzeichnis

BI	Business Intelligence
CART	Classification and Regression Tree
CHAID	Chi-square Automatic Interaction Detectors
CICS	Customer Information Control System
CLEF	Component-Level Extension Framework
CLEM	Control Language for Expression Manipulation
DBMS	Datenbankmanagementsystem
DDL	Data Definition Language
DRDA	Distributed Relational Database Architecture
ETL	Extract, Transform, Load
FPGA	Field Programming Gate Array
IDAA	IBM DB2 Analytics Accelerator
IFL	Integrated Facility for Linux
INZA	IBM Netezza Analytics
ISAO	IBM Smart Analytics Accelerator
IRLM	Internal Resource Lock Manager
ISPF	Interactive System Productivity Facility
JCL	Job Control Language
JES	Job Entry System
KDD	Knowledge Discovery in Databases
LPAR	Logical Partition
ODBC	Open Database Connectivity
OLAP	Online Analytical Processing
OLTP	Online Transaction Processing
OSA	Open Systems Adapter
PDA	Program Development Facility
RACF	Ressource Access Control Facility
RFM	Recency, Frequency, Monatary Value
SMP	Symmetric Multiprocessor
WLM	Workload Manager

Literaturverzeichnis

[AIS93] AGRAWAL, Rakesh ; IMIELIŃSKI, Tomasz ; SWAMI, Arun: Mining association rules between sets of items in large databases. In: *SIGMOD Rec.* 22 (1993), Nr. 2, S. 207–216. – ISSN 0163–5808

[AMH08] ADÈR, Herman ; MELLENBERGH, Don ; HAND, David: *Advising on research methods: A consultant's companion*. Johannes van Kessel Publishing, 2008. – ISBN 97–890–79418–09–1

[Bal] BALLINGER, Carrie: *The Teradata Scalability Story*. – `https://www.cs.sunysb.edu/~sas/courses/cse532/fall01/teradata.pdf` zuletzt gesichtet: 28.10.2012

[BAP+06] BRUNI, Paolo ; ANDERS, Mark ; PARK, KyengDong ; RADER, Mark ; RUBY-BROWN, Judy: *Redbook: DB2 for z/OS: Data Sharing in a Nutshell*. IBM Corporation, 2006. – ISBN 0738496553

[BBF+12] BRUNI, Paolo ; BECKER, Patric ; FAVERO, Willie ; KALYANASUNDARAM, Ravikumar ; KEENAN, Andrew ; KNOLL, Steffen ; LEI, Nin ; MOLARO, Christian: *Redbook: Optimizing DB2 Queries with IBM DB2 Analytics Accelerator for z/OS*. IBM Corporation, 2012. – ISBN 0738437093

[BFAE12] BEYER, Mark A. ; FEINBERG, Donald ; ADRIAN, Merv ; EDJLALI, Roxane: Magic Quadrant for Data Warehouse Database Management Systems / Gartner, Inc. 2012. – Forschungsbericht. – `http://www.gartner.com/technology/media-products/reprints/teradata/vol3/article1/article1.html` zuletzt gesichtet: 15.10.2012

[BFOS84] BREIMAN, Leo ; FRIEDMAN, Jerome ; OLSHEN, R. A. ; STONE, Charles J.: *Classification and Regression Trees*. Monterey, CA : Wadsworth and Brooks, 1984. – ISBN 0412048418

[BG08] BAUER, Andreas ; GÜNZEL, Holger: *Data-Warehouse-Systeme: Architektur, Entwicklung, Anwendung*. Dpunkt.Verlag GmbH, 2008. – ISBN 9783898645409

[BL07] BENDEL, Peter ; LANG, Steffen: Create Web services for real-time scoring using DB2 Warehouse V9.5 / IBM Corporation. 2007. – Forschungsbericht. – `http://www.ibm.com/developerworks/data/library/techarticle/dm-0712bendel/` zuletzt gesichtet 23.10.2012

