

Master Thesis

im Rahmen des
Universitätslehrganges „Geographical Information Science & Systems“
(UNIGIS MSc) am Interfakultären Fachbereich für GeoInformatik (Z_GIS)
der Paris Lodron-Universität Salzburg

zum Thema

„geoCRM“ Business GIS im Kundenbeziehungsmanagement

vorgelegt von

Paul Jereb, BSc
U1506, UNIGIS MSc Jahrgang 2010

Zur Erlangung des Grades
„Master of Science (Geographical Information Science & Systems) – MSc (GIS)“

Gutachter:
Ao. Univ. Prof. Dr. Josef Strobl

Graz, 16.03.2014

Erklärung

Ich versichere, diese Master Thesis ohne fremde Hilfe und ohne Verwendung anderer als der angeführten Quellen angefertigt zu haben, und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat. Alle Ausführungen der Arbeit die wörtlich oder sinngemäß übernommen wurden sind entsprechend gekennzeichnet.

Graz, 16. März 2014

Paul Jereb

Zusammenfassung

Die vorliegende Arbeit untersucht, ob und wie die Entwicklungen der letzten Jahre in den Bereichen der GIS-Technologien im Umfeld betrieblicher Informationssysteme, mit speziellem Fokus auf das Kundenbeziehungsmanagement, genutzt werden können. Diese Untersuchung erfolgt in zwei Schritten: durch die Umsetzung als Prototyp für ein geoCRM-System, um das Vorhandensein der technologischen Rahmenbedingungen zu überprüfen und durch die Befragung von Experten, um festzustellen in welcher Form in das CRM-System integrierte GIS-Technologien Verkaufs- und Entscheidungsfindungsprozesse optimieren.

In Kapitel 1 wird zunächst in das Thema eingeleitet, die Motivation dafür beschrieben und werden verwendete Begriffe eingeführt. Nach einem Überblick über den Stand der Forschung wird auf die Ziele der Arbeit und die zu deren Erreichung angewandte Methodik eingegangen. Im Kapitel 2 erfolgt eine Einführung in die notwendigen theoretischen und technologischen Grundlagen. In Kapitel 3 wird ein Anforderungsprofil an ein geoCRM System skizziert. Das Kapitel 4 gibt einen Überblick über Möglichkeiten die eine Integration von GIS-Technologien in CRM-Systeme eröffnet. Dabei werden aus den Perspektiven von Business Mapping, Location Intelligence, Geomarketing und der mobilen Nutzung potentielle Anwendungsgebiete entwickelt. Im Kapitel 5 wird der Prototyp eines räumlich expliziten CRM-Systems umgesetzt. Dabei wird auf das für die Integration verwendete Architekturmodell eingegangen, die Integration in die Datenbank sowie die Geokodierung der Adressdatensätze beschrieben. Nach der Entwicklung einer darauf aufbauenden Web Mapping Komponente, wird auf die Integration in die bestehende CRM-Applikation eingegangen. Das Kapitel 6 fasst die Ergebnisse der Prototypentwicklung und der Expertenbefragung zusammen, evaluiert diese und gibt einen Ausblick auf weitere Forschungsaufgaben in diesem Gebiet. Des Weiteren werden die gewonnenen Erkenntnisse im Hinblick auf die Literatur diskutiert und ein Fazit gezogen.

Abstract

This master thesis outlines, if and how the recent evolution of GIS technologies can be used in the area of enterprise information systems, with special (and spatial) focus on customer relationship management. The research is carried out in two steps: to evaluate the existence of the technical requirements, a geoCRM prototype is developed and to analyze the impact on sales- and decision making processes, experts are consulted.

In chapter 1 the topic of the thesis is introduced, the motivation of the author stated followed by definitions of the used terms. After giving an overview of the state of research, the goals of the thesis are outlined and the methodology used to reach them explained. Chapter 2 introduces the most important theoretical and technological principles. In chapter 3 a requirement profile for a geoCRM system is drafted. Chapter 4 gives an overview of the possibilities “geoenhancing” a CRM system opens up. Through the perspective of business mapping, location intelligence, geomarketing and mobile usage potential areas of application are developed. The following chapter 5 discusses the implementation of a “geoenhanced” CRM prototype. During this chapter the used software architecture is presented, the spatial database enhancements explained and the geocoding of the existing address datasets discussed. After describing the implemented web mapping component, the integration in the existing CRM application is outlined. Finally chapter 6 summarizes the results of the prototype implementation and the expert consultations, evaluates them and gives an outlook on possible research in this area. Last but not least, the findings of the thesis are discussed with regard to the literature and a conclusion drawn.

Inhaltsverzeichnis

Zusammenfassung.....	2
Abstract	3
Inhaltsverzeichnis.....	4
Abbildungsverzeichnis	7
Tabellenverzeichnis.....	9
Listingverzeichnis	10
Abkürzungsverzeichnis	11
1 Einleitung	13
1.1 Motivation	13
1.2 Begriffseinführung	15
1.3 Literaturüberblick.....	20
1.4 Bestehende Ansätze.....	23
1.5 Ziele.....	27
1.6 Methodik	27
1.7 Rahmenbedingungen	28
1.8 Struktur der Arbeit.....	29
1.9 Abgrenzung	31
2 Grundlagen.....	32
2.1 Geodatenbanksysteme	32
2.1.1 Geometrische Datentypen	33
2.1.2 Räumliche Operatoren	36
2.1.3 Räumliche Funktionen	38
2.2 Web Services	39

2.2.1	SOAP	41
2.2.2	REST	43
2.2.3	Web Map Service	44
2.2.4	Web Feature Service	45
2.3	Georeferenzierung	47
2.3.1	Geokodierung – von der Adresse zur Koordinate	48
2.3.2	Geokodierungsdienste	53
2.4	Web Mapping	56
2.4.1	Architekturmodelle	57
2.4.2	Clientseitiges Web Mapping	57
2.4.3	Serverseitiges Web Mapping	60
3	Anforderungsprofil geoCRM	62
4	Anwendungsgebiete geoCRM	64
4.1	Business Mapping	65
4.2	Location Intelligence	66
4.3	Mobile	67
5	Prototyp	71
5.1	Architekturmodell	71
5.2	Integration in die Datenbank	72
5.2.1	Analyse des Datenmodells	72
5.2.2	Erweiterung des Datenmodells	73
5.2.3	Analyse der Adressdatensätze	74
5.2.4	Geokodierung der Adressdatensätze	76
5.2.5	Aufbereitung der Daten für das Web Mapping	77

5.3	Web Mapping	80
5.4	Integration in die Applikation	84
6	Zusammenschau und Ausblick	91
7	Quellenverzeichnis	95
	Anhang A	101
	Anhang B	104
	Anhang C	105
	Anhang D	108
	Anhang E.....	112

Abbildungsverzeichnis

Abbildung 1: Klassischer Marketingmix ergänzt um die räumliche Sicht	18
Abbildung 2: Begriffseinordnung geoCRM.....	19
Abbildung 3: Darstellung des Kundenstandortes in <i>Dynamics CRM 2013</i>	23
Abbildung 4: Aus <i>Dynamics CRM</i> aufgerufene <i>Bing Maps</i>	24
Abbildung 5: Screenshot der Erweiterung eMaps für Dynamics CRM.....	25
Abbildung 6: Screenshot von <i>Esri Maps for Dynamics CRM</i>	26
Abbildung 7: Angewendete Methodik	28
Abbildung 8: Struktur der Arbeit	29
Abbildung 9: Komponenten einer GeoSOA	40
Abbildung 10: Grundlegender Aufbau einer SOAP-Nachricht	42
Abbildung 11: Schematische Darstellung der OGC Webservices.....	46
Abbildung 12: In Geocoding APIs verwendete Protokolle und Datenformate.....	47
Abbildung 13: Direkter und indirekter Lagebezug	48
Abbildung 14: Schritte des Geokodierungsprozesses	49
Abbildung 15: Probleme während des Matchings	50
Abbildung 16: Clientseitiges Web Mapping.....	58
Abbildung 17: Unterscheid zwischen 1 000 Marker-Pins und Markercluster	59
Abbildung 18: Serverseitiges Web Mapping	61
Abbildung 19: Überblick Anforderungsprofil geoCRM.....	63
Abbildung 20: Use Case Diagramm geoCRM.....	63
Abbildung 21: Übersicht Anwendungsgebiete geoCRM.....	64
Abbildung 22: Skizze einer mobilen geoCRM Umfeldsuche.....	69
Abbildung 23: Skizze einer mobilen geoCRM Verkaufsrouten	70
Abbildung 24: Architekturmodell des Prototypen	72
Abbildung 25: Vereinfachter Ausschnitt aus dem CRM-Datenmodell	73
Abbildung 26: Aufteilung der Kundenadressen nach Kontinent	75
Abbildung 27: Schematischer Ablauf der Datenaufbereitung	77
Abbildung 28: Schematischer Ablauf der Pop-Up Generierung	78

Abbildung 29: Definition des Application Process in Oracle APEX	82
Abbildung 30: HTML Image Map mit Links zu den einzelnen Karten-Seiten	85
Abbildung 31: Kartendarstellung des Kundenstandorts	86
Abbildung 32: Dialogfenster zur Bearbeitung des Kundenstandorts.....	87
Abbildung 33: Anatomie einer Map-Page	88
Abbildung 34: Pop-Ups mit Detailinformationen.....	89
Abbildung 35: Standort einer Maschine	89
Abbildung 36: Dialogfenster zur Bearbeitung des Maschinenstandorts.....	90

Tabellenverzeichnis

Tabelle 1: Attribute der Klasse SDO_GEOMETRY	34
Tabelle 2: Auswahl gültiger Geometrietypen	34
Tabelle 3: Ausgewählte räumliche Operatoren.....	37

Listingverzeichnis

Listing 1: Syntax von räumlichen Operatoren (Quelle: KOTHURI et al. 2007)	36
Listing 2: Elemente einer SOAP Nachricht	41
Listing 3: Skript zur Erweiterung der Tabelle (Quelle: nach KOTHURI et al. 2007)...	74
Listing 4: Pseudocode der Prozedur zur Geokodierung der Adressdatensätze	76
Listing 5: Aufbereiteter Datensatz im JSON-Format.....	78
Listing 6: Generierter HTML-Code für den Inhalt eines Pop-Ups	79
Listing 7: Initialisierung der Karte	80
Listing 8: Hinzufügen der Layer und Kontrollelemente zur Karte	81
Listing 9: Kommunikation mit der Datenbank	82
Listing 10: Auswertung der Daten	83
Listing 11: Funktion zur Anzeige der Standortmarker und des PopUps	84

Abkürzungsverzeichnis

AJAX	Asynchronous Javascript and XML
API	Application Programming Interface
B2B	Business-to-business
B2C	Business-to-customer
BI	Business Intelligence
CRM	Customer Relationship Management
G2C	Government-to-citizen
GIF	Graphics Interchange Format
GIS	Geographisches Informationssystem
GML	Geography Markup Language
GPS	Global Positioning System
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
JPEG	Joint Photographic Expert Group
JSON	Javascript Object Notation
KML	Keyhole Markup Language
KPI	Key Performance Indicator
LBS	Location-Based Services
LI	Location Intelligence
MBR	Minimum Bounding Rectangle
OD	Open Data
OGC	Open Geospatial Consortium
OGD	Open Government Data
PNG	Portable Network Graphic
POI	Points of Interest
REST	Representational State Transfer
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol

TCP	Transmission Control Protocol
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VGI	Volunteered Geographic Information
W3C	World Wide Web Consortium
WFS	Web Feature Service
WGS	World Geodetic System
WMS	Web Map Service
WSDL	Web Services Description Language
WWW	World Wide Web
XML	Extended Markup Language

1 Einleitung

1.1 Motivation

Seit den 90er-Jahren kam es durch die Anforderung nach der Verwendung von Geoinformationen in einer steigenden Anzahl von Anwendungsfeldern zu einer Öffnung der traditionell in sich geschlossenen Geoinformationssysteme. Der Begriff „offenes GIS“ wurde geboren und Standards für die Speicherung und den interoperablen Austausch von Geodaten durch Organisationen wie das „Open Geospatial Consortium“ (OGC) und die „International Organization for Standardization“ (ISO) definiert. Im Zuge dieser Öffnung wanderte auch der Ort an dem Geodaten gespeichert wurden, von proprietären Dateiformaten in, um räumliche Komponenten erweiterte, Datenbanken. Dadurch sind heute auch räumliche Erweiterungen für alle namhaften – kommerziellen sowie Open-Source – Datenbanksysteme verfügbar (BRINKHOFF 2010).

Ein Meilenstein in der Entwicklung der breiten Nutzung von Geodaten in Web Applikationen, war 2005 die Implementierung eines auf Kartenkacheln (sog. Tiles) aufgebauten Systems, welches unter Zuhilfenahme von AJAX das Interagieren mit der Karte durch interaktives Pannen und Zoomen revolutionierte. Dieses System hieß *Google Maps* und wurde im Folgejahr um ein Application Programming Interfaces (API) erweitert, welches die Entwicklung von modernen Karten-Mashups erstmalig ermöglichte. Neben *Google* wurden auch Karten und dazugehörige APIs von zum Beispiel *Yahoo (Yahoo Maps)* oder *Microsoft* (heute *Bing Maps*) entwickelt. Karten-Mashups sind gekennzeichnet durch die Einbindung von in Drittsystemen vorgehaltener Daten und deren verortete Darstellung als Schicht über der Basiskarte (sog. Overlay). Zu Beginn waren Overlays begrenzt auf die Darstellung von Ortsinformationen in Punktform (sog. Pins), im Laufe der Entwicklung wurden auch komplexere Geometrien – wie z.B. Linien oder Polygone – sowie weitere Dienste zur Geokodierung von Adressen oder der Berechnung von Routen in den

Leistungsumfang der Karten APIs aufgenommen (PETERSON 2012a, SCHMIDT und WEISER 2012). Eine weitere wichtige Entwicklung in diesem Zusammenhang wurde durch die Einführung des *iPhones* 2007 eingeläutet, nämlich die der Smartphones und der damit ubiquitären Verfügbarkeit des Internets. Durch deren große Bildschirme und verbesserte Positionierungstechnologien wurden sie zu einem neuen, portablen Medium für (elektronische) Karten, welche darauf auch extensiv Anwendung fanden. Sogenannte Location Based Services (LBS) entstanden, mit dem Ziel dem mobilen Benutzer Informationen und Dienste in Abhängigkeit von dessen aktuellen Standort zu liefern (SCHMIDT und WEISER 2012). Die Veröffentlichung des *iPads* 2010 schaffte eine neue Kategorie mobiler Endgeräte, der Tablets, die die Interaktion mit Karten nochmals komfortabler gestalteten.

„Map interaction using a touch screen is more natural and intuitive than any other method was. Touchscreen technology offers a direct way of manipulating content and facilitates better hand-eye coordination, thus making it superior over mice or keyboards.“

(SHNEIDERMAN 1991 zitiert von SCHMIDT und WEISER 2012)

Außerdem ist mittlerweile eine Vielzahl an ausgereiften Open-Source-Lösungen in den Bereichen der Geodatenbereitstellung (z.B. *GeoServer*) oder des Web Mappings (z.B. *OpenLayers*) verfügbar. Mit der rasanten Entwicklung im Umfeld freier Geodaten – als herausragendes Beispiel ist an dieser Stelle *OpenStreetMap* zu nennen – und nicht zuletzt der aktuell weltweit, vor allem in entwickelten Ländern, an Zuwachs gewinnenden Open Data Initiativen, gibt es kostengünstige Alternativen zu kommerziellen Geodatenanbietern.

Vor dem Hintergrund der technischen Weiterentwicklungen der letzten Jahre und dank der dadurch verfügbaren GIS-Technologien, die es leichter denn je machen diese zu nutzen bzw. in bestehende, nicht räumliche Systeme einzubinden, ist es verwunderlich, dass sie nach wie vor in wenigen betrieblichen Informationssystemen

zum Einsatz kommen. Und das obwohl ein Großteil der Unternehmenskennzahlen und –daten einen Raumbezug aufweisen. Durch die Georeferenzierung dieser Daten und der Integration von GIS-Technologien in die die Daten umgebenden Prozesse und Applikationen, ergeben sich neue und optimierte Nutzungsmöglichkeiten. So können zum Beispiel alleine durch die Geovisualisierung neue Einsichten in die Daten gewonnen und damit wertvolle Beiträge zur Unterstützung von Entscheidungsfindungsprozessen einerseits und in der täglichen, operativen Arbeit andererseits geliefert werden.

1.2 Begriffseinführung

Business Geographics, Business GIS, Business Mapping, Location Intelligence oder Geomarketing – für das thematische Umfeld in welches diese Arbeit fällt, kommen all diese Begriffe in Frage. Was verbirgt sich jedoch hinter diesen – oft (fälschlicherweise) synonym verwendeten – Begriffen? Im folgenden Absatz sollen die Begriffe des thematischen Umfelds diskutiert und das Thema der Arbeit entsprechend eingeordnet werden.

Business Geographics

CZERANKA 2001 definiert Business Geographics allgemein als Markt der sich überall dort befindet *“wo im Bereich der Wirtschaft oder bei wirtschaftsbezogenen Aspekten explizit raumbezogene Fragestellungen berücksichtigt werden”* und des Weiteren als *„breites Anwendungsfeld, welches am treffendsten als Gegenstück zu dem mittlerweile bereits traditionellen GIS-Einsatz im Natur- oder Umweltschutz, Raum-, Landschafts- oder Verkehrsplanung u.ä. definiert werden kann“*.

BRINEY 2013 beschreibt Business Geographics als *„field in business that uses geographic techniques and tools to accomplish a variety of tasks important to the world of business“* und schreibt weiter, dass das meistverwendete Werkzeug mit

Bezug zur Geographie die Visualisierung räumlicher Informationen in Kartenform und die Verwendung geographischer Informationssysteme sei.

Diese beiden Definition zeigen den Zwiespalt hinsichtlich des GIS-Einsatzes als definierendes Element von Business Geographics; auch werfen sie die Frage auf, ob Business Geographics ausschließlich ein „Bereich der Wirtschaft“ ist und nicht doch auch einer der (Angewandten) Geographie.

Business GIS

Der im Vergleich zu dem, vornehmlich um die Jahrtausendwende gebräuchlichen Begriff der Business Geographics, weiter verbreitert verwendete Begriff Business GIS, stellt die technischen Aspekte des „*GIS-Einsatzes im Bereich der Wirtschaft unter Berücksichtigung des Raumbezugs in wirtschaftsbezogenen Aspekten*“ (FÄRBER 2012) in den Vordergrund. Business Mapping bezieht sich – als Teilbereich von Business GIS – in erster Linie auf die „*raumbezogene Visualisierung im Zusammenhang mit Business-Anwendungen*“ (CZERANKA 2001), wird allerdings häufig mit den hier beschriebenen Begriffen synonym verwendet.

Location Intelligence

Unter Location Intelligence wird häufig die Erweiterung von Business Intelligence Systemen um Komponenten zur räumlichen Visualisierung und Auswertung verstanden, üblicherweise durch den Einsatz von GIS. WOLFE und MOON 2011 definieren Location Intelligence allgemeiner als „*the use of locationally-referenced information as a key input in business decision making. Location intelligence uses GIS tools and techniques to transform and analyse data that becomes valuable information to make more informed and rational business decisions*“. Location Intelligence kann damit als Teilbereich von Business Geographics verstanden werden und überschneidet sich mit dem im nächsten Absatz erläuterten Geomarketing.

Geomarketing

Geomarketing, als der im deutschen Sprachraum gängigste der hier vorgestellten Begriffe, ist – wie der zuvor beschriebene Begriff Location Intelligence – ein Teilbereich von Business Geographics und umfasst laut CZERANKA 2001 *„alle raumbezogenen Aspekte des Marketings“* und verknüpft nach FÄRBER 2012 dabei *„den GIS-Einsatz mit den raumbezogenen Zielstellungen des betrieblichen Marketings“*. Ebenso „GIS-lastig“ definiert SCHÜSSLER 2006 Geomarketing als *„die Planung, Koordination und Kontrolle kundenorientierter Marktaktivitäten von Unternehmen mittels Geographischer Informationssysteme. Es werden Methoden angewendet, die den Raumbezug der untersuchten unternehmensinternen und externen Daten herstellen, analysieren und visualisieren sowie sie zu entscheidungsrelevanten Informationen aufbereiten“*.

HERTER 2008 versteht unter Geomarketing die Ergänzung des klassischen Marketingmixes um die räumliche Betrachtungsweise (siehe Abbildung 1 auf der nächsten Seite). Er diskutiert unterschiedliche Definitionen von Geomarketing und kritisiert dabei u.a. Schüsslers Zugangsweise, welche Geomarketing als ein Anwendungsgebiet von GIS sieht, statt als Teildisziplin des Marketings, deren *„mögliche Umsetzung der räumlichen Betrachtungsweise mittels GIS [...] lediglich ein Instrument des Geomarketings“* ist. Laut seiner Definition analysiert Geomarketing *„aktuelle wie potenzielle Märkte nach räumlichen Strukturen, um den Absatz von Produkten effektiver planen und messbar steuern zu können“* (HERTER 2008).

