

# Konzeption und Entwicklung eines Informationssystems zur Unterstützung der Vertriebsplanung im Lösungsvertrieb

Markus Paatsch  
Karlsruher Institut für Technologie  
markus.paatsch@student.kit.edu

**Zusammenfassung:** Dieser Artikel beschäftigt sich mit der Konzeption, Entwicklung und Evaluation einer softwaretechnischen Lösung zur Vertriebsplanungsoptimierung in lösungsorientierten B2B-Unternehmen. Die Grundlage der Optimierungen bilden hierbei historische Vertriebsdaten, wie z.B. frühere Vertriebschancen. Es wird untersucht, wie sich eine derartige Anwendung unter Zuhilfenahme eines bestehenden Business-Intelligence-Umfeld integrieren lässt. Die konzipierte Webapplikation wird als JSF-Anwendung umgesetzt. Es wird ein Klassifizierungsverfahren implementiert, welches durch die Bibliothek weka integriert wird. Zur Optimierung werden Funktionen der Bibliothek ilog cplex benutzt. Hauptaufgabe der Anwendung ist die Maximierung des Umsatzes durch Optimierung der Zuweisungen von Vertriebsmitarbeitern zu Kunden. Alle optimierten Ergebnisse werden in übersichtlicher und anschaulicher Art visualisiert.

## 1 Einführung

Der einmalige Verkauf von Produkten verliert im Business-to-Business-Umfeld zunehmend an Bedeutung. Stattdessen konzentrieren sich immer mehr B2B-Unternehmen darauf, durch das Anbieten von komplexen, integrierten Lösungen langfristige Beziehungen mit Kunden zu erschaffen, um so einen größeren und kontinuierlicheren Gewinn zu erwirtschaften. Der lösungsorientierte Verkauf existiert so nur im B2B-Umfeld. Eine Lösung ist aus verschiedenen Produkten und Dienstleistungen zusammengesetzt [JKS03]. Für diese Arbeit ist nur das grundlegende Konzept von Lösungen wichtig, die spätere Lösungsdomäne ist irrelevant. Verkäufer bzw. Repräsentanten des Unternehmens spielen eine zunehmend wichtigere Rolle beim Aufbau von Kundenverbindungen. Durch die Zusammenarbeit von Verkäufern, bzw. deren gezielte Zuweisung zu bestimmten Kunden kann eine Gewinnmaximierung erzielt werden [KBB13]. Innerhalb dieser Teams existieren oft verschiedene Vertriebsrollen. Die Interaktionen zwischen Kunden und Verkäufern können durch die rasche IT-Entwicklung der letzten Jahrzehnte detailliert gemessen und in großem Umfang gespeichert werden. Diese Daten werden im Folgenden Kundeninteraktionsdaten genannt. Sie beschreiben auf verschiedenen Ebenen die Güte der Interaktion zwischen Kunde und Verkäufer [Ha12]. Mit Hilfe dieser historischen Daten ist es möglich, die Vertriebsplanung zu optimieren und somit den Ertrag eines Unternehmens entscheidend zu maximieren. Da dem Verkaufsprozess von komplexen, integrierten Lösungen bisher

relativ wenig Beachtung geschenkt wurde, existieren aktuell keine Softwaretools, die es ermöglichen, Verkaufsteams im B2B-Umfeld optimal bzw. gewinnbringend Kunden – auf Basis von Kundeninteraktionsdaten – zuzuweisen.

Das Ziel dieser Arbeit ist die softwaretechnische Planung, Umsetzung und Evaluation einer Webapplikation zur Vertriebsplanungsoptimierung. Die Webapplikation dient zur Unterstützung in der Vertriebsplanung und soll in ein Business-Intelligence-Umfeld integriert werden. Diese Arbeit beschäftigt sich dabei ausschließlich mit der softwaretechnischen Umsetzbarkeit der verschiedenen Komponenten, ihrer Interaktion untereinander und der anschließenden Integration des Systems in ein bestehendes Business-Intelligence-Umfeld.

Als Daten-Grundbausteine dienen *Gebiete*, mit darin enthaltenen *Kunden* und *Verkäufern* eines Unternehmens. Besteht eine *Verkaufsmöglichkeit* mit einem Kunden, muss diesem ein Verkäufer zugewiesen werden. Der Nutzer des Systems soll in der Lage sein, diese *Zuweisungen* zu ändern. Eine *Vertriebsplanung* für ein Gebiet ist komplett, wenn jedem Kunden dieses Gebiets ein Verkäufer zugeordnet wurde. Die Hauptaufgabe der Webapplikation besteht darin, unvollständige Vertriebsplanungen mittels Data-Mining-Techniken und linearer Programmierung optimal zu ergänzen bzw. das Optimum zu berechnen. Alle Ergebnisse werden visualisiert und können untereinander verglichen werden (dazu gehört auch der Vergleich mit dem Optimum).