[CCK+00] CHAPMAN, Pete ; CLINTON, Julian ; KERBER, Randy ; KHABAZA, Thomas ; REINARTZ, Thomas ; SHEARER, Colin ; WIRTH, Rudiger: CRISP-DM 1.0 Step-by-step data mining guide / The CRISP-DM consortium. 2000. – Forschungsbericht. – http://www-01.ibm.com/support/docview.wss?uid=swg27023172 zuletzt gesichtet: 28.10.2012

[CDF+08] CASSIER, Pierre ; DEFENDI, Annamaria ; FISCHER, Dagmar ; HUTCHINSON, John ; MANEVILLE, Alain ; MEMBRINI, Gianfranco ; ONG, Caleb ; ROWLEYN, Andrew: *Redbook: System Programmer's Guide to: Workload Manager*. IBM Corporation, 2008. – ISBN 073848993X

[CHN+96] CHEUNG, David W. ; HAN, Jiawei ; NG, Vincent T. ; FU, Ada W. ; FU, Yongjian: A fast distributed algorithm for mining association rules. In: *Proceedings of the fourth international conference on on Parallel and distributed information systems*. Washington, DC, USA : IEEE Computer Society, 1996 (DIS '96). – ISBN 0–8186–7475–X, S. 31–43

[CZ03] CHEUNG, William ; ZAÏANE, Osmar R.: Incremental Mining of Frequent Patterns without Candidate Generation or Support Constraint. In: *7th International Database Engineering and Applications Symposium*, IEEE Computer Society, 2003. – ISBN 0–7695–1981–4, S. 111–116

[db210] *DB2 Utility Guide and Reference*. : *DB2 Utility Guide and Reference*. IBM Corporation, 2010. – Part Number: SC18-7427-10

[db212] *DB2 10 for z/OS Internationalization Guide*. : *DB2 10 for z/OS Internationalization Guide*. IBM Corporation, 2012. – http://publibz.boulder.ibm.com/epubs/pdf/iea2b510.pdf zuletzt gesichtet: 23.10.2012

[Dix10] DIXIT, J.B.: *Mastering Data Structures Through C Language*. Laxmi Publications, 2010. – ISBN 9380386729

[ECRS12] EBBERS, Mike ; CHINTALA, Dheeraj R. ; RANJAN, Priya ; SREENIVASAN, Lakshminarayanan: *Redbook: Real-time Fraud Detection Analytics on System z*. IBM Corporation, 2012. – Part Number: SG24-8066-00

[EKOO11] EBBERS, Mike ; KETTNER, John ; O'BRIEN, Wayne ; OGDEN, Bill: *Redbook: Introduction to the New Mainframe z/OS Basics*. IBM Corporation, 2011. – ISBN 0738435341

[HK00] HAN, Jiawei ; KAMBER, Micheline: *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2000. – ISBN 1–55860–489–8

[HKS04] HERRMANN, Paul ; KEBSCHULL, Udo ; SPRUTH, Wilhelm G.: *Einführung in z/OS und OS/390: Web-Services und Internet-Anwendungen für Mainframes*. Oldenbourg Wissenschaftsverlag GmbH, 2004. – ISBN 3–486–27393–0

[HL06] HILL, Thomas ; LEWICKI, Pawel: *Statistics: Methods and Applications. A Comprehensive Reference for Science, Industry, and Data Mining*. 1st. Tulsa, OK, USA : StatSoft, 2006. – ISBN 1884233597

[Hof12] HOFFMAN, Beth L.: Forecasting Via Predictive Model. In: *IBM Systems Magazine* (2012). – `http://www.ibmsystemsmag.com/power/businessstrategy/BI-and-Analytics/spss_c-ds/` zuletzt gesichtet 23.10.2012

[HPY00] HAN, Jiawei ; PEI, Jian ; YIN, Yiwen: Mining frequent patterns without candidate generation. In: *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. New York, NY, USA : ACM, 2000 (SIGMOD '00). – ISBN 1–58113–217–4, S. 1–12

[HR99] HÄRDER, Theo ; RAHM, Erhard: *Datenbanksysteme: Konzepte und Techniken der Implementierung*. Springer, 1999. – ISBN 3–540–65040–7