An dieser Stelle ist die Problematik der sich überschneidenden Begrifflichkeiten schon deutlich sichtbar. Die Abbildung 2 auf Seite 19 versucht, durch die graphische Darstellung der Beziehungen der einzelnen Begriffe zueinander, etwas Licht in die Sache zu bringen.

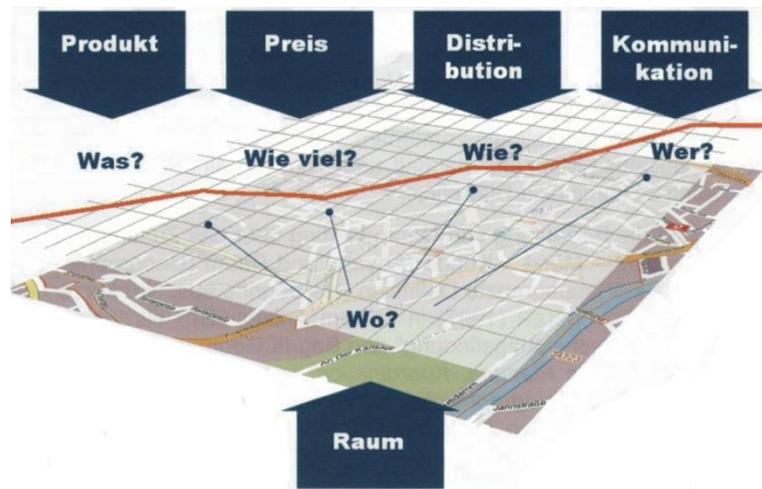


Abbildung 1: Klassischer Marketingmix ergänzt um die räumliche Sicht (Quelle: HERTER 2008)

Customer Relationship Management (CRM)

LEUBER et al. 2011 definieren das Customer Relationship Management wie folgt:

„Customer Relationship Management umfasst den Aufbau und die Festigung langfristig profitabler Kundenbeziehungen durch abgestimmte und kundenindividuelle Marketing-, Sales- und Servicekonzepte mit Hilfe moderner Informations- und Kommunikationstechnologien.“

Neben der kundenorientierten Unternehmensstrategie bedarf es zur Unterstützung der auf den Kunden ausgerichteten Geschäftsprozesse CRM-Systeme, die als integrierte Informationssysteme alle kundenbezogenen Informationen zusammenführen (LEUBER et al. 2011). Kundenbezogene Informationen sind inhärent ortsbezogen und bieten daher eine gute Grundlage für räumliche Auswertungen, Analysen und weitere Anwendungen. Im folgenden Absatz soll nun der im Titel dieser Arbeit verwendete Begriff geoCRM entsprechend eingeführt und in das Umfeld der bisher diskutierten Begriffe eingeordnet werden.

Geographic Customer Relationship Management (geoCRM)

Unter Geographic Customer Relationship Management wird im Zuge dieser Arbeit ein CRM-System verstanden, welches gezielt die räumliche Natur der

kundenbezogenen Informationen nutzt, um darauf aufbauend die operative Arbeit mit dem System zu optimieren sowie räumliche Auswertungen und Analysen zu ermöglichen. Aus technischer Sicht handelt es sich um die Anwendung und Integration von GIS-Technologien im weitesten Sinne in ein das Kundenbeziehungsmanagement unterstützendes CRM-System. Im Hinblick auf die in diesem Kapitel eingeführten Begriffe, ist geoCRM Teil der Business Geographics und befindet sich mit dem angewandten Einsatz von Business GIS im Überlappungsbereich von Business Mapping, Location Intelligence und Geomarketing, wie in der untenstehenden Abbildung 2 dargestellt.

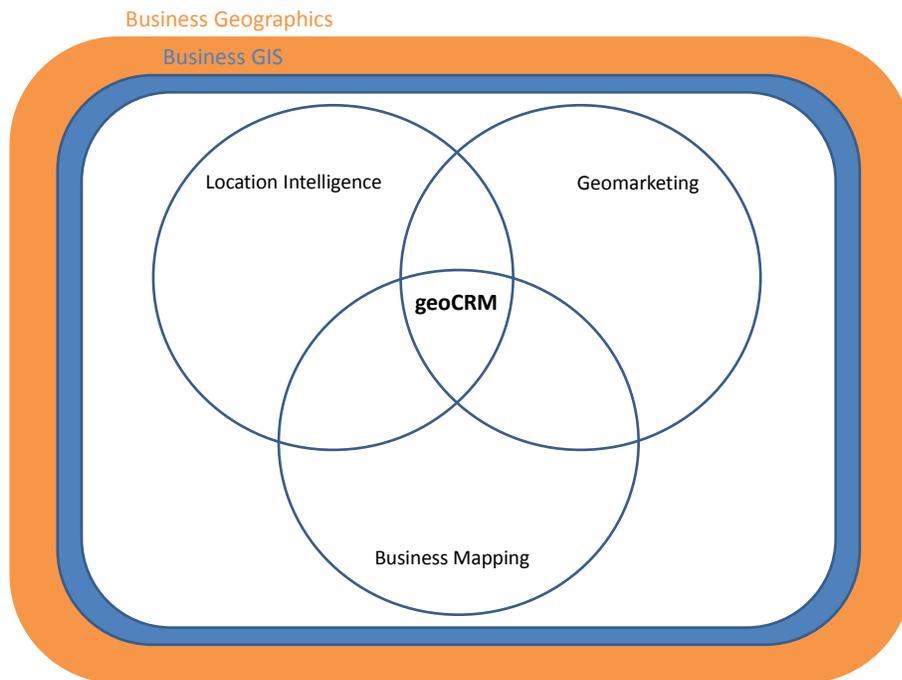


Abbildung 2: Begriffseinordnung geoCRM (Quelle: eigene Darstellung)

Nachdem in diesem Kapitel das thematische Umfeld in die diese Arbeit fällt definiert und die Arbeit begrifflich eingeordnet wurde, soll im nächsten Kapitel ein Überblick über den aktuellen Stand der Forschung gegeben werden.

1.3 Literaturüberblick

CRM-Systeme sind so unterschiedlich wie die Branchen in denen sie eingesetzt werden. Obwohl im Kern immer gleich – der kleinste gemeinsame und namensgebende Nenner ist das Management von Kundenbeziehungen – unterscheiden sich eben diese Kunden von Branche zu Branche. Auf Makroebene lässt sich eine Einteilung in B2C (Business to Customer), G2C (Government to Citizen) oder B2B (Business to Business) treffen. Jedoch sind die nach dem „to“ zusammengefassten Gruppen keineswegs homogen. Daraus ergeben sich auch unterschiedliche Ansprüche an die benötigte Genauigkeit der Standortdaten und der auf diesen basierenden Auswertungen und Analysen. Während der Markt für CRM-Systeme im B2B-Bereich von einer kleinen Anzahl großer Softwarehersteller dominiert ist, haben bisher nur wenige die Wichtigkeit der Integration von Standortdaten in ihre Produkte erkannt und diese entsprechend erweitert.

Die vorhandene Literatur lässt sich in drei Kategorien gliedern, die sich auf unterschiedlichen Ebenen mit dem Thema befassen. Eine Kategorie beschäftigt sich schwerpunktmäßig mit dem wissenschaftlichen Umfeld des Einsatzes von GIS-Technologien in der Wirtschaft und bieten – wie STROBL und FALLY 2001 – einen Querschnitt durch die vielfältigen Anwendungsgebiete; zum Teil auch mit technologischen Seitenblicken (FLEISCHMANN und KAISER 2001). Monographien und Sammelwerke in diesem Bereich verwenden meist Begriffe wie Business Geographics, Business GIS oder Geomarketing in ihren Titeln, sind dabei – vor allem bei um die Jahrtausendwende entstandenen Werken – inhaltlich allerdings schwerpunktmäßig auf Geomarketing fokussiert (STROBL und FALLY 2001, FÄRBER 2012). Einige Autoren gehen auf die Vorteile ein, die eine Nutzung von GIS im Business allgemein mit sich bringt (CZERANKA 2001, FARLEY 2004, AZAZ 2011), andere beschäftigen sich mit den organisatorischen, strategischen und finanziellen Rahmenbedingungen die bei der Einführung eines „ausgewachsenen“ GIS-Systems in ein Unternehmen berücksichtigt werden müssen (GRIMSHAW 2000, DOUGLAS 2008, MAGUIRE et al. 2008, FÄRBER 2012). Das Gebiet des Geomarketings wird mittlerwei-

le sehr ausführlich behandelt und verfügt über eine vergleichsweise ausgereifte, wissenschaftliche Basisliteratur wie beispielsweise das *Handbuch Geomarketing* von HERTER und MÜHLBAUER 2008 oder die Dissertation von SCHÜSSLER 2006 welche sich mit der Anwendung von Geomarketing im Einzelhandel beschäftigt. TAPPERT 2007 versucht eine umfassendere Auslegung des Geomarketings und geht dabei auch, als einer der wenigen Autoren, auf den B2B-Markt ein. Den Themen Business Mapping und Location Intelligence sind vereinzelt Artikel gewidmet, welche jedoch vordergründig dem Dunstkreis der Softwareunternehmen, welche Produkte in den Bereichen anbieten, entstammen (WINSLOW und CATTINI 2007, PANIAN 2012, MARTIN 2012). Die erste Kategorie zusammenfassend kann gesagt werden, dass obwohl der Themenkomplex Business Geographics seit mittlerweile zehn Jahren vereinzelt in der Forschung behandelt wird, die Anzahl an deutsch- und englischsprachigen Publikationen nach wie vor eine sehr überschaubare ist.

Die zweite Kategorie widmet sich ausgiebig mit den für die Integration von Standortdaten in betriebliche Informationssysteme notwendigen, technischen Teilgebieten und Grundlagen. Einen detaillierten Einblick in das Thema Geodatenbanken – vor allem im Hinblick auf *Oracle Spatial* – liefern BRINKHOFF 2012 mit Schwerpunkt auf den theoretischen Hintergründen und KOTHURI et al. 2007 sowie auch MURRAY 2013a mit Schwerpunkt auf die praktische Anwendung in *Oracle 11g* bzw. MURRAY 2013b in der aktuellen Version *Oracle 12c*. Viele Autoren beschäftigen sich mit der Geokodierung von Adressdatensätzen. So gibt es gute Grundlagenwerke, wie beispielsweise von BEHR 2010 oder DIOMIN 2010, Best Practice Berichte (SWIFT et al. 2008, GOLDBERG 2008) und Vergleiche verfügbarer Geokodierungsdienste (ROONGPIBOONSOPIT und KARIMI 2010, SANTOS 2012, TAMU 2013). Neben diesen Werken gibt es aber auch interessante Ansätze zur Verbesserung der Geokodierungsergebnisse (BAKSHI et al. 2004, HUTCHINSON und VEENENDAAL 2013, FLORCZYK et al. 2011, HUTCHINSON und VEENENDALL o.J.) und zu auf Volunteered Geographic Information als Referenzdatenmaterial aufbauenden Geokodierungsdiensten (AMELUNXEN 2009 zitiert von AMELUNXEN 2010). Im Gebiet

von Web Mapping liefern REHRL 2010a sowie PETERSON 2012b einen aktuellen Überblick der vorhandenen Technologien und deren Anwendung, letzterer mit besonderem Fokus auf APIs und WeServices. Die Verwendung einzelner APIs werden ausführlich in Form von technischer Ratgeber-Literatur behandelt, so beispielsweise die *Google Maps* API (SVENNERBERG 2010), die *Bing Maps* API (KAUER 2011) sowie das Open-Source Mapping Framework *OpenLayers* (JANSEN und ADAMS 2010, HAZZARD 2011) oder der Open-Source Mapserver *Geoserver* (IACOVELLA und YOUNGBLOOD 2013). Vergleiche der aktuell verfügbaren client- wie serverseitigen Web Mapping Komponenten sind in FERNANDES et al. 2013 bzw. mit speziellen Fokus auf Open-Source Technologien in BALLATORE et al. 2011 zu finden.

In die dritte Kategorie fallen Werke die sich mit der Integration von GIS-Technologien in CRM-Systemen – also mit geoCRM-Systemen – beschäftigen bzw. der Einbindung und Nutzung georeferenzierter Informationen in solche. Obwohl die wenigen, durchwegs vergleichsweisen jungen Arbeiten – meist in Form von Zeitschriftenartikeln – allesamt auf das außerordentliche Potential dieser Integration hinweisen (NITSCHKE 2008, TSAI 2010, HILLMANN-KÖSTER 2010 bzw. UFFMANN 2011, HONGJIAN et al. 2011, MCNEIL 2013), gibt es in diesem Gebiet noch äußerst wenig an – vor allem wissenschaftlicher – Literatur. Eine Ausnahme stellt dabei PLANK 2004 dar, die in ihrer Diplomarbeit GIS-Funktionalitäten von CRM-Lösungen im Energie-versorgungsunternehmen analysierte und WINKELMANN 2012, der in seinem umfassenden Werk zu den Instrumenten des integrierten Kundenmanagements auch auf die CRM-integrierte (operative) Besuchstourenplanung und die Verkaufsgebiets-optimierung eingeht. Die Recherchen des Autors haben folglich ergeben, dass zum Thema der Integration von GIS-Technologien in CRM-Systemen ein breites Forschungsfeld weitgehend offen ist; vor allem im B2B-Bereich aber auch im B2C-Bereich. Die vorliegende Arbeit soll nun zu dem Thema einen Beitrag liefern und versucht dabei die vorhandene Forschungslücke zumindest etwas zu verkleinern. Im anschließenden Kapitel werden bestehende Ansätze von geoCRM-Systemen anhand von ausgewählten Fallbeispielen vorgestellt.

1.4 Bestehende Ansätze

In diesem Kapitel soll anhand von Fallbeispielen dargestellt werden, wie GIS-Technologien in CRM-Systeme integriert werden können. Als Beispiele für CRM-Systeme werden dabei „Geo-Erweiterungen“ für das Produkt *Microsoft Dynamics CRM* näher betrachtet.

In *Microsoft Dynamics CRM* wird seit dem Release der Version 2013 eine einfache Karte mit dem geokodierten Standort des Kunden angezeigt (DICKINSON 2013), wie in Abbildung 3 dargestellt. Des Weiteren besteht die Möglichkeit *Bing Maps* in Verbindung mit dem Customizing der Software zu verwenden (Microsoft 2013).

The screenshot displays the Microsoft Dynamics CRM 2013 interface for an account named "A. Datum Corporation (sample)". The top navigation bar includes "Microsoft Dynamics CRM", "SALES", "Accounts", and "A. Datum Corporati...". Below the navigation bar, there are action buttons: "+ NEW", "DEACTIVATE", "CONNECT", "ADD TO MARKETING LIST", "ASSIGN", and "...".

The account name "A. Datum Corporation (sample)" is prominently displayed. Below it, a "Summary" section is visible, followed by "ACCOUNT INFORMATION" and "ADDRESS".

ACCOUNT INFORMATION

Account Name *	A. Datum Corporation (sample)
Phone	555-0158
Fax	--
Website	http://www.adatum.com/
Parent Account	--
Ticker Symbol	--

ADDRESS

2137 Birchwood Dr
Redmond, WA 98214
U.S.

The map below the address shows the location of the account. A green arrow points from the address text to the map. The map displays a street grid with labels for "152nd Ave NE", "160th Ave NE", "NE 85th St", "NE 83rd St", "161st Ave NE", "NE 87th St", "SAM MAMISH TRAIL", and "Edge Skate Park". A blue dot on the map indicates the account's location at 2137 Birchwood Dr. The map is credited to "© 2013 Microsoft Corporation" and "© 2013 Nokia".

POSTS ACTIVITIES NOTES

Enter post here

Both Auto posts User posts

- Required Service (sample)**
Case: Closed by Jeremy Hootington for A
On Required Service (sample)'s wall
10/18/2013 11:37 AM
- Operating manual required (sample)**
Case: Closed by Jeremy Hootington for A
On Operating manual required (sample)'
10/18/2013 11:37 AM
- Required Service (sample)**
Case: Created by Jeremy Hootington for
On Required Service (sample)'s wall
10/18/2013 11:37 AM
- Operating manual required (sample)**
Case: Created by Jeremy Hootington for
On Operating manual required (sample)'
10/18/2013 11:37 AM
- Missing parts (sample)**
Case: Created by Jeremy Hootington for
On Missing parts (sample)'s wall
10/18/2013 11:37 AM
- Very likely will order 73 Product SKU**
Jeremy Hootington won Opportunity for
Congratulations!
On Very likely will order 73 Product SKU

Abbildung 3: Darstellung des Kundenstandortes in *Dynamics CRM 2013* (Quelle: DICKINSON 2013)

Bei Klick auf die in Abbildung 3 dargestellten Karte, wird in einem neuen Browserfenster *Bing Maps* geladen (siehe untenstehende Abbildung) in dem dessen

Standard-Funktionalitäten genutzt werden können (DICKINSON 2013). Die Interaktion zwischen *Dynamics CRM* und *Bing Maps* beschränkt sich dabei auf die Übergabe der Adresse von einem System in das andere.

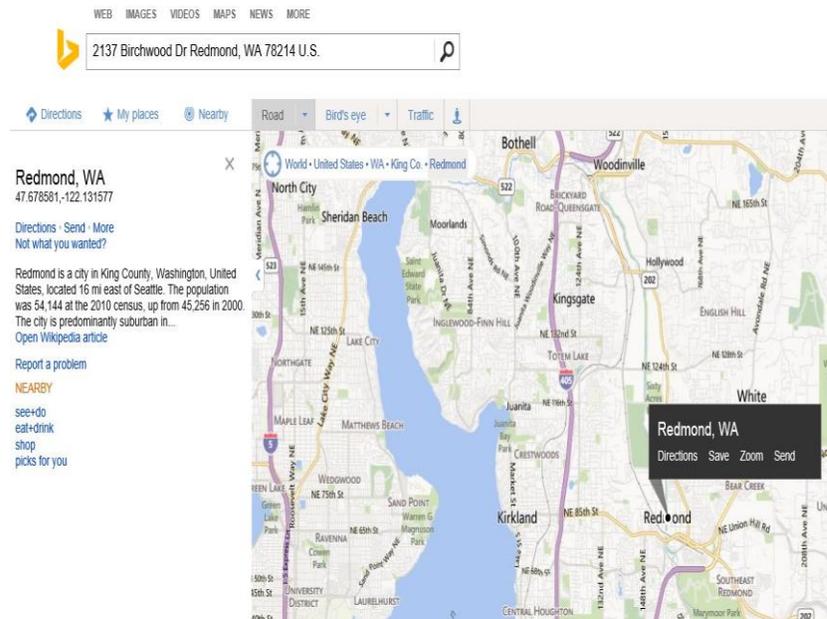


Abbildung 4: Aus *Dynamics CRM* aufgerufene *Bing Maps* (Quelle: DICKINSON 2013)

Die Firma *My CRM*¹ bietet mit dem Produkt *eMap* eine auf *Bing Maps* basierende Erweiterung für *Dynamics CRM* an, die es erlaubt, vorhandene Daten auf einer kartenbasierenden Benutzeroberfläche darzustellen und nach verschiedenen Kriterien zu filtern. Bei den dargestellten Daten handelt es sich um unterschiedliche Business Objekte (sog. Accounts, Contacts, Leads, Opportunities etc.), welche mit Hilfe von Pin-Layern auf der Karte angezeigt werden können. Details zu einzelnen Objekten werden mittels Pop-Ups angezeigt. Die zum Einsatz kommenden Filterkriterien umfassen einen „Quick Filter“ zur Einschränkung der Daten anhand des jeweiligen Objektnamens, einen „Entity Distance Filter“ der Daten innerhalb eines gewissen Radius um ein Objekt anzeigt und einen „Location Distance Filter“ der Daten im Umfeld um einen bestimmten Ort auf der Karten anzeigt. Zusätzlich bietet die

¹ <http://www.mycrmgroup.com/>

Erweiterung noch Funktionalitäten zur Routenplanung und zur Erstellung von Listen auf Basis der auf der Karte dargestellten Objekte. Die untenstehende Abbildung zeigt *Microsoft Dynamics CRM 2013* mit der beschriebenen Erweiterung *eMaps*.