Im weiteren Verlauf werden zuerst verwandte Arbeiten vorgestellt. Anschließend werden verschiedene Arbeitsschritte der Anwendung aufgezeigt. Die Anforderungsanalyse bildet die Grundlage für die nachfolgenden Abschnitte Design und Architektur sowie Implementierung. Abschließend werden die umgesetzten Anforderungen evaluiert und ein Fazit gezogen bzw. ein Ausblick gegeben.

## 2 Verwandte Arbeiten

In [KBF14] wurde eine umfangreiche Literaturrecherche zu Themen wie Sales Force Modelling, Sales Force Analytics, Sales Force Optimization, etc. durchgeführt. Diese Literaturrecherche liefert daher auch die ursprüngliche Grundlage für die in diesem Paper beschriebene Implementierung. Vorhergehende Arbeiten auf ähnlichem Themengebiet aus [KBF14] sind vor allem [Lo76], [SA98], [DA99] und [Ba12]. Daraus resultiert z.B. das in der Einführung angeführte Fehlen eines derartigen Tools zur Zuweisung von Verkaufsteams im B2B-Umfeld. Grundlage für dieses Paper bildet die Arbeit von Johannes Kunze von Bischoffshausen und Jeffrey T. Becker [KBB13], welche das zugrundeliegende Optimierungsmodell der Anwendung liefert. Die verwendete Datengrundlage durch Kundeninteraktionen basiert auf der Arbeit von Francois Habryn [Ha12].

### 3 Methodik

Im Folgenden werden die Schritte vorgestellt, die für das System notwendig sind, um optimale Ergebnisse oder Ergänzungen mit diesem Optimierungsmodells zu berechnen. Das Vorgehen basiert grundlegend auf dem Modell von [DH07]: Der Datenaufbereitung folgt eine prädiktive Analyse (Vorhersage), um mit Hilfe dieser optimale Ergebnisse zu berechnen.

#### 3.1 Daten

Die Webapplikation bezieht ihre Daten aus einem Data Warehouse, in dem bereits die Datenaufbereitung stattfand. Der Datenprozess beschränkt sich für das System daher nur auf die Datenbeschaffung, also die Kommunikation mit dem Data Warehouse, und die Transformation in das eigene Format. Die bezogenen Daten bieten verschiedene aggregierte Metriken für bisherige Verkaufsmöglichkeiten.

#### 3.2 Prädiktive Analyse

Die bezogenen Daten repräsentieren Verkaufsmöglichkeiten aus der Vergangenheit. Aufgrund dieser historischen Daten kann für eine bestehende Käufer-Verkäufer-Beziehung der Verkaufserfolg vorhergesagt werden. Dies geschieht bei der prädiktiven Analyse beispielhaft mit dem Verfahren der logistischen Regression, welches ein statistisches Verfahren zur Regressionsanalyse darstellt. Dabei wird anhand einer Funktion der Zusammenhang einer abhängigen und mehrerer unabhängiger Variablen modelliert. Damit das Modell Wahrscheinlichkeiten im Bereich  $p \in [0,1]$  liefert, muss die abhängige Variable binär sein. Der jeweilige Ausgang (verkauft, nicht verkauft) von vergangenen Verkaufsmöglichkeiten wird somit zur abhängigen Variable und alle gesammelten Daten zur vorherigen Interaktion werden als unabhängige Variablen modelliert.

#### 3.3 Optimierung

Ausgehend von den vorhergesagten Verkaufswahrscheinlichkeiten sollen optimale Zuweisungen bzw. optimale Ergänzungen zu unvollständigen Vertriebsplanungen berechnet werden. Dazu wird das Optimierungsmodell aus [KBB13] benutzt:

$$\begin{aligned} & \max \sum_{j=1}^n \left( v_j \prod_{i=1}^m q_{ij}^{x_{ij}} \right) \\ \text{s. t. } & \sum_{j=1}^m (x_{ij}) \leq W_i \quad (i = 1, \dots, m) \\ & \sum_{i=1}^n (x_{ij}) = 1 \quad (j = 1, \dots, n) \end{aligned}$$