[IBM09] IBM INDUSTRY MODELS AND ASSETS SOFTWARE GROUP (Hrsg.): *IBM Retail Data Warehouse General Information Manual*. IBM Corporation: IBM Industry Models and Assets Software Group, 2009. – `http://www-01.ibm.com/software/data/industry-models/library.html` zuletzt gesichtet: 28.10.2012

[IBM11] IBM INDUSTRY MODELS (Hrsg.): *IBM Telecommunications Data Warehouse General Information Manual*. IBM Corporation: IBM Industry Models, 2011. – `http://www-01.ibm.com/software/data/industry-models/library.html` zuletzt gesichtet: 28.10.2012

[ida11] *IBM DB2 Analytics Accelerator Studio Version 2.1 User's Guide*. : *IBM DB2 Analytics Accelerator Studio Version 2.1 User's Guide*. IBM Corporation, 2011. – Part Number: SH12-6960-00

[Inm96] INMON, William H.: *Building the Data Warehouse*. John Wiley & Sons, 1996. – ISBN 0764599445

[Kas80] KASS, G. V.: An Exploratory Technique for Investigating Large Quantities of Categorical Data. In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 29 (1980), Nr. 2, S. pp. 119–127. – ISSN 00359254

[KCPS12] KYNE, Frank ; CLITHEROW, David ; PIMISKERN, Udo ; SCHINDEL, Sim: *Redbook: GDPS Family An Introduction to Concepts and Capabilities*. IBM Corporation, 2012. – ISBN 0738436879

[KMU06] KEMPER, Hans-Georg ; MEHANNA, Walid ; UNGER, Carsten: *Business Intelligence - Grundlagen und praktische Anwendungen*. Friedr. Vieweg & Sohn Verlag/GWV Fachverlage GmbH, Wiesbaden, 2006. – ISBN 3–52805–802–1

[Koe12] KOEFFER, Sebastian: Mit Predictive Analytics in die Zukunft blicken. In: *Computerwoche* (2012), Oktober. – ISSN 0170–5121

[LK12] LEHMANN, Falk ; KUMMER, Axel: Predictive Analytics. In: *CFOworld* (2012). – `http://www.cfoworld.de/predictive-analytics?page=4` zuletzt gesichtet 23.10.2012

[LLC05] LEE, Chang-Hung ; LIN, Cheng-Ru ; CHEN, Ming-Syan: Sliding window filtering: an efficient method for incremental mining on a time-variant database. In: *Inf. Syst.* 30 (2005), Nr. 3, S. 227–244. – ISSN 0306–4379

[Man09] MANAGEMENT, IBM Software I.: *IBM Netezza Analytics*. IBM Corporation, 2009. – IMD14365-USEN-03

[MH07] MCCARTY, John A. ; HASTAK, Manoj: Segmentation approaches in data-mining: A comparison of RFM, CHAID, and logistic regression. In: *Journal of Business Research* 60 (2007), Nr. 6, S. 656–662. – ISSN 0148–2963

[Pes12] PESCHKE, Andreas (Hrsg.) ; IBM Corporation (Veranst.): *IBM DB2 Analytics Accelerator*. 2012 . – `http://www.dpc.de/fileadmin/templates/dpc/doc_pdf/APL_Tagung_2012/IBM_DB2_Analytics_Accelerator_IBM_Peschke.pdf` zuletzt gesichtet: 22.10.2012

[Rah94] RAHM, Erhard: *Mehrrechner-Datenbanksysteme - Grundlagen der verteilten und parallelen Datenbankverarbeitung*. Addison-Wesley, 1994. – ISBN 3–89319–702–8

[RM11] RESE, Joachim ; MAYER, Georg: *Rapid SAP NetWeaver BW ad-hoc Reporting Supported by IBM DB2 Analytics Accelerator for z/OS*. November 2011. – `http://www.sdn.sap.com/irj/scn/go/portal/prtroot/docs/library/uuid/0098ea1f-35fe-2e10-efa9-b4795c49389c?QuickLink=index&overridelayout=true&53051436042716` zuletzt gesichtet: 22.10.2012