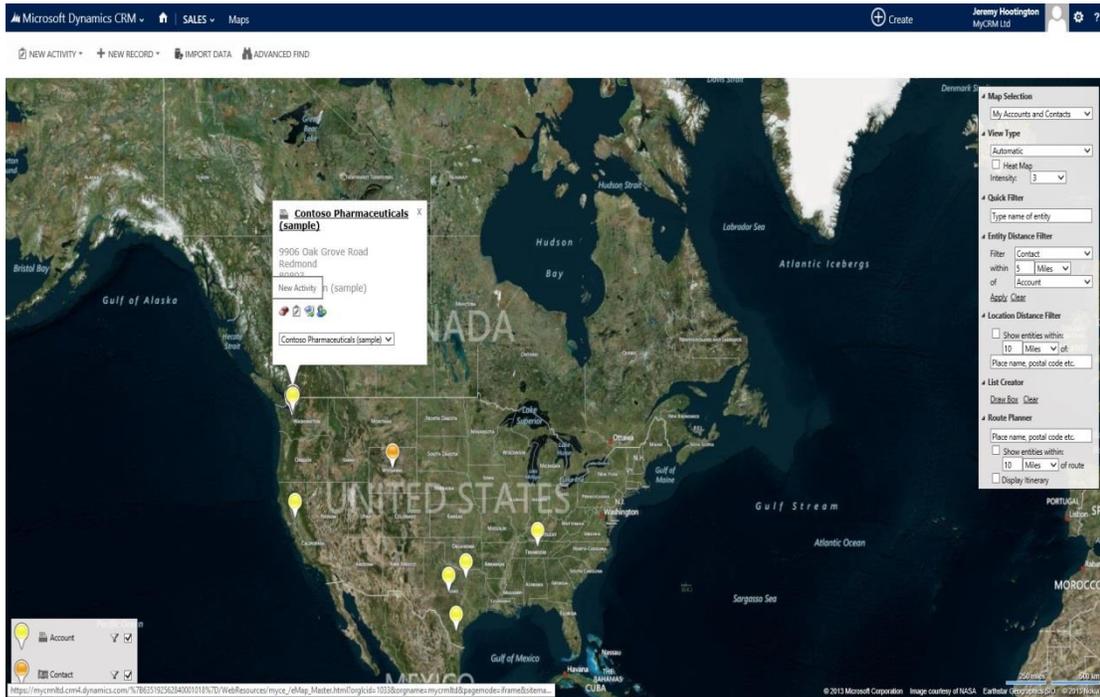


Abbildung 5: Screenshot der Erweiterung eMaps für Dynamics CRM (Quelle: DICKINSON 2013)

Die Firma *ESRI*² teilte Anfang 2014 die Veröffentlichung einer Dynamics CRM Erweiterung in einer Pressemitteilung mit (ESRI 2014). Über das Produkt ist momentan noch sehr wenig zu erfahren. Bis auf eine Handvoll Schlagwörter und dem in Abbildung 6 auf der nächsten Seite dargestellten Screenshot, welcher ein um zwei Karten erweitertes Dashboard zeigt, werden auf der Homepage keine Details verraten. Das Produkt verwendet auf ESRI-eigene Basiskarten und verwendet Funktionalitäten von ArcGIS Online zum Teilen erstellter Karten und ArcGIS Server zur Analyse. Es wird über die „Location Analytics“ Produktschiene vertrieben, was neben der folgenden Beschreibung aus der Presseaussendung darauf schließen lässt, dass auch bei dieser Erweiterung Auswertung und Visualisierung der Daten im

² <http://www.esri.com/>

Vordergrund stehen: „Esri Maps adds new location-oriented workflows that uncover hidden trends and patterns with interactive tools, data, and analysis. Now, Microsoft Dynamics CRM users can author maps and interactive infographics from their internal information resources. These interactive visualizations and analytics uncover customer intelligence and operational knowledge.“ (ESRI 2014)

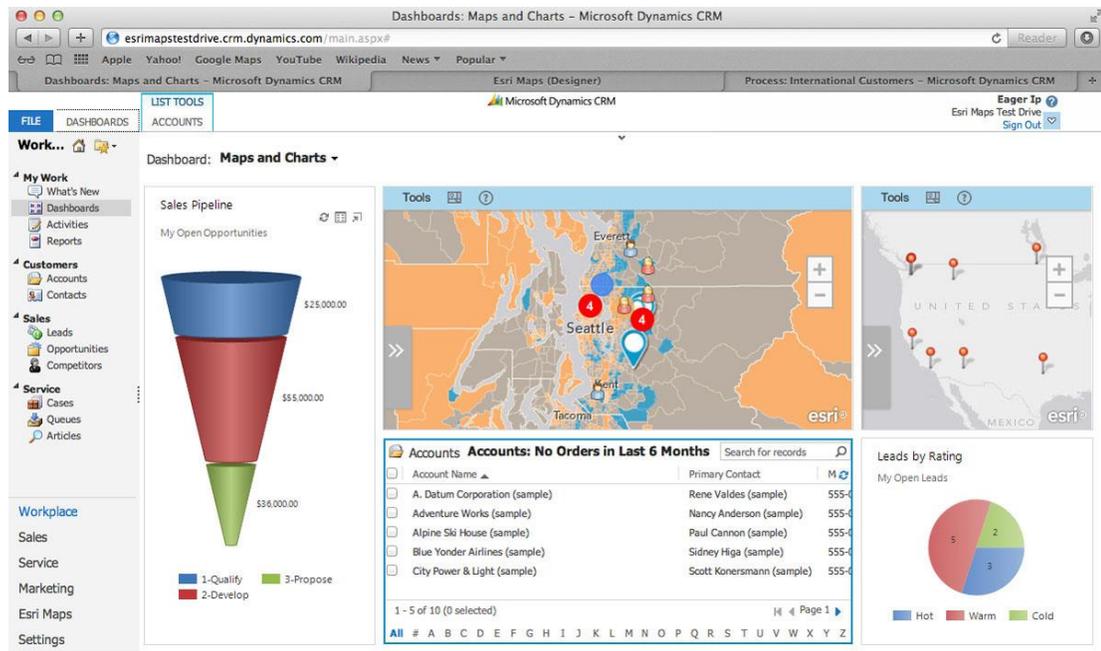


Abbildung 6: Screenshot von Esri Maps for Dynamics CRM (Quelle: ESRI 2014)

Die beiden Erweiterungen zeigen beispielhaft, dass GIS-Funktionalitäten nicht in den Standardfunktionalitäten von CRM-Systemen enthalten sind, sondern durch von Drittanbietern bereitgestellten Erweiterungen in bestehende Systeme integriert werden. Diese Erweiterungen konzentrieren sich dabei auf die Visualisierung der CRM-Daten in Form von Pins auf Karten in Verbindung mit unterschiedlichen Filtermöglichkeiten. Die Auswertung und Analyse von aggregierten Informationen anhand z.B. thematischer Karten bleibt aktuell den von CRM-Systemen getrennten Toolsets des Reportings und den dafür verfügbaren „Geo-Erweiterungen“ vorbehalten.

1.5 Ziele

Diese Arbeit untersucht, ob und wie die Entwicklungen der letzten Jahre in den Bereichen der GIS-Technologien im Umfeld betrieblicher Informationssysteme, mit Fokus auf das Kundenbeziehungsmanagement, genutzt werden können. Vor dem strategischen Ziel der Optimierung des Kundenbeziehungsmanagements, soll anhand der prototypischen Integration von GIS-Technologien in ein CRM-System evaluiert werden, ob die technologischen Rahmenbedingungen als Voraussetzung und Grundlage zur Erfüllung des Zieles vorhanden sind. Daraus abgeleitet ergeben sich die folgenden zwei Hypothesen, die anhand der Arbeit überprüft werden.

Hypothese 1:

Die technologischen Rahmenbedingungen sind vorhanden, um GIS-Technologien nahtlos in betriebliche Informationssysteme integrieren zu können.

Hypothese 2:

Die Integration von GIS-Technologien in CRM-Systeme führt zu einer Optimierung der operativen Verkaufsprozesse, sowie der strategischen Entscheidungsfindungsprozesse.

Im nächsten Kapitel wird beschrieben, welche Teilschritte für die Erreichung der Ziele vorgenommen und mit Hilfe welcher Methodik die aufgestellten Hypothesen überprüft werden sollen.

1.6 Methodik

Nach einer einführenden Literaturrecherche werden die für das weitere Verständnis notwendigen, technologischen und theoretischen Grundlagen aufbereitet. Darauf aufbauend werden mögliche Anwendungsgebiete von GIS-Technologien im Kundenbeziehungsmanagement diskutiert. Zur Überprüfung der ersten Hypothese wird eines

der Anwendungsgebiete als Prototyp für ein räumlich explizites CRM-System umgesetzt. Für die Umsetzung des geoCRM-Prototyps werden folgende Teilschritte aufeinander aufbauend durchlaufen:

- ⇒ Erweiterung des Datenmodells zur Speicherung georeferenzierter Daten
- ⇒ Geokodierung bestehender Adressdatensätze
- ⇒ Integration der Geokodierungsfunktionalität in die Applikation
- ⇒ Implementierung einer Web Mapping-Komponente
- ⇒ Integration der Web Mapping-Komponente in die Applikation

In weiterer Folge soll der Prototyp einem ausgewählten Expertenkreis aus Verkaufsmitarbeitern und Managern vorgestellt werden. Zur Überprüfung der zweiten Hypothese sollen diese befragt werden, ob und wie die entwickelten räumlichen Funktionalitäten ihrer Ansicht nach Verkaufs- und Entscheidungsfindungsprozesse optimieren. Die untenstehende Abbildung gibt einen Überblick der einzelnen Phasen der Arbeit.



Abbildung 7: Angewendete Methodik (Quelle: eigene Darstellung)

1.7 Rahmenbedingungen

Die Arbeit bewegt sich im institutionellen Umfeld eines international tätigen Konzerns in der Maschinenbaubranche und damit im Business-to-business (B2B)

Bereich. Das in diesem Unternehmen zum Einsatz kommenden CRM-System wird derzeit konzernweit von knapp 2 000 Mitarbeitern genutzt.

Die technologische Basis des CRM-Systems ist eine *Oracle* Datenbank und es wurde auf der webbasierten Rapid Application Development Plattform *Oracle Application Express* entwickelt. Da die gewonnenen Erkenntnisse dieser Arbeit in das CRM-System einfließen und in weiterer Folge in dieses implementiert werden sollen, wurde dabei auf die Integrierbarkeit in die verwendeten Technologien entsprechend geachtet.

1.8 Struktur der Arbeit

Die untenstehende Abbildung gibt einen Überblick über den Aufbau der Arbeit. In weiterer Folge wird auf den Inhalt der einzelnen Kapitel näher eingegangen.

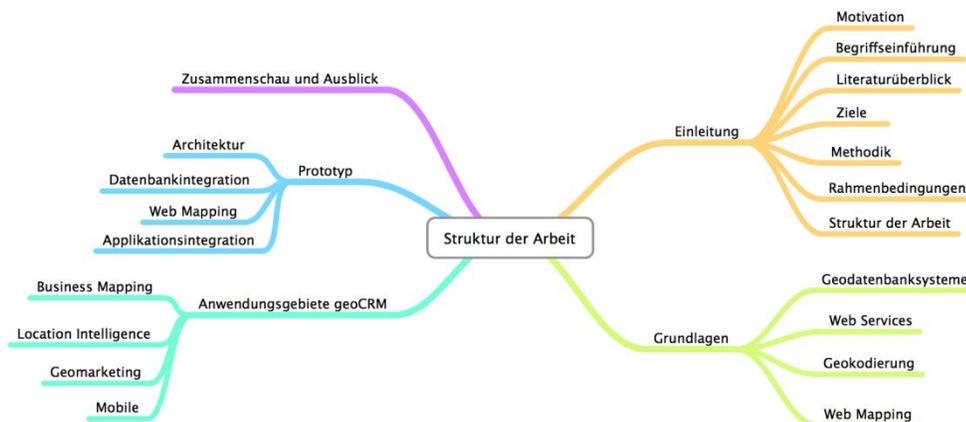


Abbildung 8: Struktur der Arbeit (Quelle: eigene Darstellung)

In Kapitel 1 wird zunächst in das Thema eingeleitet, die Motivation dafür beschrieben und werden verwendete Begriffe eingeführt. Nach einem Überblick über den Stand der Forschung wird auf die Ziele der Arbeit und die zu deren Erreichung angewandte

Methodik eingegangen. Eine Beschreibung der Rahmenbedingungen und eine thematische Abgrenzung der Arbeit runden dieses Kapitel ab.

Im Kapitel 2 erfolgt eine Einführung in die notwendigen theoretischen und technologischen Grundlagen, die für das Verständnis und die Bearbeitung relevant sind. Dabei wird auf Geodatenbanksysteme, Web Services, Geokodierung und Web Mapping eingegangen.

In Kapitel 3 wird u.a. mit Hilfe eines UML Use Case Diagrammes – welches die Interaktionen der involvierten Benutzergruppen mit dem System darstellt – ein mögliches Anforderungsprofil an ein geoCRM System skizziert.

Das Kapitel 4 gibt einen Überblick über Möglichkeiten die eine Integration von GIS-Technologien in CRM-Systeme eröffnet. Dabei werden aus den Perspektiven von Business Mapping, Location Intelligence, Geomarketing und der mobilen Nutzung potentielle Anwendungsgebiete entwickelt.

Im Kapitel 5 wird eines der Anwendungsgebiete als Prototyp eines räumlich expliziten CRM-Systems umgesetzt. Dabei wird auf das für die Integration verwendete Architekturmodell eingegangen, die Integration in die Datenbank sowie die Geokodierung der bestehenden Adressdatensätze beschrieben. Nach der Entwicklung einer darauf aufbauenden Web Mapping Komponente, soll die Integration in die bestehende CRM-Applikation diskutiert werden.

Das Kapitel 6 fasst Ergebnisse der Prototypentwicklung und der Expertenbefragung zusammen, evaluiert diese und gibt einen Ausblick auf weitere Forschungsaufgaben in diesem Gebiet. Des Weiteren werden die gewonnenen Erkenntnisse im Hinblick auf die Literatur diskutiert und ein Fazit gezogen.

1.9 Abgrenzung

In der Master Thesis wird auf eine detaillierte Evaluierung der zum Einsatz kommenden Geokodierungsdienste und client- wie serverseitigen Web Mapping Produkte verzichtet. Die beschriebenen potentiellen Anwendungsgebiete stellen Beispiele dar und erheben keinen Anspruch auf Vollständigkeit. Die Thematik der mobilen Nutzung beschränkt sich auf den Bereich geoCRM und grenzt dabei das ausführlich in der Literatur behandelte Themenfeld des mobilen CRM (mCRM) bewusst weitestmöglich aus. Bei der Entwicklung der Web Mapping Komponente wird nicht explizit auf kartographische Aspekte der Informationsdarstellung eingegangen. Obwohl Teile des entwickelten Prototyps in ähnlichen Aufgabenstellungen verwendet werden können, war die Entwicklung einer generisch einsetzbaren, interoperablen Lösung nicht Ziel der Arbeit.

2 Grundlagen

In diesem Abschnitt werden einige Begriffe aus dem Umfeld der Thematik dieser Arbeit kurz definiert bzw. erläutert. Auf eine ausführliche Darstellung der Begriffe wurde dabei bewusst verzichtet, da dies einerseits den Rahmen der Arbeit sprengen würde und andererseits ein oberflächliches Wissen darüber für das Verständnis der Arbeit ausreicht, welches zum Teil auch vorausgesetzt wird. Es werden dabei vor allem die technischen Komponenten des im Kapitel 5 entwickelten Prototypen vorgestellt.

2.1 Geodatenbanksysteme

Viele kommerzielle und Open-Source-Datenbanksysteme bieten heute die Möglichkeit, räumliche Daten abzuspeichern und entsprechend zu verarbeiten, was die Integration von Geodaten in die bestehenden IT-Infrastrukturen von Organisationen und in deren Geschäftsprozesse ermöglicht (BRINKHOFF 2012). Als Beispiele wären an dieser Stelle die räumlichen Erweiterungen *Oracle Locator* und *Oracle Spatial* für *Oracle* Datenbanken oder *PostGIS* für das Open-Source-Datenbanksystem *PostgreSQL* zu nennen. Die Schaffung dieser räumlichen Erweiterungen setzt voraus, dass das jeweilige Datenbanksystem ein **objektrelationales** ist, wodurch es hinreichende Modellierungs- und Erweiterungsmöglichkeiten zur Einbindung geometrischer Datentypen besitzt. Neben der Bereitstellung geometrischer Datentypen bzw. deren Klassenmodellen, werden Funktionen zur Beantwortung geometrischer wie topologischer Fragestellungen, sowie Indizes zur effizienten Bearbeitung räumlicher Anfragen zur Verfügung gestellt (BRINKHOFF 2010).

Räumliche Erweiterungen von Datenbanksystemen bestehen vereinfacht zusammengefasst aus den drei Elementen geometrische Datentypen, geometrische Operatoren und Funktionen und räumliche Indizes, welche folgend am Beispiel der

räumlichen Erweiterung für *Oracle* Datenbanken betrachtet werden sollen. Das Lizenzmodell von *Oracle* unterscheidet dabei zwischen den vom Funktionsumfang aufeinander aufbauenden Produkten *Oracle Locator* und *Oracle Spatial*. Im Verlauf dieser Arbeit wird nicht explizit auf die Unterschiede der beiden Produkte Rücksicht genommen und *Oracle Spatial* als Überbegriff verwendet.

2.1.1 Geometrische Datentypen

Oracle Spatial bietet zur Verwaltung räumlicher Daten **eine** generelle Geometrieklasse mit dem Namen `SDO_GEOMETRY` an. Diese Geometrieklasse kann eine breite Palette an räumlichen Daten speichern, unter anderen

- Punkte
- Linienzüge
- Polygone und
- komplexe Geometrien

Zuerst wird nun der Aufbau der Klasse selbst und die Bedeutung ihrer Attribute erklärt, danach wird anhand eines Beispiels die Abbildung und Verwendung der Klasse zur Speicherung einfacher Punktgeometrien gezeigt. Die Klasse `SDO_GEOMETRY` besteht aus fünf in der Tabelle auf der nächsten Seite aufgelisteten Attributen zur Definition der in ihr abgelegten Geometrien.

Name	Typ	Beschreibung
SDO_GTYPE	NUMBER	Geometriotyp
SDO_SRID	NUMBER	ID des räuml. Bezugssystems
SDO_POINT	SDO_POINT_TYPE	Punktgeometrie
SDO_ELEM_INFO	SDO_ELEM_INFO_ARRAY	Interpretation der Koordinaten
SDO_ORDINATES	SDO_ORDINATE_ARRAY	Koordinaten

Tabelle 1: Attribute der Klasse SDO_GEOMETRY (Quelle: BRINKHOFF 2012)

Das Attribut SDO_GTYPE definiert den Geometriotyp (Punkt, Linie, Polygon etc.) anhand einer vierstelligen Zahl, welche folgendermaßen strukturiert ist: D00T. Die erste und die letzte Ziffer definieren die Dimension und den Typ der Geometrie (KOTHURI et al. 2007). Tabelle 2 gibt eine Übersicht über die für diese Arbeit wichtigsten Geometriotypen. Eine vollständige Auflistung gültiger Geometriotypen von SDO_GTYPE ist in Anhang B ersichtlich.

Wert von SDO_GTYPE	Geometriotyp nach OGC-Spezifikation ³	Beschreibung
D001	POINT	Punkt
D002	LINE or CURVE	Linienzug
D003	POLYGON or SURFACE	Einfaches Polygon, ggf. mit Löchern

Tabelle 2: Auswahl gültiger Geometriotypen (Quelle: BRINKHOFF 2012)

³ OGC-Spezifikation zu Simple Feature Access, siehe <http://www.opengeospatial.org/standards/sfa>

Das Attribut `SDO_SRID` gibt die ID des räumlichen Bezugssystems (Koordinatensystem) an in dem die Geometrie spezifiziert ist. Eine Liste aller in *Oracle Spatial* verfügbarer Bezugssysteme liefert die Datenbank-Tabelle `SDO_COORD_REF_SYS` (MURRAY 2013a). Die SRID für WGS 84 ist zum Beispiel mit der ID 8307 definiert.

Mit Hilfe des Attributes `SDO_POINT` lassen sich einfach einzelne Punkte (vom Geometrietyp 2001 für zweidimensionale bzw. 3001 für dreidimensionale Punkte) definieren. Das Attribut ist vom Typ `SDO_POINT_TYPE` der aus drei Koordinatenwerten (X, Y, Z) besteht. Bei zweidimensionalen Punkten wird der der Z-Koordinate ggf. zugewiesene Wert ignoriert. Bei Nutzung des Attributes `SDO_POINT` werden die beiden restlichen Attribute von `SDO_GEOMETRY` (`SDO_ELEM_INFO` und `SDO_ORDINATES`, siehe Tabelle 1) nicht verwendet (BRINKHOFF 2012).

Sollen komplexere Geometrien als Punkte abgespeichert werden, kommen die beiden Attribute `SDO_ELEM_INFO` und `SDO_ORDINATES` zum Einsatz. Beim Attribut `SDO_ORDINATES` handelt es sich um ein Zahlenfeld in dem alle Koordinaten einer Geometrie gespeichert werden. Besteht eine komplexe Geometrie aus mehreren Teilen, werden diese einfach hintereinander angeführt. Die Bedeutung der einzelnen Teile wird anhand des Attributes `SDO_ELEM_INFO` bestimmt, welches ebenso als Zahlenfeld definiert ist. Innerhalb dieses Zahlenfeldes werden Dreiergruppen mit den folgenden Elementen gespeichert:

- *Offset*: dieser Wert bezieht sich auf die jeweilige, durch die Dreiergruppe definierte Position im Koordinatenfeld `SDO_ORDINATES`.
- *Elementtyp*: beschreibt den Typ der Teilgeometrie (Punkt, Linie, Polygon o.a.)
- *Interpretation*: spezifiziert den Elementtyp näher (z.B. Anzahl der Punkte)

Auf Basis des in diesem Abschnitt vorgestellten geometrischen Datentyps stellt *Oracle Spatial* eine Vielzahl räumlicher Operatoren und Funktionen zu Verfügung. Einen kurzen Überblick über diese gibt der folgende Abschnitt.