Dabei steht  $i$  für Verkäufer und  $j$  für Kunde.  $x_{ij}$  ist eine binäre Variable: 1, wenn Verkäufer  $i$  Kunde  $j$  zugewiesen ist (0, sonst).  $W_i$  ist die maximale Anzahl an Kunden, die einem Verkäufer zugewiesen werden können. Dieser Parameter kann manuell geändert werden.  $v_j$  bezeichnet den möglichen Ertrag bei Verkauf an Kunde  $j$ . Die Wahrscheinlichkeit, mit welcher ein Verkäufer  $i$  bei Kunde  $j$  Erfolg hat, heißt  $q_{ij}$ .

## 4 Anforderungsanalyse

Ein Benutzer soll die vorgestellte Methodik anhand von verschiedenen Interaktionsmöglichkeiten steuern können. Diese werden nachfolgend durch die funktionalen Anforderungen spezifiziert. Die Rahmenbedingungen der funktionalen Anforderungen werden durch die nichtfunktionalen Anforderungen beschrieben.

### 4.1 Funktionale Anforderungen

- **FA1** *Anfordern einer Datenauswahl*: Der Nutzer wählt die Gebiete für die Optimierung.
- **FA2** *Verändern der Optimierungsparameter*: Verschiedene Optimierungsparameter (maximale Anzahl von Kunden pro Verkäufer, etc.) können manuell geändert werden.
- **FA3** *Ändern des Gebiets in dem Zuweisungen optimiert werden*: Der Nutzer kann in ein anderes Gebiet wechseln.
- **FA4** *Verkäufer für die Optimierung im aktuellen Gebiet an- bzw. abwählen*: Im aktuellen Gebiet können Verkäufer in die Berechnung integriert oder von dieser ausgeschlossen werden.
- **FA5** *Manuelles Anpassen der Zuweisungen zwischen Verkäufern und Kunden*: Für eine Berechnung können feste Zuweisungen gesetzt werden.
- **FA6** *Berechnung optimaler (ergänzender) Zuweisungen im aktuellen Gebiet*: Fehlende Zuweisungen zwischen Kunden und Verkäufer sollen automatisch von der Anwendung ergänzt werden.
- **FA7** *Vergleich verschiedener Berechnungen*: Verschiedene Berechnungen können miteinander oder dem Optimum verglichen werden.

### 4.2 Nichtfunktionale Anforderungen

- **NA1** *Zugriffsanforderungen*: Der Zugriff auf die Webapplikation erfolgt ausschließlich per Webbrowser. Dazu muss der Server bzw. die URL der Anwendung bekannt und erreichbar sein.
- **NA2** *Nutzungsanforderungen*: Alle funktionalen Anforderungen sollen in maximal 3 Sekunden bearbeitet und ausgeführt werden.

- **NA3 Bedienungsanforderungen:** Sobald man Zugriff auf die Webapplikation besitzt, kann diese ausschließlich mit der Maus bedient werden. Durch die Integration von Tastaturbefehlen wird die Bedienung erleichtert.
- **NA4 Fehlerbehandlung:** Jede unerlaubte Operation des Nutzers bzw. jeder durch das System erkannte Fehler wird dem Nutzer in angemessener Art und Weise angezeigt.

Durch die Definition von Anforderungen an die Anwendung kann das System anschließend näher spezifiziert werden.

## 5 Design und Architektur

Im folgenden Abschnitt werden verschiedene Anforderungen technisch näher spezifiziert. Die Grundlage für die Geschäftslogik bilden die funktionalen Anforderungen aus Abschnitt 4.1. Exemplarisch wird pro Schritt (Abschnitt 3) eine Anforderung betrachtet. Abschließend wird die grundlegende Architektur des Systems erläutert.

*Spezifikation - Datenauswahl anfordern* (Abschnitt 3.1): Alle Funktionen des Systems können nur auf einer vorhandenen Datenbasis ausgeführt werden. Aus diesem Grund wird beim Aufruf der Webapplikation zuerst eine Daten- bzw. Gebietsauswahl vom Nutzer getroffen. Die Daten sind in einem Data Warehouse gespeichert und können mit Hilfe der Vorauswahl abgefragt werden. Die bezogenen Daten müssen anschließend in ein eigenes Format überführt werden.