[Spr10] SPRUTH, Wilhelm G.: System z and z/OS unique Characteristics / Wilhelm Schickard Instiute for Informatik, Tuebingen University. 2010. – Forschungsbericht. – ISSN 0946–3852

[sps11a] *IBM SPSS Modeler 14.2 In-Database Analytics Reference Guide*. : *IBM SPSS Modeler 14.2 In-Database Analytics Reference Guide*. IBM Corporation, 1994-2011. – Part Number: 00J2214-03 Rev. 2

[sps11b] *IBM SPSS Modeler 14.2 In-Database Mining Guide*. : *IBM SPSS Modeler 14.2 In-Database Mining Guide*. IBM Corporation, 2011. – http://www-01.ibm.com/support/docview.wss?uid=swg27022140 zuletzt gesichtet: 28.10.2012

[sps11c] *IBM SPSS Modeler 14.2 User's Guide*. : *IBM SPSS Modeler 14.2 User's Guide*. IBM Corporation, 2011. – http://www-01.ibm.com/support/docview.wss?uid=swg27022140 zuletzt gesichtet: 28.10.2012

[sps11d] *IBM SPSS Modeler Server 14.2 Administration and Performance Guide*. : *IBM SPSS Modeler Server 14.2 Administration and Performance Guide*. IBM Corporation, 2011. – http://www-01.ibm.com/support/docview.wss?uid=swg27022140 zuletzt gesichtet: 28.10.2012

[sps12a] *IBM SPSS In-Database Analytics Developer's Guide*. : *IBM SPSS In-Database Analytics Developer's Guide*. IBM Corporation, 2012. – Part Number: 00J2213-03 Rev. 2

[sps12b] *IBM SPSS Modeler 15 Applications Guide*. : *IBM SPSS Modeler 15 Applications Guide*. IBM Corporation, 2012. – http://www-01.ibm.com/support/docview.wss?uid=swg27023172 zuletzt gesichtet: 28.10.2012

[sps12c] *IBM SPSS Modeler 15 CLEF Developer's Guide*. : *IBM SPSS Modeler 15 CLEF Developer's Guide*. IBM Corporation, 2012. – http://www-01.ibm.com/support/docview.wss?uid=swg27023172 zuletzt gesichtet: 28.10.2012

[sps12d] *IBM SPSS Modeler 15 Modeling Nodes*. : *IBM SPSS Modeler 15 Modeling Nodes*. IBM Corporation, 2012. – http://www-01.ibm.com/support/docview.wss?uid=swg27023172 zuletzt gesichtet: 28.10.2012

[sps12e] *IBM SPSS Modeler 15 Source, Process and Output Nodes*. : *IBM SPSS Modeler 15 Source, Process and Output Nodes*. IBM Corporation, 2012. – http://www-01.ibm.com/support/docview.wss?uid=swg27023172 zuletzt gesichtet: 28.10.2012

[SR11] SPRUTH, Wilhelm G. ; ROSENSTIEL, Wolfgang: Revitalisierung der akademischen Großrechnerausbildung. In: *Informatik Spektrum* 34 (2011), Nr. 3, S. 295–303. – ISSN 0170–6012

[WKRQ+07] WU, Xindong ; KUMAR, Vipin ; ROSS QUINLAN, J. ; GHOSH, Joydeep ; YANG, Qiang ; MOTODA, Hiroshi ; MCLACHLAN,

Geoffrey J. ; NG, Angus ; LIU, Bing ; YU, Philip S. ; ZHOU, Zhi-Hua ; STEINBACH, Michael ; HAND, David J. ; STEINBERG, Dan: Top 10 algorithms in data mining. In: *Knowl. Inf. Syst.* 14 (2007), Nr. 1, S. 1–37. – ISSN 0219–1377

[zos01] *z/OS MVS JCL User's Guide*. : *z/OS MVS JCL User's Guide*. IBM Corporation, 2001. – Part Number: SA22-7598-01

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe, insbesondere sind wörtliche oder sinngemäße Zitate als solche gekennzeichnet. Mir ist bekannt, dass Zuwiderhandlung auch nachträglich zur Aberkennung des Abschlusses führen kann.

Ort Datum Unterschrift