2.1.2 Räumliche Operatoren

Wie relationale Operatoren in SQL Statements (z.B. „<“, „>“, „=“, ...) zur Abfrage von logischen Sachverhalten, gibt es räumliche Operatoren (z.B. `SDO_WITHIN_DISTANCE`), die in der `WHERE`-Bedingung einer (räumlichen) Abfrage angeführt werden können. Eine Voraussetzung zur Verwendung von räumlichen Operatoren, ist die Definition eines räumlichen Indexes vom Typ `MDSYS.SPATIAL_INDEX` für alle im Operator verwendeten `SDO_GEOMETRY` Spalten (KOTHURI et al. 2007). Alle räumlichen Operatoren in *Oracle Spatial* folgen dem in Listing 1 beschriebenen, generischen Syntax.

```
<spatial_operator>
(
  table_geometry          IN SDO_GEOMETRY,
  query_geometry         IN SDO_GEOMETRY
  [, parameter_string    IN VARCHAR2
  [, tag                 IN NUMBER]]
)
= ,TRUE`
```

Listing 1: Syntax von räumlichen Operatoren (Quelle: KOTHURI et al. 2007)

Eine Übersicht über die im Zuge dieser Arbeit wichtigsten, zur Verfügung stehenden Operatoren und deren potentielle Anwendungsgebiete im Umfeld des Kundenbeziehungsmanagements gibt die folgende Tabelle.

Operator	Beschreibung
SDO_WITHIN_DISTANCE	Ermittelt alle Geometrien die innerhalb eines definierten Entfernungsradius liegen. So können z.B. alle Kunden innerhalb eines 10 km Radius um einen Firmenstandort ermittelt werden.
SDO_NN	Ermittelt die nächsten Nachbarn um einen bestimmten Punkt und sortiert das Ergebnis nach der aufsteigenden Distanz zu diesem Punkt. Dadurch können z.B. die nächsten Service Center um einen Kundenstandort ermittelt werden.
SDO_INSIDE / SDO_CONTAINS	Die beiden logisch gegengleichen Operatoren ermitteln, ob sich eine Geometrie innerhalb einer anderen befindet. Eine beispielhafte Anwendung wäre die Ermittlung von Kunden innerhalb einer Verkaufsregion.

Tabelle 3: Ausgewählte räumliche Operatoren (Quelle: nach KOTHURI et al. 2007)

Die beiden Operatoren zur Umfeldsuche (SDO_WITHIN_DISTANCE und SDO_NN) sind insbesondere in Verbindung mit der mobilen Nutzung eines geoCRM von Bedeutung (siehe Kapitel 0). Dabei können räumliche Points of Interest (POI) aus der Kundendatenbank in Abhängigkeit des aktuellen Standorts eines Verkaufs- oder Servicemitarbeiters ermittelt und dargestellt werden. Sie sind damit auch eine Voraussetzung für die Bereitstellung von standortbezogenen Diensten (engl. Location-based Services (LBS)).

2.1.3 Räumliche Funktionen

Im Unterschied zu den im letzten Abschnitt vorgestellten räumlichen Operatoren, benötigen die räumlichen Funktionen keinen räumlichen Index und können sowohl in der *WHERE*-Bedingung als auch in der *SELECT*-Liste Anwendung finden. Die in *Oracle Spatial* zu Verfügung stehenden räumlichen Funktionen lassen sich nach KOTHURI et al. 2007 in folgende fünf Kategorien einteilen:

- Bufferfunktionen
- Funktionen zur Beziehungsanalyse zwischen Geometrien (z.B. Distanz)
- Funktionen zur Kombination von Geometrien (z.B. Union, Intersection)
- Geometrische Analysefunktionen (z.B. Fläche, Länge, Zentroid)
- Geometrische Aggregationsfunktionen (z.B. konvexe Hülle, MBR)

Diese Funktionen finden in erster Linie bei der räumlichen Analyse im Zuge der Anwendung von Geomarketingmethoden Einsatz. Im Bereich geoCRM sind sie weniger von Bedeutung, am ehesten noch in der großmaßstäbigen Analyse von Verkaufsgebieten. Eine zusätzliche Kategorie an Funktionen beinhaltet solche für den sog. Transport (Import und Export) von räumlichen Daten und für deren Validierung. So gibt es Funktionen für die Konvertierung von *SDO_GEOMETRY* Daten u.a. in die Formate GML, KML oder WKT.

Weiter in die Tiefe soll an dieser Stelle nicht gegangen werden, da die Thematik bereits von anderen Autoren ausführlich behandelt wurde. Einen detaillierten Einblick in das Thema – vor allem im Hinblick auf *Oracle Spatial* – liefern BRINKHOFF 2012 mit Schwerpunkt auf den theoretischen Hintergründen und KOTHURI et al. 2007 sowie auch MURRAY 2013a mit Schwerpunkt auf die praktische Anwendung in *Oracle 11g* bzw. MURRAY 2013b in der aktuellen Version *Oracle 12c*. Für einen Einstieg in das Thema Geodatenbanksysteme gibt BRINKHOFF 2010 einen guten ersten Überblick.

Im Rahmen dieser Arbeit finden die von Geodatenbanksystemen bereitgestellten Technologien bei der im Kapiteln 5.2.2 beschriebenen Erweiterung eines Datenmodells im CRM-Umfeld Anwendung. Ebenso bei der Anreicherung der vorhandenen Daten um räumliche Informationen durch Georeferenzierung in Kapitel 5.2.4 sowie bei der Abfrage räumlicher Daten im Zuge der Geovisualisierung mittels Web Mapping in Kapitel 5.2.5 Anwendung.

Viele der bei einer Integration von GIS-Technologien in ein CRM-System zum Einsatz kommenden Technologien – als Beispiele wäre an dieser Stelle die Geokodierung und das Web Mapping zu nennen – werden als Web Service zur Verfügung gestellt und auch als solcher in eine Applikation eingebunden bzw. wiederum anderen Applikationen angeboten. Aus diesem Grund ist ein Verständnis von Web Services und deren unterschiedlichen Architekturen notwendig. Im nächsten Kapitel wird daher beschrieben, wie Daten und/oder Geschäftslogik aus einer Geodatenbank mit Hilfe von Web Services einem breiten – internen wie externen – Benutzerkreis zu Verfügung gestellt werden können und welche Arten von Web Services im Web Mapping Umfeld Anwendung finden.

2.2 Web Services

Bei der Bereitstellung von Anwendungen oder Diensten im Internet ist heute oft von serviceorientierten Architekturen (SOA) die Rede. Auch im GIS-Umfeld kommen diese Architekturen zum Einsatz, die den Aufbau verteilter Systeme überhaupt erst ermöglichen. Eine SOA besteht – wie in Abbildung 9 dargestellt - aus den Komponenten *Anwendungen*, *Dienste* und *Dienste-Support* und stellt die vermittelnde Schicht zwischen den Konsumenten und den Produzenten der zu Verfügung gestellten Dienste oder Web Services dar (ESRI 2007).

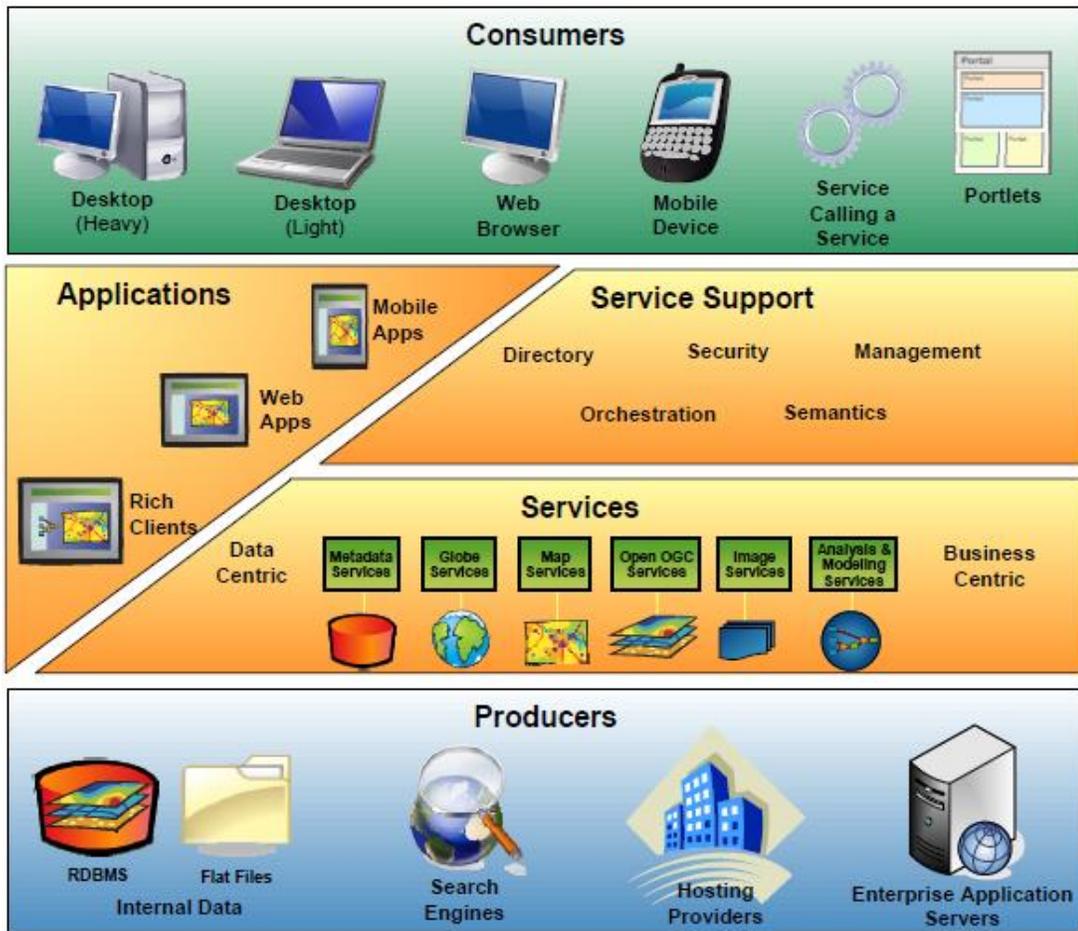


Abbildung 9: Komponenten einer GeoSOA (Quelle: ESRI 2007)

Die Konsumenten kommunizieren mit der SOA bzw. den über eine solche angebotenen Services unter Zuhilfenahme verschiedener Protokolle. Die zwei aufeinandertreffenden Zugänge sind **SOAP** (ursprünglich Simple Object Access Protocol) einerseits und **REST** (Representational State Transfer) andererseits. In den folgenden zwei Absätzen sollen diese beiden Zugänge kurz erläutert werden, da die in den Kapiteln 2.3.2 und 2.4.2 vorgestellten Geokodierungs- und Web Mapping APIs in erster Linie auf SOAP und REST basieren. Im Anschluss folgt eine Beschreibung der offenen Standards des OGC⁴ **Web Map Service** (WMS) und **Web Feature Service** (WFS), die bei der interoperablen Bereitstellung von Geodaten – nicht nur im

⁴ Open Geospatial Consortium, siehe <http://www.opengeospatial.org/>

Open-Source-Bereich – zunehmend Verbreitung finden und dadurch an Wichtigkeit gewinnen.

2.2.1 SOAP

Die Grundidee von SOAP ist, die Kommunikation mit Web Services in Form eines XML-basierten Nachrichtenformates plattform- und programmiersprachenunabhängig abzubilden. Dieses Nachrichtenformat kann in fast beliebige Transportprotokolle (in der Regel HTTP und TCP) eingebunden werden. SOAP stand ursprünglich als Akronym für „Simple Object Access Protocol“, wurde aber im Zuge der Standardisierung beim W3C⁵ durch die schon zu diesem Zeitpunkt weite Verbreitung und in Ermangelung von Alternativen „entakronymisiert“. Bei der Verwendung von SOAP wird ein Request in Form einer definierten XML-Datenstruktur an einen bestimmten Server geschickt. Der Server interpretiert in weiterer Folge die XML-Datenstruktur, erkennt die aufzurufende Operation am Server und ruft diese mit den ebenso in der übergebenen Datenstruktur enthaltenen Parametern auf. Eine SOAP-Nachricht ist ein XML-Dokument, welches aus den nachfolgend beschriebenen drei Teilen besteht (MELZER 2010).