*Spezifikation - Zuweisungen anpassen* (Abschnitt 3.2): Ein Verkäufer kann (einstellbar vielen) Kunden – jedem Kunden aber nur einmal - zugeordnet werden. Wird die maximale Anzahl an Kunden pro Verkäufer überschritten, informiert die Anwendung den Nutzer per Fehlerinformation und bricht die Operation ab. Beim eigentlichen Zuweisen von Verkäufern zu Kunden wird gleichzeitig die Verkaufswahrscheinlichkeit aufgrund der vorhandenen Daten berechnet (prädiktive Analyse).

*Spezifikation - Berechnung optimaler Zuweisungen* (Abschnitt 3.3): Die Hauptaufgabe der Anwendung liegt darin, den erwarteten Umsatz von Vertriebsplanungskonfigurationen zu berechnen. Diese können in zwei Arten eingeteilt werden. Bei vollständig zugewiesenen Konfigurationen kann nichts optimiert werden. Der Nutzer hat bereits jedem Kunden einen Verkäufer zugeordnet. Die Anwendung hat in diesem Fall die Aufgabe, verschiedene Kenngrößen wie den erwarteten Umsatz zu berechnen. Wenn der Nutzer allerdings nicht alle Zuweisungen manuell setzt, wird die Anwendung diese Aufgabe übernehmen bzw. optimieren (unvollständige Konfiguration). Es muss sichergestellt sein, dass ausreichend berechnungsrelevante Verkäufer vorhanden sind.

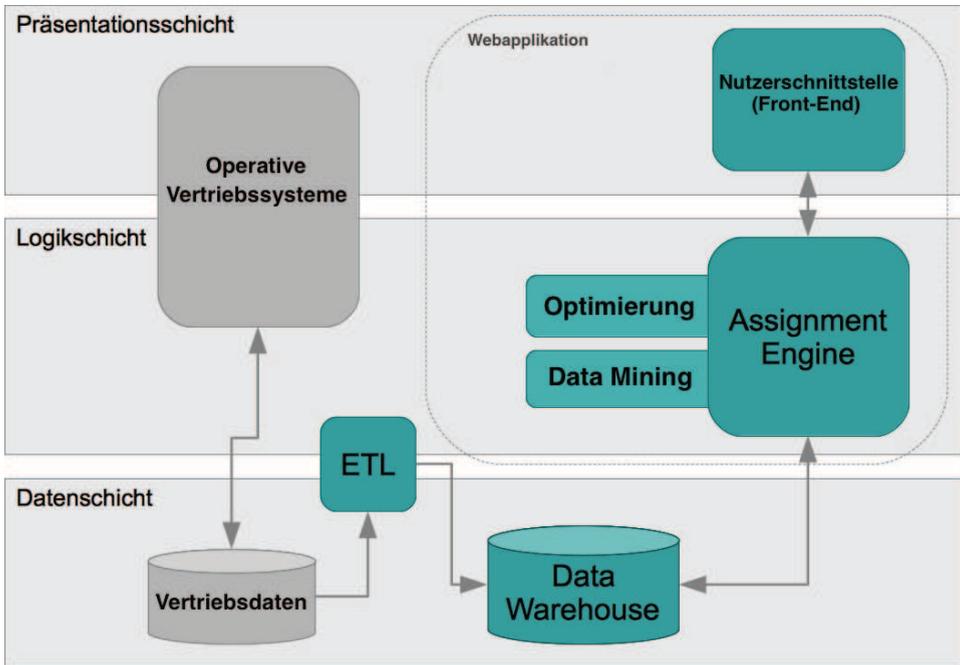


Abbildung 1: Systemarchitektur

Abbildung 1 beschreibt die Einbettung der Anwendung in die Gesamt- bzw. Systemarchitektur. Das komplette System ist in drei Schichten untergliedert (Daten-, Logik- und Präsentationsschicht), wobei die Webapplikation nur auf Logik- und Präsentationsschicht arbeitet. Daten werden aus einem bestehenden Data Warehouse bezogen, welches Schnittstellen zum Datentransfer anbietet. Die Datenschicht ist daher nicht Teil dieser Arbeit.

Die Anwendungsarchitektur in Abbildung 2 ist ein detailliert dargestellter Teilbereich der Systemarchitektur. Sie bezieht sich nur auf die Realisierung des Webapplikationsteils aus Abbildung 1. Da die Anwendung auf einem zentralen Server gehostet wird und jeder Nutzer nur per Webbrowser darauf zugreifen soll, bietet sich hier eine einfache Client-Server-Architektur an, welche wiederum als MVC-Pattern [Gal1] realisiert wird. Zusätzlich zur Grundarchitektur sind in der Abbildung auch die Datenströme der Anwendung integriert.

Mit den Anforderungsspezifikationen und dem Architekturentwurf kann das System implementiert werden.

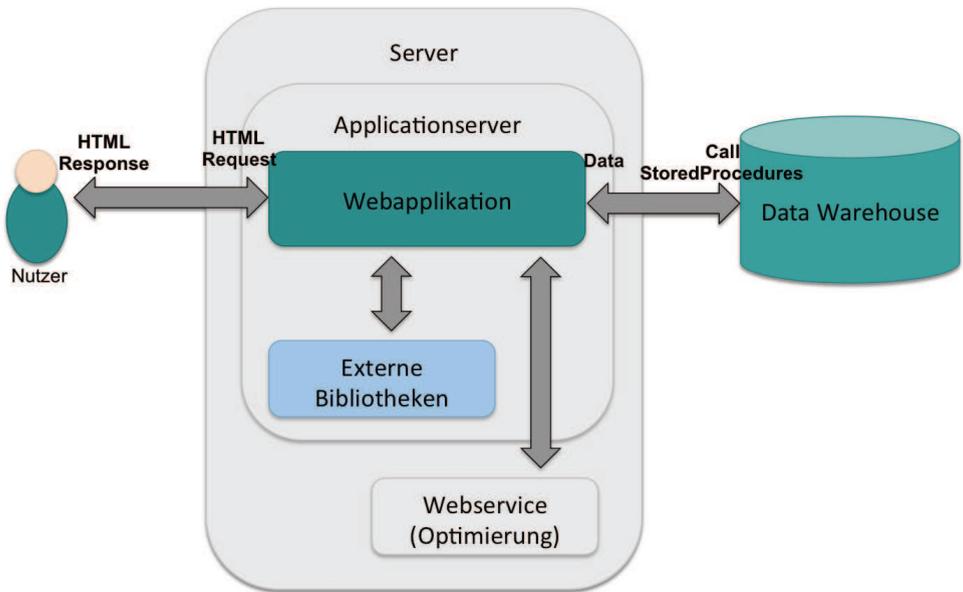


Abbildung 2: Anwendungsarchitektur

## 6 Implementierung

Die Webapplikation wird als JSF-Anwendung (JavaEE 7, [Ja14]) umgesetzt. Zusätzlich werden verschiedene Frameworks wie Twitter's Bootstrap [Bo14], jQuery [Jq14] sowie D3.js [D314] zur Graph-Visualisierung auf Clientseite eingesetzt. Durch das *Model-View-Controller Pattern* [Ga11] ist eine gewisse Paketstruktur vorgegeben, welche allerdings noch um Services und Parser erweitert wurde. Demnach befindet sich jeglicher HTML/Javascript-Content im View-Bereich. Das *Model* beinhaltet die interne Datenstruktur der Webapplikation. Der *Controller*, hier als Mediator umgesetzt, steuert die gesamte Kommunikation zwischen Front- und Back-End sowie mit dem Data Warehouse.

### 6.1 Model

Damit die Daten aus dem Data Warehouse verarbeitet werden können, müssen diese in eine entsprechende Datenstruktur überführt werden. Diese Datenstruktur ist im Model-Paket verankert. Der Mediator verwaltet ein *Map*-Objekt und hält damit alle Gebiete bzw. alle angeforderten Daten. Bei Instanziierung erhält ein Gebiet eine Standardkonfiguration ohne jegliche Zuweisungen. Alle aktuellen Änderungen von Nutzerseite werden innerhalb einer aktuellen Konfiguration abgelegt. Sobald eine Berechnung erfolgreich war, wird diese an eine Liste von allen Berechnungen angehängt. Diese Berechnungen hängen immer am aktuellen Konfigurationsobjekt,

welches alle für die Berechnung verwendeten Verkäufer und Kenngrößen, wie erwarteter Umsatz, speichert. Die innerhalb einer Konfiguration integrierten Verkäufer-Objekte verwalten ihre eigenen Zuweisungen wobei ein Zuweisungsobjekt den jeweiligen Kunden und die Verkaufswahrscheinlichkeit enthält. Außerdem verwaltet jedes Verkäuferobjekt seine eigenen vergangenen Verkaufsmöglichkeiten. Diese sind entscheidend für die logistische Regression, da sie neben dem Kunden alle Metriken und das Resultat der damaligen Verkaufsmöglichkeit speichern.