- **SOAP Envelope**, enthält die eigentliche Nachricht und bildet das Wurzelement des XML-Dokuments, in dem die anderen beiden Teile der SOAP-Nachricht gekapselt werden.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-  
envelope">  
  <!-- SOAP Header -->  
  <!-- SOAP Body -->  
</env:Envelope>
```

Listing 2: Elemente einer SOAP Nachricht

⁵ Word Wide Web Consortium, siehe <http://www.w3.org/>

- **SOAP Header**, optionales Element welches nur einmal und dann als erstes Kindelement des SOAP Envelopes vorkommen kann. Der Header einer SOAP-Nachricht wird üblicherweise für die Übertragung von sicherheitsrelevanten Informationen verwendet.
- **SOAP Body**, zwingend erforderliches Element welches die zu übertragende Information der Nachricht enthält (z.B. den Aufruf für die Bereitstellung eines bestimmten Kartenausschnitts).

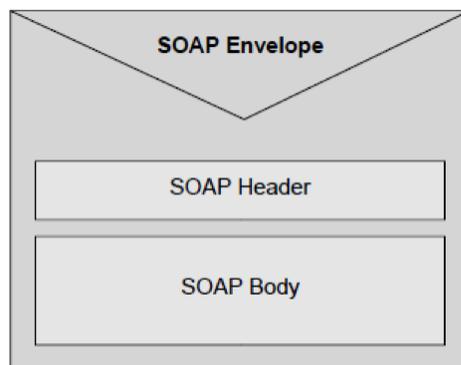


Abbildung 10: Grundlegender Aufbau einer SOAP-Nachricht (Quelle: MELZER 2010)

Die von der Schnittstelle erwarteten, in der SOAP-Nachricht für einen erfolgreichen Aufruf notwendigen Informationen, werden in der Regel mit Hilfe der ebenso auf XML basierenden Web Services Description Language (WSDL) definiert. Dadurch erfährt der Entwickler einer Client-Applikation wie eine SOAP-Nachricht für einen bestimmten Dienst aufgebaut ist und dient demzufolge als eine Art Schnittstellenbeschreibung (CHAPPELL 2009, MELZER 2010).

Der Austausch von in *Oracle* Datenbanken gespeicherten Geodaten mittels SOAP kann mit Hilfe der Technologie *XML DB* erfolgen. Unter Verwendung von *XML DB* können aus der Datenbank native Webservices auf Basis von PL/SQL Prozeduren erstellt werden (WANG 2011). Innerhalb dieser Prozeduren können die Daten entsprechend aufbereitet und in eine XML-Struktur gebracht werden. Bei der

Verwendung von GML kann auf Funktionen von *Oracle Spatial* zur Generierung von GML-Elementen aus `SDO_GEOMETRY` Objekten zurückgegriffen werden.

2.2.2 REST

Im Gegensatz zu SOAP ist REST kein Netzwerkprotokoll bzw. kein XML-basierendes Nachrichtenformat, sondern ein Architekturstil, der im Zuge der Dissertation von Roy Fielding (FIELDING 2000) entwickelt wurde. Er beschreibt darin, dass jede von einem Server bereitgestellte Ressource durch einen eigenen URI (Uniform Resource Identifier) erreichbar ist. Die bekannteste Ausprägung dieses Stils ist das Internet (WWW) selbst mit der Bereitstellung von Homepages und der Adressierung dieser mit Hilfe des URI Subtyps URL (Uniform Resource Locator). Die Adressen einzelner REST-Services werden auch *REST-Endpoints* genannt. Für die Umsetzung des REST-Architekturstils wird die am weitesten verbreitete Web-Technologie HTTP und die darin implementierten Methoden zur Übertragung von Informationen vom Client zum Server (HTTP-GET, HTTP-POST) verwendet. Auch definiert REST keinen Standard für die Repräsentation der Daten, diese ist damit flexibel und erlaubt die Verwendung von eXtended Markup Language (XML), JavaScript Object Notation (JSON) und anderer Formate. TILKOV 2011 bezeichnet das Format JSON vor allem im mobilen Umfeld als zunehmend beliebt und als mittlerweile De-facto-Standard für den Datenaustausch in Web-Applikationen. Für den Austausch von Geodaten gibt es in beiden Formaten offene Standards, die vom OGC definierte Geography Markup Language (GML) und das von einer Internet-Arbeitsgruppe von Entwicklern definierte Format GeoJSON.

Um einen Blick auf die Möglichkeiten zur Bereitstellung von Geodaten mit auf REST basierenden Webservices im Umfeld von *Oracle* Datenbanken zu werfen, mit dem Rapid Application Development Framework *Oracle Application Express* können komplexe Webservices deklarativ erstellt und Ergebnisse zum Beispiel im JSON oder XML Format bereitgestellt werden (SWADENER 2013).

David Chappell kommt bei dem Vergleich der beiden Web Service Zugänge REST und SOAP in seiner Key Note auf dem *ESRI Developer Summit 2009* (CHAPPELL 2009) zu dem Schluss, dass die Verwendung von REST in den meisten Situationen die bessere Wahl ist. Insbesondere für Services, die im Internet veröffentlicht werden sollen, da sie für sehr viele parallele Benutzer und in einem heterogenen Umfeld starke Performanz und Skalierbarkeit bieten. SOAP hat laut Chappell vor allem im firmeninternen Einsatz seine Berechtigung, da es dort mehr auf verteilte Transaktionen zwischen gekoppelten Systemen und die Möglichkeit der Authentifizierung und Autorisierung ankommt.

Nachdem anhand von SOAP und REST zwei allgemeine Architekturmodelle für Web Services vorgestellt wurden, soll in den folgenden beiden Absätzen auf zwei von OGC speziell für den interoperablen Austausch von Geodaten definierten Web Services eingegangen werden.

2.2.3 Web Map Service

Der Web Map Service (WMS) ist ein spezieller Web Service zur Abfrage von georeferenzierten Karten. Diese Karten werden in der Regel in Rasterformaten (PNG, GIF oder JPEG) geliefert, können aber auch in Vektorformaten wie z.B. SVG abgefragt werden. Anfragen erfolgen in Form einer Standard-URL-Adresse. Der OGC Standard definiert die folgenden drei Operationen (Open Geospatial Consortium 2006):

- `GetCapabilities`, liefert den Dienst beschreibende Metadaten
- `GetMap`, zur Abfrage einer anhand der Aufrufparameter definierten Karte
- `GetFeatureInfo`, liefert Informationen zu bestimmten Features auf der Karte

2.2.4 Web Feature Service

Der Web Feature Service (WFS) definiert Schnittstellen für den Zugriff auf und die Manipulation von Geodaten im Vektorformat (sog. Features) über HTTP bzw. Standard-URL-Adressen. Die Daten zu einem Feature werden in Form von XML Formaten wie GML ausgetauscht. Der OGC Standard definiert folgende Operationen (Open Geospatial Consortium 2005):

- `GetCapabilities`, liefert den Dienst beschreibende Metadaten
- `DescribeFeatureType`, liefert Informationen zur Struktur einzelner Feature Typen
- `GetFeature`, dient zur Abfrage der anhand der Abfrageparameter definierten Features
- `GetGmlObject`, ermöglicht die Abfrage einzelner Elemente aus dem GML per XLink.
- `LockFeature`, dient dem Sperren eines Features um zu gewährleisten, dass dieses nicht während einer Transaktion durch einen anderen Benutzer geändert wird.
- `Transaction`, bietet die Möglichkeit Features anzulegen, zu verändern oder zu löschen.

Die Abbildung 11 auf der nächsten Seite zeigt den schematischen Aufbau der Verwendung von OGC Webservices. Dabei greift eine Client Applikation über das Internet oder ein lokales Netzwerk auf die von einem Map Server bereitgestellten Endpunkte zur Nutzung des WMS oder WFS zu. Der Map Server bezieht die angefragten Daten in der Regel von einer Datenbank und liefert das Ergebnis in dem angeforderten Format an die Client Applikation zurück.

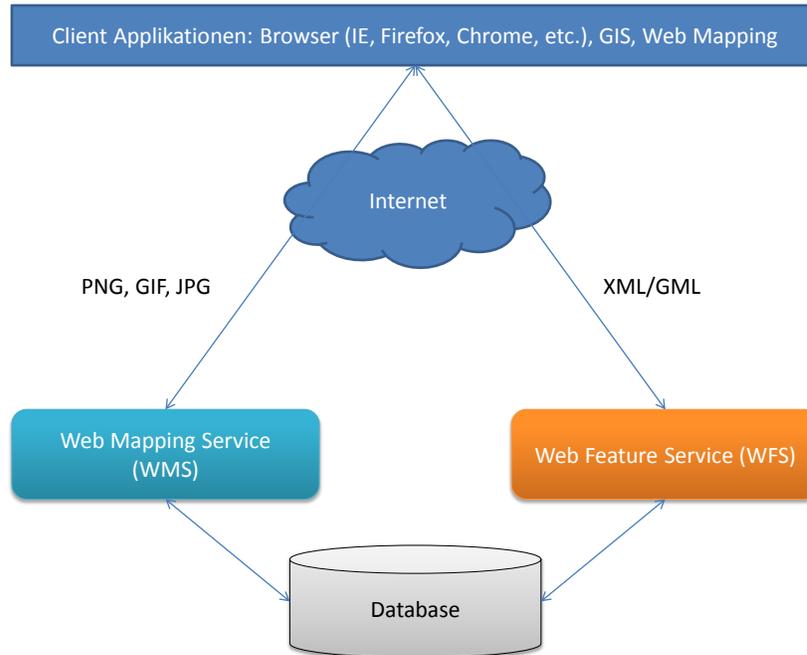


Abbildung 11: Schematische Darstellung der OGC Webservices (Quelle: eigene Darstellung)

Nachdem in diesem Kapitel die Grundlagen von Web Services erläutert wurden, soll im nächsten Kapitel auf eine Methode zur Umwandlung von Adressdaten zu Koordinaten– der sog. Geokodierungsdienst oder Geocoder – näher eingegangen werden. Das Kapitel der Web Services abschließend und überleitend zum Thema der Georeferenzierung, zeigt Abbildung 12 auf der nächsten Seite eine Übersicht über die Verteilung der verwendet Protokolle und Datenformate der im API-Verzeichnis ProgrammableWeb⁶ aufgelisteten 56 Geocoding APIs. Dabei wird die Wichtigkeit der beiden in diesem Kapitel vorgestellten Protokolle SOAP und REST in diesem Umfeld ersichtlich.

⁶ ProgrammableWeb, siehe <http://www.programmableweb.com/>

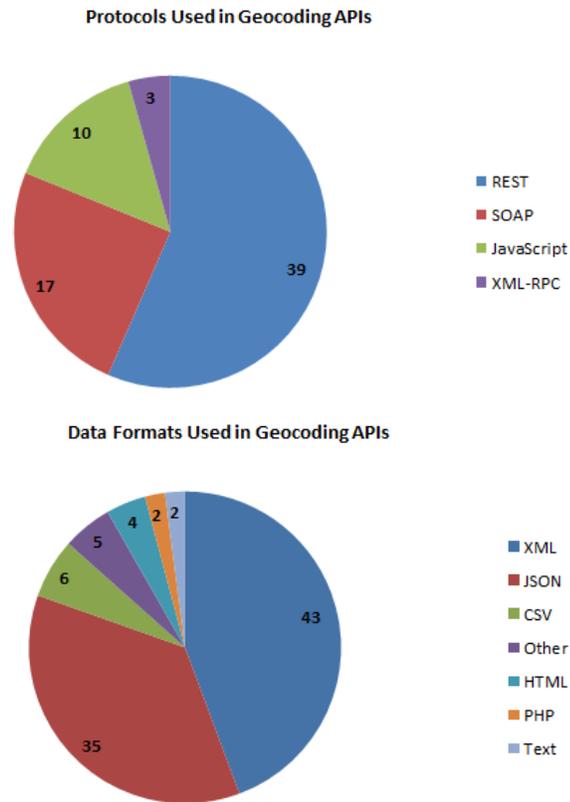


Abbildung 12: In Geocoding APIs verwendete Protokolle und Datenformate (Quelle: SANTOS 2012)

2.3 Georeferenzierung

Im Mittelpunkt eines geoCRM steht die Frage “Wo befindet sich der Kunde?”. Um diese und vor allem davon abgeleitete Fragestellungen – z.B. „Welche Kunden befinden sich in einem Umfeld von 100 km um den aktuellen Standort?“ – zufriedenstellend beantworten zu können, bedarf es eines absoluten, koordinatenbasierenden Referenzsystems. Unter Zuhilfenahme eines solchen Referenzsystems ist es möglich, jeden Ort auf der Welt exakt zu definieren, mit anderen Worten zu georeferenzieren. Das in diesem Umfeld am häufigsten zum Einsatz kommende Referenzsystem ist das World Geodetic System 1984 (WGS 84), welches auch von den globalen Navigationssatellitensystem GPS (Global Positioning System) verwendet wird (REHRL 2010b). Der Prozess der Daten auf der Welt verortet

bzw. technischer ausgedrückt ihnen Koordinatentupel zuweist, wird allgemein als Georeferenzierung bezeichnet. Werden nun speziell Adressen in Koordinaten umgewandelt, kommt ein Teilbereich der Georeferenzierung, die Geokodierung, zur Anwendung.

2.3.1 Geokodierung – von der Adresse zur Koordinate

Um den indirekten Raumbezug vorhandener Adressdaten für die Nutzung als Grundlage zur Geovisualisierung oder für räumliche Analysen nutzen zu können, ist es notwendig, den einzelnen Adressdatensätzen direkten Raumbezug in Form von Realweltkoordinaten zuzuweisen. Der in Abbildung 14 veranschaulichte Vorgang der Überführung der sekundären Metrik (Adresse) in die primäre Metrik (Koordinate) wird Geokodierung genannt (BILL und ZEHNER 2001, Wikipedia 2013).

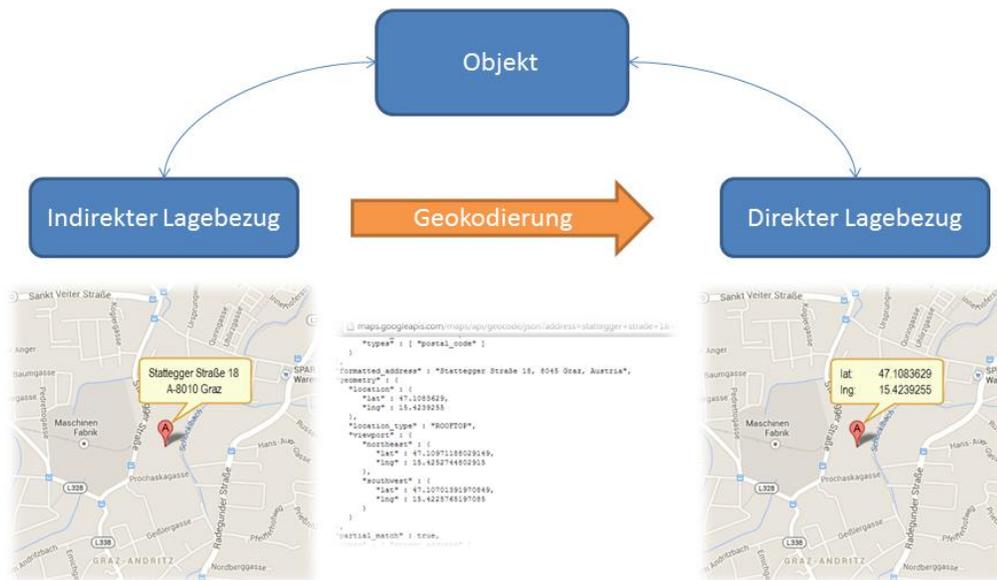


Abbildung 13: Direkter und indirekter Lagebezug (Quelle: eigene Darstellung)

Der im Grunde einfach klingende Prozess der Geokodierung führt durch die Georeferenzierung zu einer Veredelung der Daten und eröffnet eine neue Dimension

möglicher Auswertungen und Analysen, die räumliche. Dass der Geokodierungsprozess alles andere als einfach ist, zeigt der folgende Absatz, in dem die einzelnen Schritte des Prozesses und die dabei auftretenden Herausforderungen dargestellt werden.

Geokodierungsprozess

Das Verständnis für die Abläufe innerhalb des Geokodierungsprozesses ist die Voraussetzung zur Gewinnung möglichst akkurater Ergebnisse durch die im Kapitel 5.2.4 beschriebene Aufbereitung vorhandener Adresdaten. Der Prozess der Geokodierung lässt sich - wie in Abbildung 14 dargestellt - in fünf Prozessschritte unterteilen, welche in weiterer Folge näher betrachtet werden sollen.



Abbildung 14: Schritte des Geokodierungsprozesses (Quelle: eigene Darstellung)

Geoparsing und Strukturierung der Adresse

Im ersten Schritt wird die entgegengenommene Adresse geparkt und in einzelne Komponenten - wie z.B. Straße, Postleitzahl und Ort - zerlegt. Ziel des Parsens ist die Aufbereitung der semi-strukturierten Eingangsadresse in die von dem folgenden Prozessschritt benötigte, strukturierte Form (BEHR 2010). Eine Herausforderung des Geoparsings ist dabei das Erkennen der verschiedenen Adresskomponenten, unter Berücksichtigung der unterschiedlichen sprach- und vor allem auch landes-spezifischen Reihenfolge. Weitere, potentiell auftretende Hindernisse werden im folgenden Absatz nach DING et al. 2000 (zitiert von BEHR 2010) beschrieben, siehe eine Übersicht in Abbildung 15.

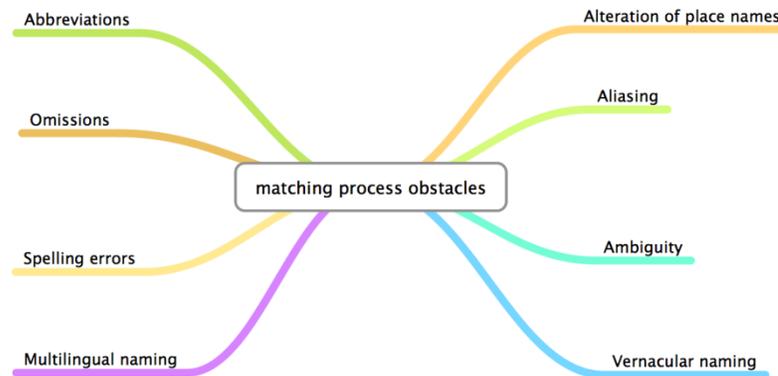


Abbildung 15: Probleme während des Matchings (Quelle: eigene Darstellung nach DING et al. 2000)

- **Änderung von Ortsnamen über die Zeit**

Ortsnamen können sich – meist aus politisch motivierten Gründen – ändern. Ein Beispiel dafür wäre die Umbenennung von „Leningrad“ bzw. „Petrograd“ in „Sankt Petersburg“ oder von „Bombay“ in „Mumbai“.

- **Aliasing**

Darunter wird die Verwendung mehrerer Namen für einen Ort sowohl auf regionaler Ebene, als auch in unterschiedlichen Sprachen verstanden. Zum Beispiel die Bezeichnung der Stadt „Lissabon“ auf Deutsch, „Lisbon“ auf Englisch und „Lisboa“ auf Portugiesisch.

- **Ambiguität**

Ein Ortsname bezieht sich auf Orte in unterschiedlichen Ländern. Ein Beispiel hierfür wäre „Saint Petersburg“ für eine Stadt in Russland sowie Städten in den US-Bundesländern Florida und Pennsylvania.

- **Umgangssprachliche Benennung**

Leben in einer Region mehrere, unterschiedliche Dialekte sprechende Volksgruppen, gibt es für einzelne Orte eventuell unterschiedliche Namen. In Abhängigkeit des im

Süden von Frankreich gesprochenen regionalen Dialekts wird zum Beispiel „Hyères“ auch „Iero“ oder „Ieras“ genannt.

- **Mehrsprachige Benennung**

In Zwei- oder Mehrsprachigen Regionen wird der Matching-Prozess noch weiter erschwert. So werden in manchen Städte der Schweiz Straßennamen sowohl auf Deutsch als auch auf Französisch angegeben. Zusätzlich sind in den beiden Sprachen die oben erwähnte Reihenfolge der Adresskomponenten unterschiedlich: bei deutschen Adressen steht die Straße gefolgt von der Hausnummer, wohingegen bei französischen Adressen die Hausnummer der Straße vorangestellt wird.

- **Rechtschreibfehler**

Die Rechtschreibung der übergebenen Adresse kann sich von der im Referenzdatensatz unter anderem durch fehlende Akzente oder diakritische Zeichen unterscheiden. Zusätzlich können einzelne Buchstaben vergessen oder Namen fehlerhaft geschrieben werden.

- **Abkürzungen**

Viele der einzelnen Komponenten einer Adresse, z.B. die Straßenart, werden oft in Form von Abkürzungen in verschiedenen Schreibweisen angegeben. Im Englischen könnte „Avenue“ als „AV“ oder „AVE“ abgekürzt aber ebenso auch ausgeschrieben werden.

Wie diese Punkte zeigen, ist die Geokodierung – vor allem auf internationaler Ebene – eine Herausforderung, mit der nicht alle Anbieter von Geokodierungslösungen oder –diensten auf gleich gute Weise zurecht kommen. Ein wesentlicher Faktor für das erfolgreiche Geokodieren sind daher qualitativ hochwertige Adressdaten. Nationale Post-Unternehmen und Unternehmen im Bereich des Geomarketings bieten Dienstleistungen zur Adressdatenpflege und Geokodierung von Adressen auf vorwiegend nationaler Ebene.

Matching der Adresse

Im nächsten Schritt wird anhand der nun standardisierten Adresse die Referenzdatenbank durchsucht, mögliche Ergebniskandidaten bestimmt und bewertet. BEHR 2010 unterscheidet zwei grundlegende Klassen von Matching-Algorithmen: solche die auf Äquivalenz zweier Strings prüfen und solche die deren Ähnlichkeit anhand von Rankingmethoden bewerten.

Im einfachsten Fall wird dabei der String der standardisierten Eingangsadresse mit allen Adressen der Referenzdatenbank auf Übereinstimmung hin überprüft. Dieses Verfahren ist das für die im obigen Absatz beschriebenen Herausforderungen und Hindernisse anfälligste, da ein einfacher String-Vergleich schon bei den kleinsten Unterschieden kein Ergebnis liefert (DIOMIN 2010). Um dieses Problem zu umgehen, gibt es eine Vielzahl an Algorithmen um den Vergleich von Strings zu optimieren, WHITE 2005 gibt einen guten Überblick über die unterschiedlichen Ansätze, auf die an dieser Stelle nicht näher eingegangen werden soll.

Der String-Vergleich wird natürlich nicht ausschließlich und alleine für den gesamten übergebenen Adressstring durchgeführt, sondern für dessen einzelnen Komponenten, was mehrere Vorteile mit sich bringt: zum einen kann die Suche in der Referenzdatenbank dahingehend optimiert werden, dass vorab eine Einschränkung der zu durchsuchenden Datensätze zum Beispiel auf Staaten, Bundesland oder Bezirk vorgenommen wird. Zum anderen ermöglicht es einen mehrstufigen Matching-Prozess, der im Falle von fehlerhaften Adressen oder mangelnder Detailgenauigkeit der Referenzdatenbank auf eine höhere Ebene zurückfallen kann. Wird zum Beispiel eine Straße nicht gefunden, beinhaltet das Ergebnis zumindest den Mittelpunkt des Bezirkes oder der Stadt.