## 6.2 Services

Services sind als modulare Bausteine der Anwendung gedacht. Ein schneller Austausch des Optimierungsverfahrens oder des Klassifikators wird somit vereinfacht. Das Services-Paket beinhaltet folgende Funktionen:

- Kommunikation mit dem Data Warehouse – Daten (Abschnitt 3.1)
- Prädiktive Analyse (Abschnitt 3.2)
- Optimierung von Konfigurationen (Abschnitt 3.3)

Die Klasse *DataService* ist dafür zuständig, mit dem Data Warehouse zu kommunizieren. Dabei geht es nur um die Etablierung einer Verbindung und die Entgegennahme von Daten. Um die Rohdaten in die Model-Datenstruktur zu übersetzen, wird der *DataParser* aus dem Parser-Paket beansprucht. Letztendlich liefert der *DataService* ein komplettes Map-Objekt, gefüllt mit den angewählten Daten.

Die ursprünglich vorgestellten Techniken zum Data Mining und zur Optimierung werden nicht eigenhändig in den Klassen *ClassificationService* und *OptimizationService* implementiert. Vielmehr sind auch diese beiden Klassen Schnittstellen zu externen Bibliotheken. Eigene Daten müssen lediglich in das externe Format übersetzt werden. Im Rahmen dieser Arbeit wird für die logistische Regression das Open-Source-Framework *weka* [We14] und für die binäre ganzzahlige Optimierung das von IBM entwickelte *ilog cplex* [IC14] benutzt. Der Optimierungsansatz wurde in Abschnitt 3.3 erläutert.

## 6.3 Controller

Das Paket *Controller* bildet die Controller-Komponente des MVC-Patterns. Die einzige darin enthaltene Klasse *Mediator* (Vermittler-Pattern, [Ga11]) steuert die gesamte Kommunikation zwischen Front- und Back-End und zwischen der Webapplikation und dem Data Warehouse. Damit der Nutzer Funktionen des Back-Ends ansprechen kann, ist die Klasse *Mediator* direkt mit dem Front-End verbunden (Callback-Funktionen). Um die Nutzeranweisungen weitergeben zu können, verwaltet der *Mediator* ein komplettes Model-Objekt (gefüllt mit den vom Nutzer angeforderten Daten).

## 6.4 View

Als Nutzerschnittstelle bzw. View (MVC, [Ga11]) wird im aktuellen Kontext jeglicher HTML-/CSS-/Javascript-Content bezeichnet. Die Datenauswahl kann den Eintrittspunkt in die Anwendung darstellen. Hier sollen dem Nutzer alle vorhanden, auswählbaren Gebiete aufgelistet werden. Diese initiale Gebietsliste wird beim Seitenaufruf aus dem Data Warehouse abgefragt. Erst wenn eine nichtleere Auswahl an Gebieten getroffen wurde, ist es möglich, auf die Übersichtsseite zu navigieren. Die Übersichtsseite ist in Abbildung 3 exemplarisch mit 4 Gebieten dargestellt. Verbindungen zwischen Kunde (engl. customer) und Verkäufer bzw. Repräsentant (engl. representative) symbolisieren in dieser Ansicht (*Sales Collaboration History*) vergangene Verkaufsmöglichkeiten und deren Kardinalität. Der Nutzer hat hier die Möglichkeit, sich Daten detaillierter anzeigen zu lassen (*Additional Information*), oder den Datenbestand zu filtern. Mit einem Klick auf *What-If Analysis* kann der Nutzer Zuweisungen für ein Gebiet verändern. Im nächsten Reiter können die so veränderten Konfigurationen optimiert und verglichen werden. In allen Ebenen der Webapplikation werden (Zwischen-) Ergebnisse angemessen visualisiert.

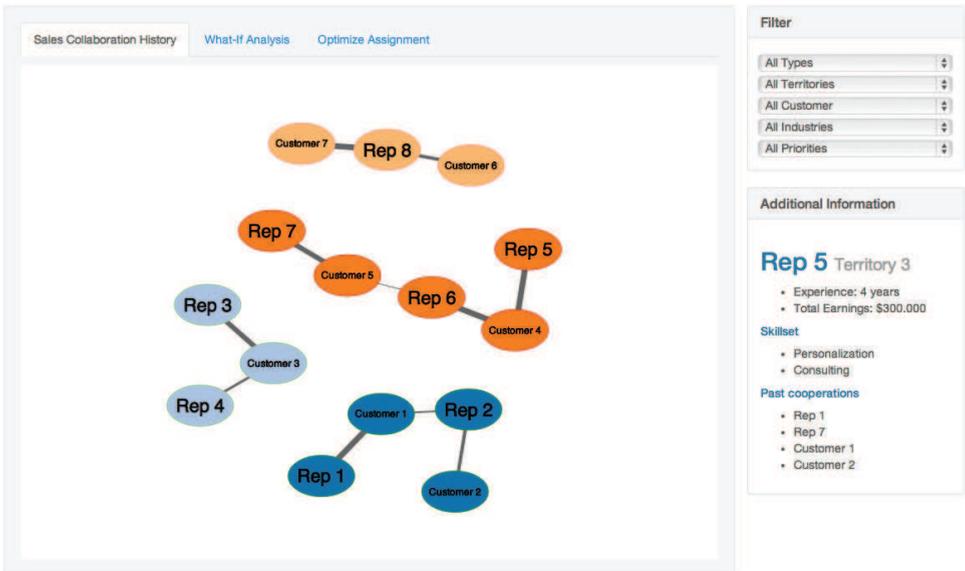


Abbildung 3: Ausschnitt der Oberfläche (Zuweisungen in Gebieten)

## 7 Evaluation

Nachfolgend wird untersucht, ob die entwickelte Webapplikation den spezifizierten Anforderungen aus Kapitel 4 gerecht wird.

Die einzelnen Klassen bzw. Module/Pakete des Back-Ends können mit Hilfe von Unit Tests automatisiert auf korrekte Funktionsausführung überprüft werden. Dafür wurde für jede Klasse des Back-Ends ein Unit-Test geschrieben. Unit-Tests prüfen jede implementierte Funktionalität mit vorgefertigten Ergebnissen. Bei Änderungen am Quelltext kann so leicht erkannt werden ob die Funktionalität verletzt wurde. Für das Back-End der Webapplikation wurden alle Unit-Tests mittels JUnit entwickelt. Alle Unit-Tests wurden bestanden. Nachdem jedes Modul isoliert auf seine grundlegende Funktionalität geprüft wurde, muss anschließend durch Integrationstests das richtige Zusammenspiel aller Module untersucht werden. Mit Hilfe von Test-Daten und dem Debug-Werkzeug der Entwicklungsumgebung wurde sichergestellt, dass jeder Testprozess korrekt abläuft. Nachfolgend wird exemplarisch ein solches Testszenario vorgestellt:

**Berechnung nicht-vollständiger Konfiguration:** Die Datenbeschaffung war erfolgreich und der Nutzer kann mit der Manipulation der Zuweisungen beginnen. Zunächst aktiviert er eine Teilmenge der Verkäufer für die Berechnung. Durch diese Funktionalität ist die Verbindung zwischen Controller und Model sichergestellt. Die What-If-Seite zeigt stets nur das an, was tatsächlich als Model im Back-End liegt. Anschließend setzt der Nutzer Zuweisungen in der Form, dass mindestens ein Kunde keinen Verkäufer zugewiesen bekommt. Unter der Annahme, dass so mindestens eine Zuweisung existiert, kann nun überprüft werden, ob die zugehörige, berechnete Verkaufswahrscheinlichkeit korrekt ist. Dies wurde mit verschiedenen Daten manuell überprüft. Das Beispiel läuft korrekt ab, also funktioniert auch die Verbindung zwischen Controller und ClassificationService. Danach kann die Berechnung gestartet werden. Auch hier wurden Beispiele manuell mit den Ergebnissen verglichen, um die Optimalität der hinzukommenden Zuweisungen zu überprüfen.

## 8 Fazit und Ausblick

In dieser Arbeit sollte eine Webapplikation zur Vertriebsplanungsoptimierung – basierend auf Kundenbeziehungen – softwaretechnisch geplant, umgesetzt und evaluiert werden. Durch die Entwicklung des Systems sollte grundlegend geprüft werden, ob eine solche Anwendung in ein bestehendes Business-Intelligence-Umfeld integriert werden kann. Durch die erfolgreiche Umsetzung der Webapplikation können mehrere Fragen beantwortet werden. Zum einen wurde gezeigt, dass die geforderten Funktionen (Kapitel 4) und Komponenten mit vorhandener Technologie implementiert werden können. Das bedeutet weiter, dass eine derartige Anwendung in ein bestehendes Business-Intelligence-System integriert werden kann. Zum anderen ist der Nutzen für Unternehmen mindestens genauso wichtig wie die Integration. Da das System alle Anforderungen umsetzt, kann aus bisher wenig beachteten Daten automatisiert neues

Wissen über eine optimierte Vertriebsplanung gewonnen werden. Trotz der erfolgreichen prototypischen Umsetzung kann ein derartiges System kaum mit vorhandener Software verglichen werden. Es existieren bisher keine vergleichbaren Systeme die im B2B-Umfeld auf Kundeninteraktionsdaten Vertriebsplanungsoptimierungen für Verkaufsteams durchführen.