Das Ergebnis des Matching-Prozesses besteht demnach aus einem oder mehreren Ergebniskandidaten inklusive deren Ranking und Koordinatentupel, oftmals um einen Hinweis auf die Qualität des Ergebnisses im Hinblick auf die Granularität und der

standardisierten, in deren Komponenten aufgeteilten Adresse erweitert. Letzteres fällt sozusagen als Nebenprodukt des Geokodierungsprozesses an und kann zur Steigerung der Qualität von Adressdatensätzen verwendet werden. Wie das Ergebnis des Matchin-Prozesses in das des Geokodierungsprozess Einzug findet, wird im nächsten Absatz erläutert.

Ergebnis

Das Ergebnis des vorangegangenen Prozessschrittes wird abschließend noch entsprechend Aufbereitet und in ein für den Client verständliches Format gebracht. Übliche Formate in diesem Umfeld beinhalten HTML, XML und verwandte Formate wie z.B. GML oder KML sowie JSON bzw. GeoJSON (WAGNER et al. 2009). Beispiele für unterschiedliche Geokodierungsergebnisse sind in Anhang A ersichtlich.

Nachdem in diesem Kapitel der Prozess der Geokodierung betrachtet wurde, soll im folgenden Abschnitt, das Kapitel der Georeferenzierung abschließend, ein Überblick über die wichtigsten im Internet verfügbaren – kommerziellen sowie öffentlichen – Geokodierungsdienste und deren Nutzungsbedingungen gegeben werden. Wie ein solcher Dienst mittels Web Service konsumiert und in einer Web-Anwendung mit Hilfe von JavaScript zum Einsatz kommen kann, wird dann anhand der Geocoding API von *Google Maps* in Kapitel 5.4 gezeigt.

2.3.2 Geokodierungsdienste

Im Internet stehen Entwicklern eine Vielzahl von Geokodierungsdiensten zur Verfügung. Einen guten Überblick über die verschiedenen, verfügbaren Dienste gibt TAMU 2013 und geht dabei auch auf die durch die jeweiligen Nutzungsbedingungen definierten Limitierungen der Nutzung, das verwendete Referenzdatenmaterial und etwaige mit der Nutzung verbundenen Kosten ein. In diesem Kapitel werden drei Kategorien von Geokodierungsdiensten vorgestellt und jeweils Beispieldienste näher

betrachtet. Die erste Kategorie umfasst Geokodierungsdienste kommerzieller Anbieter – unabhängig davon ob diese „frei“ (im Sinne von kostenlos) nutzbar sind oder nicht – mit der bekannten und weit verbreiteten *Google Geocoding API* als Beispiel. Die zweite Kategorie beinhaltet Dienste die Open Data bzw. Volunteered Geographic Information (VGI) als Referenzdatenmaterial verwenden und meist auch als Open-Source Produkte entwickelt wurden. Als Beispiel für diese Kategorie dient der auf *OpenStreetMap* Daten basierende Dienst *Nominatim*. In die dritte Kategorie fallen Frameworks zur Erstellung von Diensten auf Basis von VGI (z.B. *OpenStreetMap*, *Geonames*) oder kommerzieller Daten (z.B. *NAVTEQ*).

Kommerzielle Geokodierungsdienste

Geokodierungsdienste werden von allen großen Suchmaschinenanbietern u.a. *Yahoo*, *Microsoft Bing* und *Google* sowie Geodatenanbieter wie z.B. *NAVTEQ*, *TomTom* bzw. *Tele Atlas* angeboten. All diese Dienste bieten in der Regel ein gewisses Kontingent an freien Geokodierungsanfragen pro Zeiteinheit (meistens pro Tag) an und regeln die Nutzung der Dienste und der davon bezogenen Daten mit Hilfe von Nutzungsbedingungen (eng. Terms of Service). Über das freie Kontingent bzw. die erlaubte freie Nutzung hinausgehende Verwendung der Dienste und der bezogenen Daten, werden dann auf Basis unterschiedlicher Verrechnungsmodelle abgerechnet.

Die Nutzungsbedingungen der *Google Maps API* erlauben 2 500 Geokodierungsanfragen pro 24 Stunden Periode (100 000 bei *Google Maps API for Business* Kunden) und schreibt die Nutzung in Verbindung mit *Google Maps* vor; Anfragen deren Ergebnis nicht auf einer *Google Map* dargestellt werden sind also nicht erlaubt, ein Zwischenspeichern der Daten zur Entlastung des Dienstes ausgenommen (Google 2013c, Google 2013b). Der Geokodierungsdienst von *Google* unterstützt aktuell 209 Länder (Google 2009) und kann als Web Service und mit Hilfe einer JavaScript API genutzt werden. Zusätzlich gibt es noch *Google Maps SDKs* für iOS und Android, in denen jeweils auch eine Klasse zur Geokodierung enthalten ist (Google 2013a).

Open Data Geokodierungsdienste

Unter dieser Kategorie werden alle Dienste zusammengefasst, die als Referenzdatenmaterial „offene Daten“ verwenden. Mit *Open Data* (OD) sind in diesem Zusammenhang sowohl offene Regierungsdaten (*Open Government Data* - OGD) als auch von privatwirtschaftlich agierenden Unternehmen, Non-Profit-Einrichtungen, Hochschulen etc. der Öffentlichkeit zur freien Nutzung zur Verfügung gestellte Daten (LUCKE und GEIGER 2010). Zusätzlich umfasst der Begriff damit auch jegliche Art von *Volunteered Geographic Information* wie von GOODCHILD 2007 definiert, also Projekte wie *OpenStreetMap*, *OpenGeocoding* oder *Geonames*.

Auf Open Data basierende Geokodierungsdienste wurden meist auch als Open-Source Projekte und/oder von Hochschulen entwickelt. Die Hardware auf denen diese Dienste laufen ist daher nicht vergleichbar mit denen kommerzieller Anbieter, was sich in den stark eingeschränkten Nutzungsbedingungen widerspiegelt. Beispiele für solche Dienste/Produkte wären *Nominatim*⁷ (basiert auf *OpenStreetMap* Daten), *OpenGeocoding*⁸ oder *Gisgraphy*⁹ (basiert auf *OpenStreetMap* Daten in Kombination mit *Geonames* Daten). Die öffentlich Zugänglichen Web Services sind für Testzwecke, nicht aber für den produktiven Betrieb geeignet. Da aber sowohl Daten als auch Software frei, sowie in der Regel gut dokumentiert, verfügbar sind, können Geokodierungsdienste z.B. von Unternehmen für den internen Gebrauch bereitgestellt werden, worauf im nächsten Absatz näher eingegangen werden soll.

Do it yourself Geokodierungsdienste

Die letzte Kategorie beinhaltet die unterschiedlichsten Varianten zur Erstellung von Geokodierungsdiensten innerhalb des eigenen Technologiestacks. Dazu sind zwei Komponenten notwendig: das Referenzdatenmaterial und die als Web Service zur Verfügung gestellte Geokodierungssoftware. Das Referenzdatenmaterial kann aus den im letzten Absatz erwähnten offenen Datenquellen (z.B. *OpenStreetMap*)

⁷ Nominatim, siehe <http://wiki.openstreetmap.org/wiki/Nominatim>

⁸ OpenGeocoding, siehe <http://opengeocoding.org/>

⁹ Gisgraphy, siehe <http://gisgraphy.com/>

stammen oder von kommerziellen Geodatenanbietern (z.B. *TomTom*¹⁰) zugekauft werden. Auf Basis dieser Daten können vorhandene Geokodierungssoftware (wie beispielsweise der in *Oracle Spatial* enthaltene Geocoder) oder Open-Source Software (u.a. *Gisgraphy*, *Nominatim*) zum Einsatz kommen.

Die Verwendung der Geokodierungsdienste ist unabhängig von der Kategorie in die diese fallen, sehr einfach möglich. Die meisten können über REST abgefragt werden und liefern für einen übergebenen Adressstring ein Koordinatentupel – je nach Dienst zusätzlich auch die standardisierte Adresse – als Ergebnis in unterschiedlichen Formaten zurück. Eine beispielhafte Anfrage an den Geocoder von *Google* ist inkl. Ergebnis in Anhang A ersichtlich.

Nachdem in diesem Kapitel ausgeführt wurde, wie aus Adressdaten „echte“ Geodaten werden, soll im folgenden Kapitel auf die Geovisualisierung dieser Daten in Form von Web Mapping eingegangen werden. Es werden dabei zwei unterschiedliche Zugänge im Hinblick auf die Softwarearchitektur betrachtet und mit der Beschreibung von Beispielen aus dem Bereich der Mapping APIs und der Web Mapping Server verdeutlicht.

2.4 Web Mapping

Nach einer Definition von NEUMANN 2011 ist Web Mapping „*the process of designing, implementing, generating and delivering maps on the World Wide Web*“. Im Zuge dieser Arbeit wird Web Mapping weniger als Prozess sondern aus einer stärker technologischen Sichtweise gesehen, und als ein Set von Werkzeugen zur Bereitstellung von interaktiven, webbasierten Karten verstanden. In diesem Sinne werden im nächsten Abschnitt zwei grundsätzliche Architekturmodelle vorgestellt, gefolgt von einer näheren Betrachtung dieser.

¹⁰ TomTom Geospatial for Oracle, siehe http://www.tomtom.com/en_gb/licensing/products/maps/geospatial-oracle/

2.4.1 Architekturmodelle

Im Bereich des Web Mappings lassen sich zwei Architekturmodelle unterscheiden, bei einem geschieht die Geovisualisierung vornehmlich auf Seite des Clients, wobei auf eventuelle Datenquellen direkt zugegriffen bzw. diese direkt abgefragt werden. Beim zweiten Architekturmodell kommt zusätzlich eine Middleware in Form eines Web Mapping Servers zum Einsatz, der sich um die Beschaffung der Daten und die Geovisualisierung dieser kümmert. In den nächsten beiden Absätzen sollen diese beiden Architekturmodelle vorgestellt werden.

2.4.2 Clientseitiges Web Mapping

Beim clientseitigen Web Mapping kommen hauptsächlich JavaScript APIs zum Einsatz, welche für die Darstellung der Basiskarte und das Rendern der Overlays im Browser sorgen. Diese Overlays können, je nach verwendeter API, aus unterschiedlichen Quellen stammen. Typische Vertreter des clientseitigen Web Mappings wären die proprietären APIs von *Google Maps* und *Mircosoft Bing Maps*, die Open Source Mapping Bibliothek *OpenLayers* oder die *ArcGIS JavaScript API*. Bevor auf diese näher eingegangen wird, zeigt die folgende Abbildung den schematischen Aufbau des clientseitigen Web Mappings. Der Browser bezieht die Basiskarte dabei direkt von z.B. Google Maps. Die darauf darzustellenden Daten werden mittels AJAX über den Webserver von der Datenbank abgefragt und im Browser von der verwendeten Mapping API gerendert.

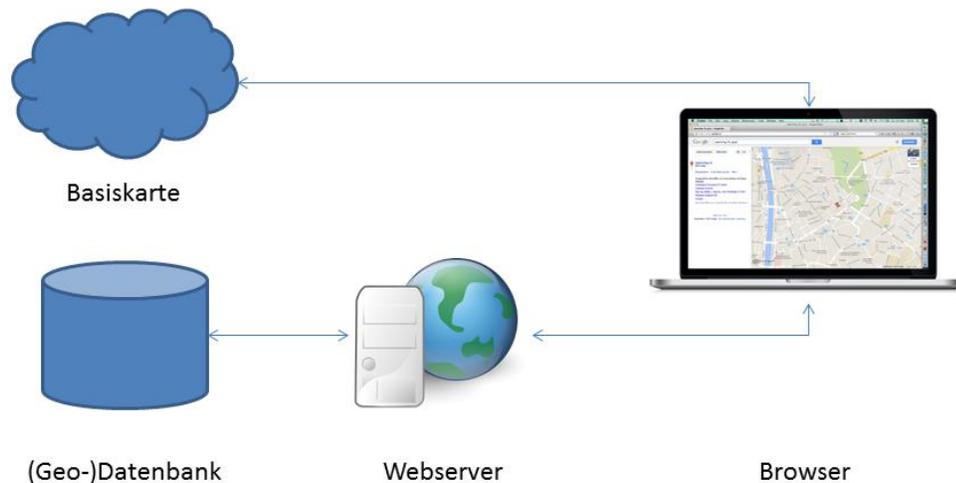


Abbildung 16: Clientseitiges Web Mapping (Quelle: eigene Darstellung)

An dieser Stelle soll jeweils als Beispiel für eine proprietäre Mapping API, die von *Google Maps* und für eine Open Source Mapping Bibliothek *OpenLayers* vorgestellt werden. Das Ergebnis, die im Browser dargestellte Karte, ist in beiden Fällen sehr ähnlich. Worin sich die beiden APIs voneinander unterscheiden, wird in den folgenden beiden Absätzen beschrieben.

Google Maps API

Die *Google Maps API* ist eine der am weitesten verbreiteten Mapping API und die am häufigsten verwendete API für die Erstellung von Mashups. Nach einer Statistik von Programmableweb.com¹¹ verwenden im November 2013 knapp 40 Prozent aller Mashups die *Google Maps API*. Ein Grund dafür ist sicherlich die leichte Anwendung der API für die einfache Darstellung von Standorten, Polylinien und Polygonen, sowie die Unterstützung von mobilen Endgeräten. Die Einbindung von Daten ist in den Formaten KML und GeoRSS möglich und es gibt ausgereifte Lösungen von Drittherstellern, die sich um das Clustering von Pins (siehe Abbildung 17) kümmern (SVENNERBERG 2010). Die Nutzung ist kostenlos möglich, solange die Karte öffentlich frei zugänglich ist und das Limit von 25 000 Kartenaufrufen nicht an

¹¹ ProgrammableWeb, siehe <http://programmableweb.com/apis>

90 aufeinanderfolgenden Tagen überschritten wird. Für die firmeninterne Nutzung – wie es bei einem CRM-System der Fall ist – ist es notwendig eine *Google Maps API for Business* Lizenz zu lösen, die aktuell 8 000 Euro pro Jahr kostet (Google 2013b).

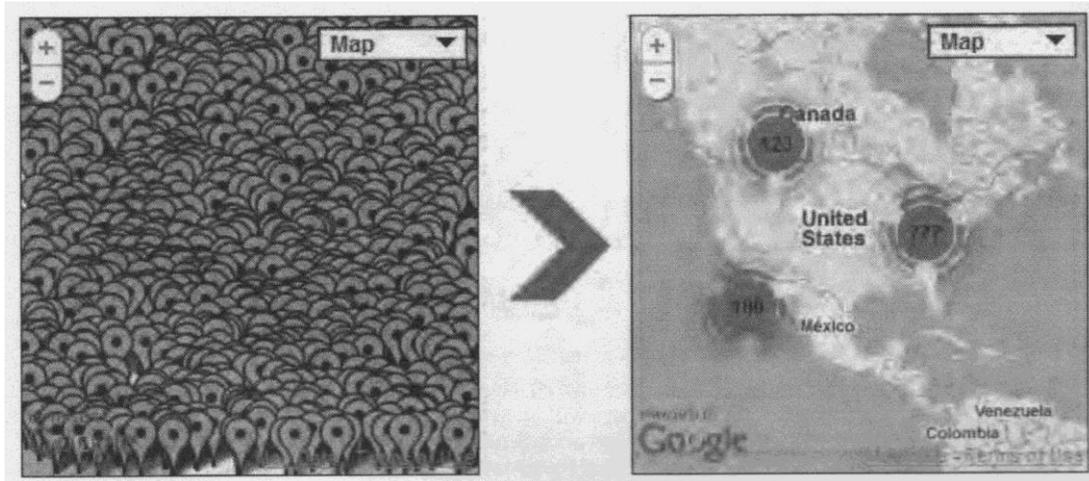


Abbildung 17: Unterscheid zwischen 1 000 Marker-Pins und Markercluster (Quelle: SVENNERBERG 2010)

OpenLayers

OpenLayers ist eine Open-Source JavaScript Bibliothek zur Entwicklung von Web Mapping Anwendungen, welche unter der FreeBSD Lizenz¹² veröffentlicht wurde. Mit *OpenLayers* können eine Vielzahl an unterschiedlichen Basiskarten eingebunden werden (u.a. *Google Maps*, *Bing Maps*, *OpenStreetMap*) und mit Overlays aus zahlreichen, unterschiedlichen Quellen (u.a. WMS, WFS, JSON, GML, *ArcGIS Server*) kombiniert werden (JANSEN und ADAMS 2010). Eine mobile Nutzung ist in der aktuellen Version 2.13 zwar grundsätzlich möglich, im Vergleich mit der *Google Maps API* aber noch sehr verbesserungswürdig. *OpenLayers* ist eine sehr ausgereifte Bibliothek mit unzähligen Features und einer großen Community an Entwicklern und Benutzern. Allerdings hat es in der Benutzung einen schon etwas veralteten Charme und kann in diesem Bereich nicht mit den neueren Entwicklungen mithalten. In diese Kerbe schlagen einerseits die sich in Entwicklung befindlichen Bibliotheken *Leaflet*

¹² FreeBSD Lizenz, siehe <http://www.freebsd.org/copyright/freebsd-license.html>

und die dritte Version von *OpenLayers*, die beide verstärkt auf HTML5 und CSS3 aufbauen. Der große Vorteil einer Bibliothek wie *OpenLayers* ist die Unabhängigkeit von proprietären APIs und deren Nutzungsbedingungen bei gleichzeitiger Möglichkeit proprietäres Kartenmaterial als Basiskarten einzubinden (HAZZARD 2011).

In manchen Fällen ist es notwendig oder zumindest ratsam, die Erstellung der Karten und die Kommunikation mit der räumlichen Datenbank einem darauf spezialisierten Serverdienst zu überlassen. Dadurch erhöht sich zwar die Komplexität einer Lösung, dafür ist neben anderen Vorteilen deren Wiederverwertbarkeit gesichert. Auf dieses sogenannte serverseitige Web Mapping soll im nächsten Absatz näher eingegangen werden.

2.4.3 Serverseitiges Web Mapping

Serverseitiges Web Mapping bedeutet, dass die Komposition der Karte auf einem Web Map Server passiert und den Clients in unterschiedlichen Formaten zur Konsumation angeboten wird. Der Web Map Server fungiert dabei als Mittelschicht zwischen dem Client und der Geodatenbank und übernimmt als solche die Kommunikation mit ihr, wie in der Abbildung 18 auf der nächsten Seite dargestellt. Mittels Web Map Server erstellte Lösungen können dadurch im Vergleich zu clientseitigen Web Mapping leichter auch in anderen Applikationen bzw. für andere Services wiederverwendet werden. Karten die in der Erstellung aufwändig sind, können durch Web Map Server zentral, auf leistungsstarker Server Hardware gerendert und für zukünftige Nutzung zwischengespeichert werden. Weitere Anwendungsgebiete wären das Orchestrieren von Geodaten aus den verschiedensten Datenquellen, die Bearbeitung von Geodaten, die Geoprozessierung sowie der hochformatierte Druck von Karten. Beispiele für Web Mapping Server Produkte wären, wohlgemerkt mit teils sehr unterschiedlichen Leistungsumfang, das

proprietäre Produkt *ArcGIS Server*¹³ von *ESRI* und die zwei bekannten Open-Source Produkte *MapServer*¹⁴ und *GeoServer*¹⁵.

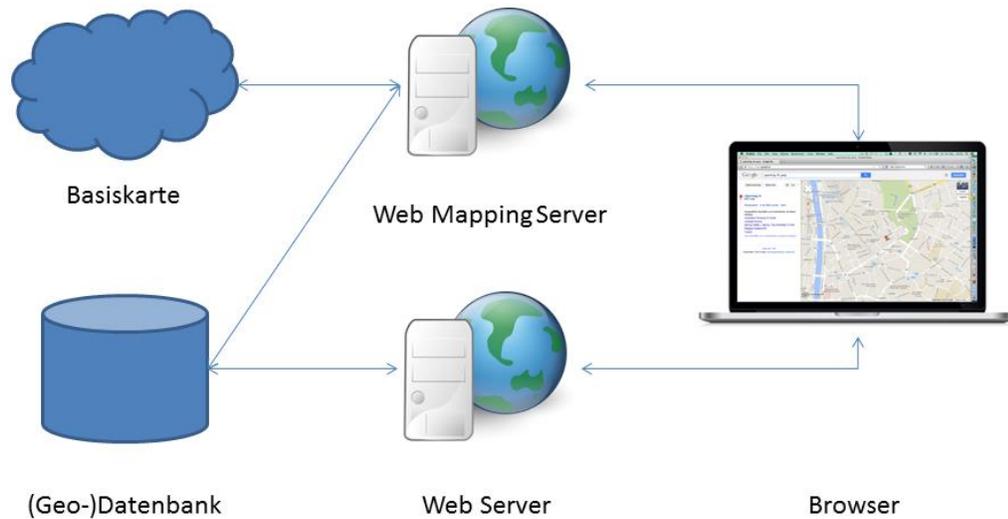


Abbildung 18: Serverseitiges Web Mapping (Quelle: eigene Darstellung)

Zur clientseitigen Einbindung der von Web Map Servern generierten Karten kommen meist die im letzten Absatz vorgestellten Web Map APIs zum Einsatz. Dabei können die generierten Karten als Basiskarte oder als Overlay dargestellt werden. Informationen zu einzelnen Features können vom Web Map Server als auch direkt von der Datenbank abgefragt und angezeigt werden.

In den letzten Kapiteln wurden die technologischen Grundlagen der Integration von GIS-Technologien im Kundenbeziehungsmanagement behandelt. Im nächsten Kapitel soll nun ein mögliches Anforderungsprofil eines geoCRM diskutiert werden.