Nachfolgend werden einige Ideen aufgezeigt, wie eine derartige Anwendung weiterentwickelt werden könnte.

- Verwerfung des Gebietsparadigma – Ein globales Gebiet
- Einführung zusätzlicher Bedingungen in der Optimierung
- (Echte) Serviceorientierte Architektur
- Realisierung für mobile Geräte
- Manuelle Erstellung einer Datenbasis zu Simulations-/Forschungszwecken
- Verbesserung des Optimierungsmodells
- Austausch des Prädiktionsalgorithmus

Für den Nutzen des noch jungen Gebiets der Customer Intimacy Analytics bzw. der Sales Force Analytics/Optimization bietet das entwickelte System einen ersten demonstrativen Grundstein und die Möglichkeit für weitere Forschungen.

## Danksagung

Dieses Projekt wurde im Rahmen meiner Bachelorarbeit unter Betreuung von Johannes Kunze von Bischhoffshausen umgesetzt und seitdem weiterentwickelt. Ihm gilt ein besonderer Dank, da er mich nicht nur bei der Abschlussarbeit, sondern auch beim Verfassen dieses Papers tatkräftig unterstützt hat.

## Literaturverzeichnis

- [Ba12] Baier, M., Carballo, J. E., Chang, A. J., Lu, Y., Mojsilovic, A., Richard, M. J., Singh, M., Squillante, M. S., Varshney, K. R.: Sales-force performance analytics and optimization. In IBM Journal of Research and Development (56:6), S. 8:1-8:10, 2012
- [Bo14] Bootstrap, 2014, <http://www.getbootstrap.com>, zuletzt geprüft am 19.06.2014
- [D314] D3.js, 2014, <http://www.d3js.org>, zuletzt geprüft am 19.06.2014
- [DH07] Davenport, T. H., Harris, J.G.: Competing on analytics: The new science of winning. Harvard Business School Press, Cambridge, 2007
- [DH99] Drexl, A., Haase, K.: Fast approximation methods for sales force deployment. In Management Science (45:10), S. 1307-1323, 1999

- [Ga11] Gamma, E.: Entwurfsmuster: Elemente wiederverwendbarer objektorientierter Software. Addison-Wesley, München, 2011
- [Ha12] Habryn, F.: Customer intimacy analytics: Leveraging operational data to assess customer knowledge and relationships and to measure their business impact. Dissertation, Karlsruhe, 2012
- [IC14] IBM Ilog Cplex, 2014, <http://www-03.ibm.com/software/products/de/ibmilogcpleoptistud>, zuletzt geprüft am 19.06.2014
- [Ja14] Java EE 7, 2014, <http://www.oracle.com/technetwork/java/javaee/overview/index.html>, zuletzt geprüft am 19.06.2014
- [JKS03] Johansson, J. E., Krishnamurthy, C., Schliessberg, H.E.: Solving the solutions problem. In McKinsey Quarterly (3), S. 116-125, 2003
- [Jq14] jQuery, 2014, <http://jquery.com/>, zuletzt geprüft am 19.06.2014
- [KBB13] Kunze von Bischoffshausen, J., Becker, J.: Towards a Personal Relationship-Based Assignment of Client Representatives to Accounts. In IESS, S. 329-335, 2013
- [KBF14] Kunze von Bischoffshausen, J., Fromm, H.: Sales Force Analytics for the Solution Selling Firm: A Predictive Model for Assessing the Impact of Sales Team Assignments. In AMICS, 2014, i.E.
- [Lo76] Lodish, L.M.: Assigning salesmen to accounts to maximize profit. In Journal of Market Research (13:4), S. 440-444, 1976
- [SA98] Skiera, B., Albers, S.: COSTA: Contribution Optimizing Sales Territory Alignment. In Marketing Science (17:3), S. 196-213, 1998
- [We14] Weka, 2014, <http://www.cs.waikato.ac.nz/ml/weka/>, zuletzt geprüft am 19.06.2014