¹³ ArcGIS Server, siehe <http://www.esri.com/software/arcgis/arcgisserver>

¹⁴ MapServer, siehe <http://mapserver.org/>

¹⁵ GeoServer, siehe <http://geoserver.org/>

3 Anforderungsprofil geoCRM

In diesem Kapitel sollen die Anforderungen an ein geoCRM System diskutiert werden. In Anlehnung an das EVA-Prinzip (Eingabe-Verarbeitung-Ausgabe), lassen sich die Aktivitäten der Benutzer bzw. in weiterer Folge die Funktionalitäten des Systems in drei Bereiche gliedern:

1. Funktionalitäten die den Benutzer bei der **Eingabe** und Erfassung von z.B. Kundenstandorten unterstützen und damit u.a. einen Beitrag zur Steigerung der Datenqualität leisten.
2. Die **Verarbeitung** im weitesten Sinne umfasst die (räumliche) Filterung der Daten im Zuge der „Geosuche“ und der Vorbereitung der geovisualisierten Ausgabe.
3. Die **Ausgabe** erfolgt dabei sowohl in Form statischer Karten innerhalb eines Georeports, als auch in Form interaktiver, dynamischer Karten zur Analyse.

Die Abbildung 19 auf der nächsten Seite gibt einen Überblick über die unterschiedlichen Funktionsbausteine des geoCRM. Auf der rechten Seite der Mind Map sind Funktionalitäten der Bereiche Eingabe und Verarbeitung aufgelistet, auf der linken Funktionalitäten des Bereichs Ausgabe. Daran anschließend zeigt die Abbildung 20 in Form eines UML Use Case Diagrammes Aktivitäten, die die unterschiedlichen Benutzergruppen eines geoCRM – Verkaufsmitarbeiter und Manager – mit Hilfe des Systems ausführen können sollen.

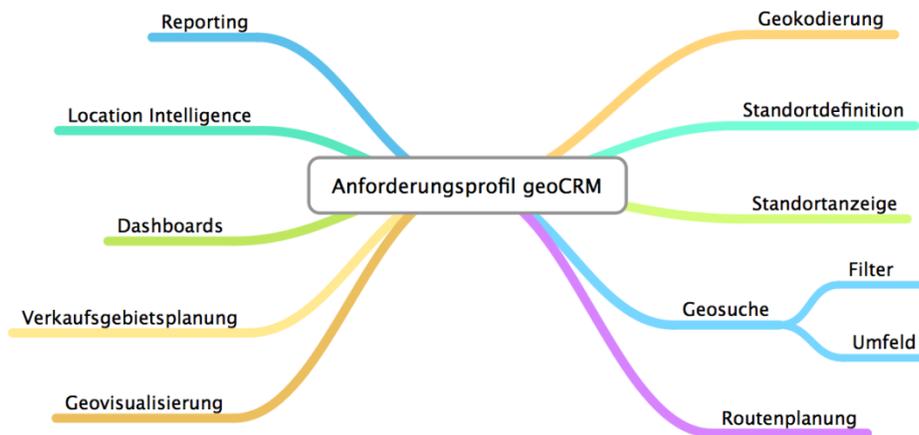


Abbildung 19: Überblick Anforderungsprofil geoCRM (Quelle: eigene Darstellung)

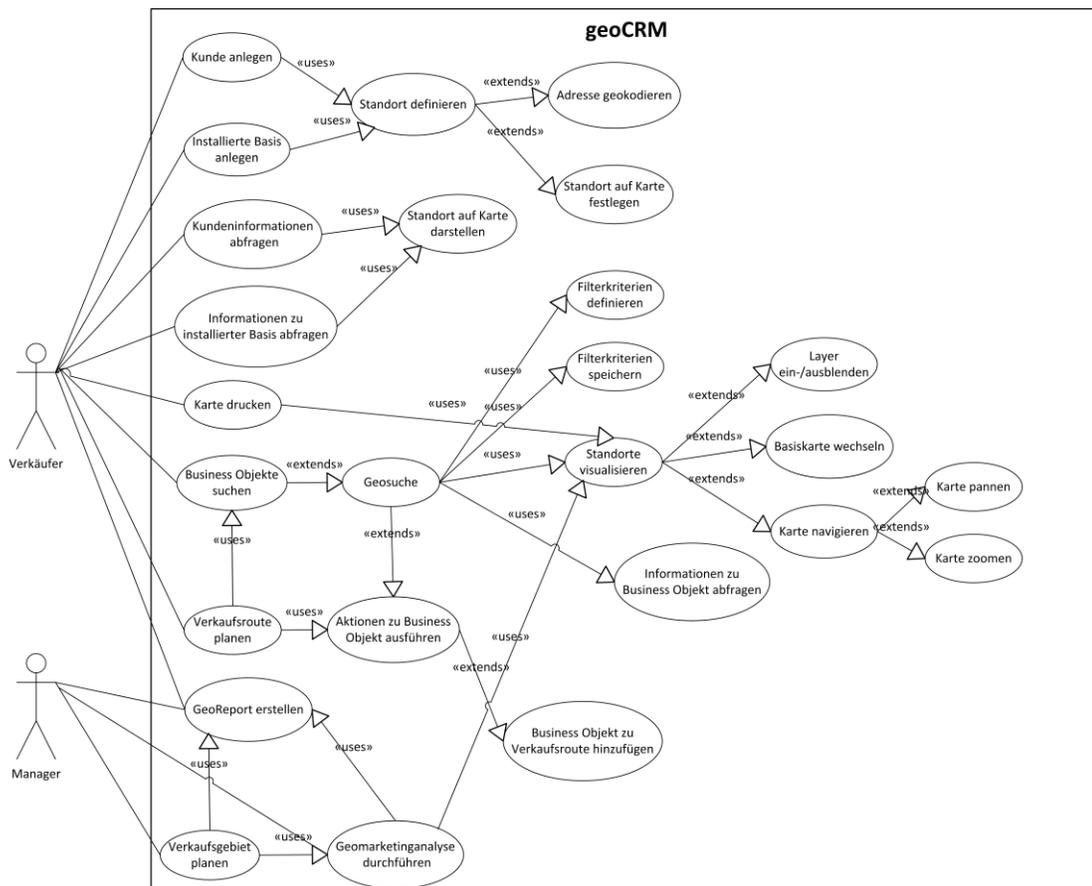


Abbildung 20: Use Case Diagramm geoCRM (Quelle: eigene Darstellung)

Im folgenden Kapitel soll nun auf konkrete Anwendungsgebiete aus dem Anforderungsprofil eingegangen werden, welche die Nutzung eines geoCRM ermöglicht.

4 Anwendungsgebiete geoCRM

Wie im einführenden Kapitel 1.2 beschrieben, befindet sich ein geoCRM-System an der Schnittstelle zwischen Business Mapping, Location Intelligence und Geomarketing. Durch die Betrachtung potentieller Anwendungsgebiete aus eben diesen drei Perspektiven und abschließender, übergreifender Einbeziehung mobiler Technologien, sollen die ersten beiden Forschungsfragen (siehe Kapitel 1.5) beantwortet werden. In diesem Kapitel werden vorbereitende Schritte wie ein entsprechend erweitertes Datenmodell sowie bereits geokodierte Adressdatensätze vorausgesetzt. Die folgende Abbildung gibt einen Überblick über die verschiedenen Anwendungsgebiete von GIS-Technologien im Kundenbeziehungsmanagement.

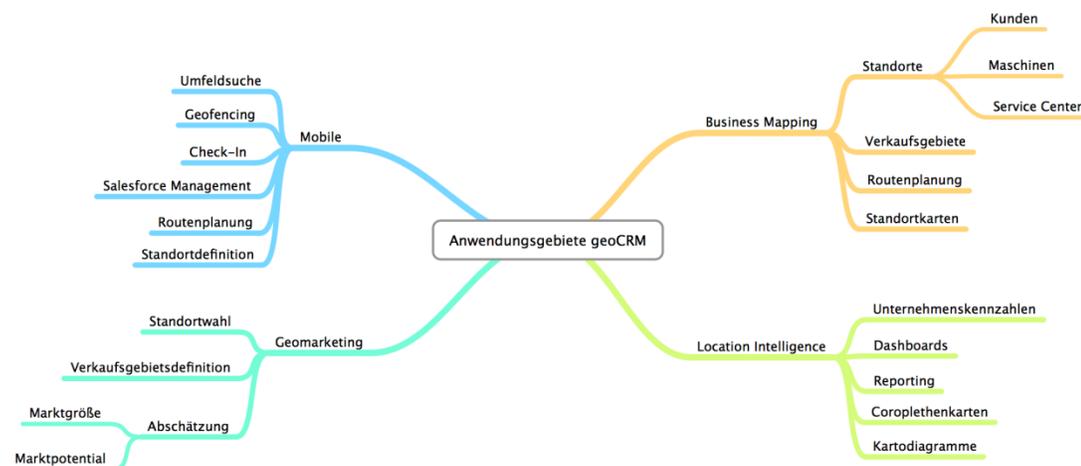


Abbildung 21: Übersicht Anwendungsgebiete geoCRM (Quelle: eigene Darstellung)

4.1 Business Mapping

Die sehr allgemeine Definition von Business Mapping als „*raumbezogene Visualisierung im Zusammenhang mit Business-Anwendungen*“ (CZERANKA 2001) umfasst jegliche Kartendarstellung („raumbezogene Visualisierung“) in CRM-Systemen („Business-Anwendungen“) und überschneidet sich damit per Definition mit den Bereichen Location Intelligence und Geomarketing. In diesem Kapitel wird unter dem Begriff Business Mapping die Visualisierung von Standorten in Form von Positionskarten bzw. Standortkarten nach der Definition von HAKE et al. 2002 verstanden. Aber welche Informationen führen, räumlich visualisiert zur Optimierung von Entscheidungsfindungsprozessen oder von operativen Verkaufsprozessen?

Schon die Darstellung eines Kunden- oder Maschinenstandorts auf einer Karte beschleunigt die Perzeption der räumlichen Lage im Vergleich zur textuellen Adresse. Die besonderen Vorteile der Geovisualisierung kommen bei der Darstellung mehrerer, nach unterschiedlichen Kriterien gefilterter Standorte zu tragen, da dadurch räumliche Zusammenhänge deutlich besser zu erfassen sind als durch die klassische Ansicht in Tabellenform. Ein Beispiel dafür wäre die Visualisierung aller Standorte eines bestimmten Maschinentyps weltweit. Für die operative Ebene des Verkaufs ergeben sich dadurch unter anderen folgende Vorteile:

- Generierung einer graphischen Referenzliste
- Überblick der installierten Basis eines Landes/einer Region
- Visualisierung der räumlichen Verteilung von Maschinentypen, Verkaufschancen oder Kunden inklusive z.B. Statusinformationen
- Basis für das Benchmarking mit Verkäufern und Management
- Visualisierung der Kunden im Umfeld eines geplanten Kundenbesuches
- Räumlich-interaktiver Zugang zu vorhandenen Informationen
- Werkzeug zur Kontrolle der Datenqualität

Auf strategischer Ebene des Managements können durch Geovisualisierung Grundlagen bereitgestellt werden, die zur Optimierung der Entscheidungsfindungsprozesse beitragen können. So können zum Beispiel anhand der Verteilung der Installed Base Entscheidungen über die Standortwahl für Service Center getroffen werden. Des Weiteren liefern die Karten Grundlagen für die Abschätzung der Marktgröße und des Marktpotentials (vor allem auch im Hinblick auf das Servicegeschäft), sowie zur strategischen Ausrichtung auf bestimmte Märkte oder die Definition von Verkaufsgebieten (Überschneidung mit dem Geomarketing). In den meisten Fällen werden für Entscheidungen auf strategischer Ebene jedoch Karten auf Basis von hochaggregierten Unternehmensdaten in Form von thematischen Karten von besonderem Interesse sein, wie sie im nächsten Abschnitt beschrieben werden.

4.2 Location Intelligence

„Location Intelligence is the use of locationally-referenced information as a key input in business decision making. Location intelligence uses GIS tools and techniques to transform and analyse data that becomes valuable information to make more informed and rational business decisions“ (WOLFE und MOON 2011).

Als räumliche Erweiterung von Business Intelligence (BI) ist Location Intelligence (LI) damit im analytisch-explorativen Bereich der Auswertung von Daten angesiedelt. Eine Integration in das Kundenbeziehungsmanagement erfolgt demnach in erster Linie auf Ebene der Daten. Auswertungen von LI berücksichtigen dabei die geographischen Aspekte von KPIs und sonstiger Unternehmenskennzahlen, meist in Form von thematischen Karten (Coroplethenkarten, Kartodiagrammen) in Kombination mit herkömmlichen Diagrammen und Tabellen. Die Kennzahlen werden dabei auf Landes- oder Regionsebene hochaggregiert nach unterschiedlichen Regeln dargestellt, oft auch in Kombination mit Daten aus anderen Quellen wie zum Beispiel einem Data-Warehouse. Neben der interaktiven Analyse ist das Berichtswesen (Reporting) ein wichtiger Bereich von BI und damit von LI. Dabei

werden vordefinierte Berichte zu bestimmten Zeitintervallen automatisch generiert und versendet oder bei Bedarf generiert (sog. Ad-Hoc-Reporting). Business Mapping und Location Intelligence dienen auch als Basis für das Geomarketing.

4.3 Mobile

Durch die Verbreitung von Smartphones und Tablets im privaten Bereich und in Unternehmen, steigt der Bedarf an Business Applikationen für die mobile Verwendung der innerbetrieblichen Informationssysteme (sog. In-House-Apps). Entwicklung und Einsatz mobiler CRM-Systeme (sog. mCRM) werden seit der Jahrtausendwende von einer Vielzahl an Autoren (u.a. WINKELMANN 2012) behandelt. Alle namhaften CRM-Softwareanbieter haben zumindest eine „Companion-App“ in ihrem Portfolio. In diesem Absatz soll daher vornehmlich auf die potentiellen mobilen Anwendungsgebiete des geoCRM eingegangen werden, welche sich natürlich zwangsläufig mit dem Bereich mCRM überschneiden bzw. diesen ergänzen.

Location-Based Services (LBS)

„Location-based services (LBS) provide targeted information to individuals based on their geographic location in real or near-real time, typically through wireless communication networks and clients such as portable computers, personal digital assistants, mobile phones, and in-vehicle navigation systems“ (BRIDWELL und MILLER 2011).

Der besondere Vorteil der Nutzung mobiler Geräte in Business Applikationen wie CRM ist, neben der ubiquitären Verfügbarkeit der Informationen und der Möglichkeit der Einführung mobiler Geschäftsprozesse, die Bereitstellung von ortsbezogenen Daten und Diensten. In der Domäne des geoCRM ergeben sich dadurch klassische Anwendungsgebiete wie z.B. die Umfeldsuche, Routenplanung aber auch jüngere

Anwendungsfelder wie Check-In Dienste oder das im nächsten Abschnitt beschriebene Geofencing.

Geofencing

Der Begriff Geofence beschreibt einen virtuellen Perimeter um bestimmte reale Gebiete; ein virtueller geographischer Zaun. Geofencing wird dabei die Nutzung dieser virtuellen Perimeter zu den unterschiedlichsten Zwecken genannt. Für ein geoCRM lassen sich Geofences nutzen, um Benutzer auf bestimmte Sachverhalte aufmerksam zu machen, in deren Nähe sie sich befinden. So lassen sich um Kundenstandorte virtuelle Perimeter ziehen, innerhalb derer Vertriebsmitarbeiter informiert werden, wenn beispielsweise ein bestimmter Maschinentyp bei einem Kunden installiert ist.

Umfeldsuche

Die Umfeldsuche ist eine der klassischen LBS-Anwendung mobiler CRM-Systeme und daher auch in fast allen verfügbaren mCRM Apps verfügbar. Dabei wird um den aktuellen Standort des Benutzers in Karten- oder Listenform angezeigt, welche Kunden sich innerhalb eines definierten Radius befinden. Eine weitere Variante ist die Anzeige einer definierten Anzahl an nächsten Nachbarn, also z.B. die zehn sich dem aktuellen Standort am nächsten befindlichen Kunden (siehe Abbildung 22). Der Nutzen einer Geovisualisierung des Umfeldes auf Basis von CRM-Daten lässt sich steigern, indem Informationen über den Kunden oder über die bei einem Kunden installierten Maschinen kartographisch aufbereitet dargestellt werden. Ein Beispiel wäre die Verwendung unterschiedlicher Marker für Kunden und Maschinen in Verbindung mit einer Farbkodierung für den Status einer Maschine.

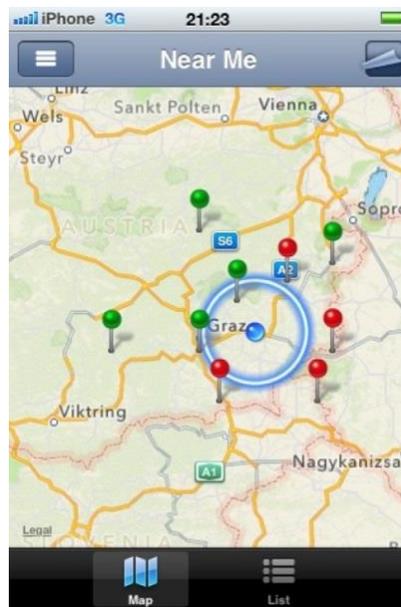


Abbildung 22: Skizze einer mobilen geoCRM Umfeldsuche (Quelle: eigene Darstellung)

Routenplanung

Unter Routenplaner werden Anwendungen verstanden, mit deren Hilfe ein Weg zwischen einem Start- und einem oder mehreren Zielorten gefunden werden kann. Diese Start- und Zielorte werden meist anhand einer Adresse definiert und als Ergebnis die Route zwischen diesen als Karte und/oder als eine Folge sogenannter Turn-by-turn Anweisungen geliefert. Unter Zuhilfenahme eines geoCRM können Verkaufsrouten erstellt werden, deren Start- und Zielorte Kunden- oder Maschinenstandorte sind, wodurch die Routenerstellung vereinfacht werden kann. Die mobile Nutzung ermöglicht das Betrachten von Teilabschnitten einer Route oder die zeitnahe Bearbeitung einer solchen abseits vom Arbeitsplatz. Die Abbildung 23 auf der nächsten Seite skizziert die Routenplanung als möglichen Teil einer mCRM App. Neben der in diesem Absatz besprochenen, operativen und in ein CRM-System integrierten Tourenplanung, gibt es noch die strategische Tourenplanung wie in WINKELMANN 2012 beschrieben.



Abbildung 23: Skizze einer mobilen geoCRM Verkaufsrouten (Quelle: eigene Darstellung)

Check-In Dienste

Seit sozialen Check-In Diensten wie *Foursquare*, *Gowalla* und *Facebook* ist das „Einchecken“ – wenn auch in Europa nicht weit verbreitet – dank der medialen Behandlung für viele Menschen ein Begriff. In Verbindung mit Geofencing können Check-In Dienste genutzt werden um bei einem Kunden „einzuchecken“. Dadurch können dem Außendienstmitarbeiter einerseits alle Informationen zu dem jeweiligen Kunden bereitgestellt und andererseits der Besuchsbericht entsprechend vorbefüllt werden.

Der Abschnitt zum Thema Mobile zeigt anhand ausgewählter Beispiele, dass es verschiedenste Anwendungsgebiete für die mobile Nutzung eines geoCRM gibt bzw. erst durch ein solches die notwendige Basis für ein vollwertiges mCRM geschaffen wird. Nachdem in diesem Kapitel verschiedene Anwendungsgebiete von geoCRM auszugsweise vorgestellt wurden, soll im nächsten Kapitel nun ein mögliches Anforderungsprofil eines geoCRM diskutiert werden.

5 Prototyp

Anhand eines Prototyps soll nun eine mögliche Integration von den im Kapitel 4.1 beschriebenen Business Mapping Funktionalitäten in ein bestehendes CRM-System gezeigt werden. Dabei wird zuerst die zur Anwendung kommende Architektur im nächsten Kapitel beschrieben, darauf folgend die notwendigen Schritte zur Integration der Geodaten auf Ebene der Datenbank in Kapitel 5.2 betrachtet und in Kapitel 5.3 auf die Entwicklung einer Web Mapping Komponente eingegangen. Abschließend wird im Kapitel 5.4 gezeigt, wie sich diese Funktionalitäten in die Applikation einfügen lassen.

5.1 Architekturmodell

Die zum Einsatz kommende Softwarearchitektur leitet sich einerseits aus der im vorliegenden CRM-System verwendeten Entwicklungsumgebung *Oracle Application Express* ab und wird andererseits von dem gewählten clientseitigen Web Mapping Architekturmodell (siehe Kapitel 2.4.2) beeinflusst. Der Einsatz des clientseitigen Web Mappings wurde aufgrund der im Vergleich mit dem serverseitigen Web Mapping geringeren Komplexität bei der Entwicklung des Prototypen gewählt. Als Web Mapping Bibliothek wurde *OpenLayers* verwendet, da dadurch die Abhängigkeit von den Nutzungsbedingungen einer proprietären API minimiert werden können. Die Abbildung 24 auf der nächsten Seite zeigt die schematische Darstellung des Architekturmodells. Die wichtigsten Komponenten sind dabei die Datenbank und der Web Browser. Der Web Server dient als Schnittstelle zwischen Browser und Datenbank während *Google Maps* als Basiskarte von *OpenLayers* geladen wird.

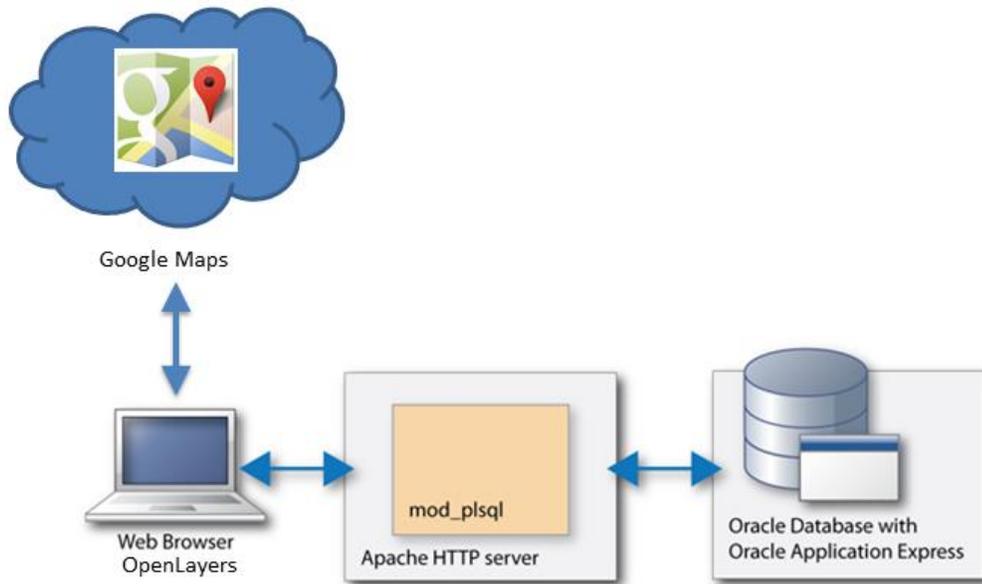


Abbildung 24: Architekturmodell des Prototypen (Quelle: in Anlehnung an Oracle o. J.)

5.2 Integration in die Datenbank

In diesem Kapitel wird beschrieben, welche Schritte notwendig sind um eine bestehende Kundendatenbank um Spalten zur Aufnahme von geometrischen Datentypen zu erweitern. Dazu wird im ersten Schritt das vorhandene Datenmodell analysiert, um Integrationspunkte zu identifizieren und zu definieren.

5.2.1 Analyse des Datenmodells

In einem CRM-System steht sinngemäß der Kunde im Mittelpunkt der Betrachtung. Mit dem Ziel eine umfassende Kenntnis über diesen zu gewinnen. Das heißt alle sogenannten Business Objekte innerhalb einer Kundendatenbank, haben eine direkte oder indirekte Beziehung zum Kunden und damit zu dessen Standort, der zumindest durch die Adresse des Kunden definiert ist. Die folgende Abbildung zeigt einen stark vereinfachten Ausschnitt aus dem Datenmodell des CRM-Systems, mit den

wichtigsten Business Objekten und deren Beziehung mit dem Kunden bzw. dessen Standort.

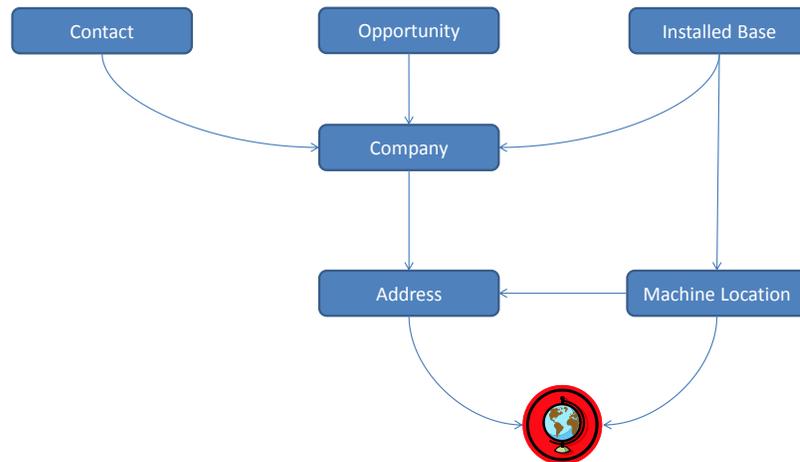


Abbildung 25: Vereinfachter Ausschnitt aus dem CRM-Datenmodell (Quelle: eigene Darstellung)

5.2.2 Erweiterung des Datenmodells

Zur Speicherung der Standortinformationen eines Kunden wird die Tabelle `COMPANY_DETAILS` um eine Spalte mit dem in Kapitel 2.1 vorgestellten Geometriedatentyp von *Oracle Spatial* – `SDO_GEOMETRY` – erweitert. Das untenstehende PL/SQL Listing zeigt die dafür notwendigen Schritte

- (1) Hinzufügen einer Spalte vom Typ `SDO_GEOMETRY`,
- (2) Eintragen der notwendigen Informationen über die hinzugefügte Spalte bzw. den neuen räumlichen Layer in die Metadatentabelle und
- (3) Anlage des räumlichen Indexes.

```

-- add column location of datatype SDO_GEOMETRY
ALTER TABLE company_details ADD location SDO_GEOMETRY;

-- define metadata information for the added spatial layer
INSERT INTO user_sdo_geom_metadata VALUES (
'COMPANY_DETAILS', -- TABLE_NAME
'LOCATION', -- COLUMN_NAME
SDO_DIM_ARRAY -- DIMINFO attribute for defining dimension bounds,
tolerance
(
SDO_DIM_ELEMENT
(
'LONGITUDE', -- DIMENSION_NAME for first dimension
-180, -- SDO_LB for the dimension (the lower bound)
180, -- SDO_UB for the dimension (the upper bound)
0.5 -- Tolerance of 0.5 meters
),
SDO_DIM_ELEMENT
(
'LATITUDE', -- DIMENSION_NAME for second dimension
-90, -- SDO_LB for the dimension (the lower bound)
90, -- SDO_UB for the dimension (the upper bound)
0.5 -- Tolerance of 0.5 meters
)
),
8307 -- SRID value for specifying the coordinate system
(WGS84)
);

-- create spatial index for the added spatial layer
CREATE INDEX comp_details_sidx -- indexname
ON comp_details (location) -- tablename(columnname)
INDEXTYPE IS mdsys.spatial_index
PARAMETERS ('LAYER_GTYPE=POINT'); -- defining index for a layer of
points

```

Listing 3: Skript zur Erweiterung der Tabelle (Quelle: nach KOTHURI et al. 2007)

Nachdem nun die Voraussetzungen zur Speicherung und performanten Abfrage von Geometriedaten geschaffen wurde, soll im nächsten Abschnitt ein Blick auf die vorhandenen Adressdatensätze geworfen werden um die Geokodierung dieser bestmöglich vorzubereiten und die Qualität des Ergebnisses abschätzen zu können.

5.2.3 Analyse der Adressdatensätze

Im ersten Schritt wurde analysiert, in welchen Ländern bzw. Kontinenten sich die Adressen befinden. Die untenstehende Abbildung zeigt die Aufteilung der

Kundenadressen nach Kontinenten und damit, dass sich knapp die Hälfte der Adressen in Europa befindet und zusammen mit den Adressen in Nord Amerika, gute zwei Drittel der Adressdatensätze in „westliche“ Länder fallen.

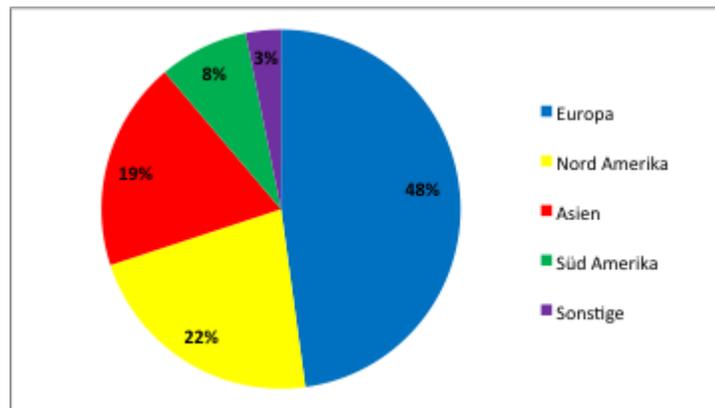


Abbildung 26: Aufteilung der Kundenadressen nach Kontinent (Quelle: eigene Darstellung)

Im Hinblick auf die Qualität der Adressdaten zeigt sich ein ernüchterndes Bild. So sind ausnahmslos alle der im Kapitel 2.3.1 nach DING et al. 2000 beschriebenen Probleme in der einen oder anderen Form vorhanden. Zusätzlich sind über zehn Prozent der Adressen Postfächer, angegeben in den unterschiedlichsten Varianten und Sprachen. Knapp fünf Prozent der Adressen bestehen nur aus Land und Ort, ohne Angaben zur Straßenanschrift (was applikationsseitig auch nicht verpflichtend ist) und zwei Prozent haben keine Hausnummer angegeben. Aus diesen Zahlen kann abgeleitet werden, dass zumindest siebzehn Prozent der Daten im besten Fall auf Ortsebene geokodiert werden können. Bezieht man die in Abbildung 26 angeführten Zahlen unter der Annahme mit ein, dass die Ergebnisse der Geokodierung außerhalb Europas und Nord Amerikas – vor allem auch außerhalb der großen Städte in diesen Gebieten – ohnehin nicht auf Gebäude genau bestimmt werden können, lassen sich für nicht einmal zwei Drittel der Daten gute Ergebnisse erwarten. Der nächste Absatz beschreibt die angewandte Vorgehensweise bei der Geokodierung der Adressdatensätze.

5.2.4 Geokodierung der Adressdatensätze

Das untenstehende Listing beschreibt stark vereinfacht die notwendigen Schritte bei der Geokodierung einer Adressdatenbank. Aufgrund der bei der Analyse der Adressdatensätze erkannten Probleme, können gewisse Elemente bei dem Aufbau des zu geokodierenden Adressstrings ausgenommen werden. Nachdem der String den Voraussetzungen des jeweiligen Geokodierdienstes entsprechend formatiert wurde, kann die URL für den Aufruf des Dienstes erstellt werden. Im nächsten Schritt wird die Antwort des Geokodierdienstes geparkt und die interessierenden Daten extrahiert. Wenn die Nutzungsbedingungen des verwendeten Dienstes die Speicherung der Koordinaten nicht verbietet, werden diese abschließend gespeichert. Die im Zuge der Prototypentwicklung verwendete, komplette PL/SQL Prozedur ist in Anhang C ersichtlich.

```
Für alle Adressdatensätze
  Auswahl der Elemente
  Aufbau des zu geokodierenden Adressstrings
  Vorbereitung der URL für die Anfrage beim Geokodierdienst
  Aufruf des Geokodierdienstes
  Parsen des Ergebnisses und Extrahieren der Daten
  Speichern der Daten
```

Listing 4: Pseudocode der Prozedur zur Geokodierung der Adressdatensätze

Für den Prototypen wurde an dieser Stelle zu Testzwecken der Geokodierungsdienst der *Google Maps* API verwendet. Wie in Kapitel 2.3.2 beschrieben, verbieten die Nutzungsbedingungen jedoch ein über das Zwischenspeichern der Koordinaten zur Anzeige in *Google Maps* hinausgehende Speicherung und Nutzung der Daten. Aus diesem Grund müsste bei einer produktiven Nutzung ein alternativer webbasierter oder lokaler Geokodierungsdienst verwendet werden.

Im folgenden Absatz wird erläutert, wie die nun vorhandenen Geodaten zur Geovisualisierung an eine auf *OpenLayers* basierenden Web Mapping Komponente geliefert und von dieser Dargestellt werden können.

5.2.5 Aufbereitung der Daten für das Web Mapping

Für eine aussagekräftige Geovisualisierung der Daten muss einerseits eine Möglichkeit geschaffen werden diese entsprechend zu filtern und müssen diese andererseits in ein Format gebracht werden, die von der jeweilig zum Einsatz kommenden Web Mapping Komponente verarbeitet werden kann. Ein Überblick über die unterstützten Formate einzelner Web Mapping Bibliotheken bzw. APIs wurde in Kapitel 2.4 gegeben. Im Kapitel 2.1 wurde erwähnt, dass Oracle Spatial über eine integrierte Funktion zur Konvertierung von `SDO_GEOMETRY` Daten in GML verfügt. Um flexibler bestimmen zu können, welche Daten übertragen werden, wurde entschieden ein einfaches, auf JSON basierendes Format zu verwenden. Mit Hilfe der PL/SQL Funktion `APEX_UTILS.json_from_sql` kann auf Basis eines SQL-Query-Strings ein JSON-Format generiert werden, anhand dessen definiert wird, welche Informationen an das Web Mapping übergeben werden. Um Interoperabilität einer derartigen Lösung zu gewährleisten, sollte im Endausbau die Verwendung von GeoJSON oder, wie im Kapitel 2.4 beschrieben, einer Web Mapping Server Komponente angedacht werden.

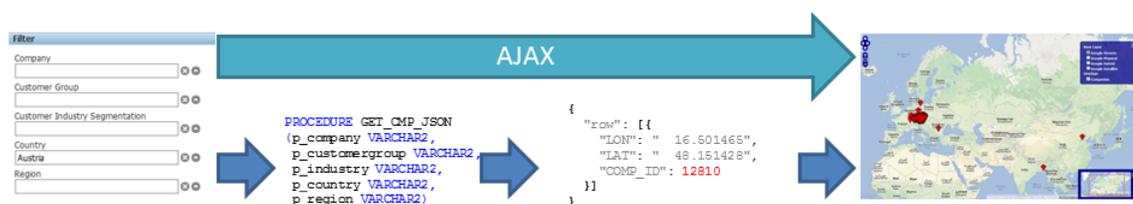


Abbildung 27: Schematischer Ablauf der Datenaufbereitung (Quelle: eigene Darstellung)

Abbildung 27 zeigt den schematischen Ablauf der Datenaufbereitung inkl. Geovisualisierung und Listing 5 beispielhaft das Ergebnis der PL/SQL Prozedur für einen Datensatz im JSON-Format. Da das komplette für den Prototypen entwickelte PL/SQL Paket für die Filterung und der Aufbereitung der Daten für die im nächsten Kapitel gezeigten Integrationspunkte knapp 5 000 Zeilen umfasst, wird in Anhang D nur ein Ausschnitt des Pakets für einen Bereich der Applikation gezeigt.

```

{
  "row": [{
    "LON": " 16.501465",
    "LAT": " 48.151428",
    "COMP_ID": 12810
  }]
}

```

Listing 5: Aufbereiteter Datensatz im JSON-Format

Nachdem auf diesem Weg die Pins an das Web Mapping übertragen wurden, fehlt noch die Beschreibung der Erstellung des Pop-Ups. Ein Pop-Up mit zusätzlichen, kontextbezogenen Informationen wird bei einem Mausklick auf einen der Pins angezeigt. Um die Informationen entsprechend gefiltert aufzubereiten, wird die Kunden-ID und gegebenenfalls die gesetzten Filterkriterien an eine PL/SQL Prozedur übergeben. Diese Prozedur generiert anhand dieser Parameter dann den HTML-Code der in dem Pop-Up angezeigten Informationen.



Abbildung 28: Schematischer Ablauf der Pop-Up Generierung (Quelle: eigene Darstellung)

Die obenstehende Abbildung zeigt wiederum den schematischen Ablauf bei der Generierung eines Pop-Ups und Listing 6 auf der nächsten Seite den generierten

HTML-Code. Dabei handelt es sich um eine einfache HTML-Tabelle mit Verknüpfungen zu anderen Seiten in der Applikation. Grundsätzlich können bei diesem Verfahren jedoch alle HTML, CSS und JavaScript Funktionalitäten genutzt werden.

```
<h2>
<a href="f?p=120:12:::NO:12:P12_CMP_ID:12810" target="_blank">Test Performanz Firma</a>
</h2>
<table>
<tr>
<td nowrap="nowrap">
<a href="f?p=120:1050:::NO:1050:P12_CMP_ID:12810" target="_blank">39 Contacts</a>
</td>
</tr>
<tr>
<td nowrap="nowrap">
<a href="f?p=120:1060:::NO:1060:P12_CMP_ID:12810" target="_blank">6 Notes</a>
</td>
</tr>
<tr>
<td nowrap="nowrap">
<a href="f?p=120:1070:::NO:1070:P12_CMP_ID:12810" target="_blank">46 Opportunities</a>
</td>
</tr>
<tr>
<td nowrap="nowrap">
<a href="f?p=120:1090:::NO:1090:P12_CMP_ID:12810" target="_blank">30 Installed Bases</a>
</td>
</tr>
<tr>
<td nowrap="nowrap">
<a href="f?p=120:1120:::NO:1120:P12_CMP_ID:12810" target="_blank">4 Interactions</a>
</td>
</tr>
</table>
<br />
<br />
<a href="javascript:zoomToLocation(16.501465,48.151428)">Zoom to Location</a>
```

Listing 6: Generierter HTML-Code für den Inhalt eines Pop-Ups

In diesem Kapitel wurde gezeigt, dass eine bestehende Datenbank mit relativ geringem Aufwand für die Speicherung und Verarbeitung von Geodaten erweitert werden kann. Für bestehende Adressdatensätze kann mit Hilfe einer Vielzahl an im Internet verfügbaren Geokodierdiensten (unter Berücksichtigung der jeweiligen Nutzungsbedingungen) direkter Raumbezug hergestellt werden. Im letzten Absatz wurde, überleitend auf das nächste Kapitel, beschrieben, wie die in der Datenbank vorhandenen (Geo-)Daten für die Geovisualisierung mittels Web Mapping aufbereitet werden können. Im nächsten Kapitel soll nun erläutert werden, wie die von der

Datenbank aufbereiteten Daten von der Web Mapping Komponente als Basis für die Geovisualisierung genutzt werden können.

5.3 Web Mapping

Für die Web Mapping Komponente des Prototypen wurde das im Kapitel vorgestellte Architekturmodell des clientseitigen Web Mappings unter Einsatz der JavaScript Mapping Bibliothek *OpenLayers* verwendet. Dabei werden die – wie im letzten Kapitel beschrieben – von der Datenbank aufbereiteten Daten als Marker dargestellt. Als Basiskarten werden die drei Google Maps Layer *Physical*, *Hybrid* und *Satellite* verwendet. In diesem Kapitel soll anhand von Ausschnitten aus dem Quellcode gezeigt werden, wie die das Web Mapping funktioniert. Der gesamte Quellcode für einen der im nächsten Kapitel beschriebenen Integrationspunkte (der „Company Map“) ist in Anhang E ersichtlich. Vor der Einbindung einer Karte in eine Webseite muss diese entsprechend initialisiert werden. Dabei werden grundlegende Parameter wie verwendete Projektionen, maximaler Kartenausschnitt, die Basiskarten sowie Overlays und Kontrollelemente definiert und der Karte hinzugefügt; das ganze geschieht beim Laden der Seite. Listing 7 zeigt die Definition der Parameter und die Erstellung der Karte.

```
// Definition der maximalen Ausdehnung der Karte
var maxExt = new OpenLayers.Bounds(-20037508.34, -20037508.34,
                                   20037508.34, 20037508.34);

// Definition der Kartenparameter
var options = {
  projection: new OpenLayers.Projection("EPSG:900913"), // Mercator
  displayProjection: new OpenLayers.Projection("EPSG:4326"), // WGS84
  numZoomLevels: 21,
  maxExtent: maxExt,
  maxResolution: 156543.0339,
  units: 'm'
};

// Erstellen der Karte
map = new OpenLayers.Map('map', options);
```

Listing 7: Initialisierung der Karte

Das untenstehende JavaScript Listing zeigt das Hinzufügen der einzelnen Layer der Basiskarten. In diesem Fall wurden zwei der insgesamt vier unterschiedlichen *Google Maps* Layer verwendet. Des Weiteren wird das Overlay für die Darstellung der Kundenstandorte als Marker erstellt und der Karte hinzugefügt.

```
// Hinzufügen der Basiskarten-Layer
map.addLayers([
  new OpenLayers.Layer.Google("Google Streets", {
    numZoomLevels: 20,
    sphericalMercator: true,
    maxExtent: maxExt
  }),
  new OpenLayers.Layer.Google("Google Satellite", {
    type: google.maps.MapTypeId.SATELLITE,
    sphericalMercator: true,
    maxExtent: maxExt
  })
]);

// Erstellen und Hinzufügen des Marker-Layers
markerLayer = new OpenLayers.Layer.Markers("Companies");
map.addLayer(markerLayer);

// Hinzufügen der Kontrollelemente für Layerwechsel und Übersichtskarte
map.addControl(new OpenLayers.Control.LayerSwitcher());
map.addControl(new OpenLayers.Control.OverviewMap({maximized: true}));
```

Listing 8: Hinzufügen der Layer und Kontrollelemente zur Karte

In den nächsten Listings werden ausschnittsweise JavaScript Quellcodeteile dargestellt, die zur Kommunikation mit der Datenbank mit Hilfe von AJAX (siehe Listing 9), dem Parsen und der Auswertung der von der Datenbank im JSON-Format zurückgelieferten Werte (siehe Listing 10) und zu deren Darstellung auf der Karte (siehe Listing 11) verwendet werden.

```

// Aufbau der Anfrage
ajaxRequest = new htmldb_Get(null, $('pFlowId').value,
                            'APPLICATION_PROCESS=loadLocCmpJSON', 0);

// Hinzufügen der Filterkriterien als Parameter
ajaxRequest.add('P1010_COUNTRY', $v('P1010_COUNTRY'));
ajaxRequest.add('P1010_REGION', $v('P1010_REGION'));
ajaxRequest.add('P1010_CUSTOMERGROUP', $v('P1010_CUSTOMERGROUP'));
ajaxRequest.add('P1010_INDUSTRY', $v('P1010_INDUSTRY'));

// Absetzen der Anfrage
retJSON = ajaxRequest.get();

```

Listing 9: Kommunikation mit der Datenbank

Für die Kommunikation mit der Datenbank werden die von der Entwicklungsumgebung angebotenen Funktionen – htmldb_Get in Verbindung mit einem sog. Application Processes – verwendet. Diese sind in diesem Umfeld gegenüber der Standard-AJAX-Calls deutlich komfortabler zu verwenden sind. Die in Abbildung 29 dargestellte Definition des Application Process veranschaulicht die Schnittstelle zwischen Web Applikation und Datenbank-Prozedur.

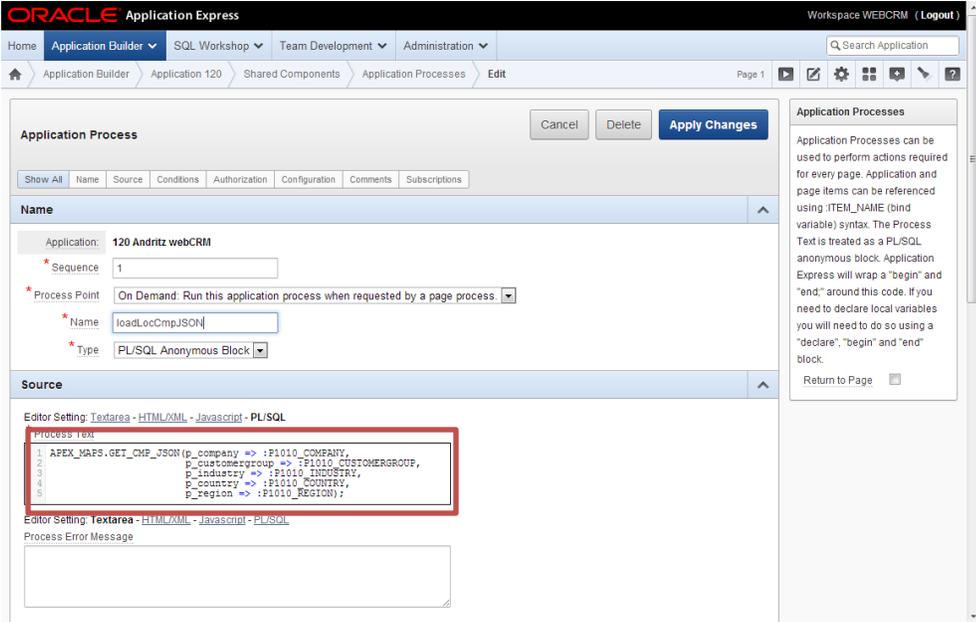


Abbildung 29: Definition des Application Process in Oracle APEX (Quelle: eigene Darstellung)

Das Listing 10 zeigt, wie mit Hilfe der jQuery Funktion `parseJSON` aus einem JSON String ein JavaScript Objekt erzeugt wird. Das JavaScript Objekt wird in weiterer Folge in einer Schleife durchlaufen und für jeden Eintrag – der einem Kundenstandort entspricht – ein Marker mit der in Listing 11 gezeigten Funktion `addMarker` erzeugt. Die aus der Datenbank gelieferten Koordinatenwerte werden dabei von WGS84 in die von Google Maps verwendete Transversale Mercator Projektion transformiert.

```
// Parsen der Antwort
json = jQuery.parseJSON(retJSON);

// Auswertung der Daten und Hinzufügen der Standortmarker
for(var i = 0; i < jsonObj.row.length; i++){
    addMarker(new OpenLayers.LonLat(
        jsonObj.row[i].LON,
        jsonObj.row[i].LAT
    ).transform(projWGS84, projMercator),
        jsonObj.row[i].COMP_ID,
        markerLayer);
}

// Zoomen der Karte auf die Ergebnisse
map.zoomToExtent(markerLayer.getDataExtent());
```

Listing 10: Auswertung der Daten

Dieses Kapitel abschließend zeigt Listing 11 auf der nächsten Seite die Funktion zur Erstellung eines Features und eines dazugehörigen Marker-Objekts sowie eines Event-Handlers. Der Event-Handler zeigt bei einem Mausklick auf den Marker das entsprechende Pop-Up mit aus der Datenbank gelieferten Informationen an, vorausgesetzt dieses wird noch nicht angezeigt. Die im Pop-Up dargestellten Informationen werden wiederum mit Hilfe von AJAX von der Datenbank abgefragt (siehe Abbildung 28 bzw. Listing 6 in Kapitel 5.2.5).

```

function addMarker(ll, compID, markLayer)
{
    var feature = new OpenLayers.Feature(markLayer, ll);

    feature.closeBox = true;
    feature.popupClass = OpenLayers.Class(OpenLayers.Popup.Anchored,
        {autoSize: true}
    );
    feature.data.popupContentHTML = compID;
    feature.data.overflow = "auto";
    feature.data.icon = new OpenLayers.Icon('/c/maps/marker.png',
        new OpenLayers.Size(21, 25),
        new OpenLayers.Pixel(-10.2, -25)
    );

    // Definition des Event-Handlers
    var markerClick = function (evt) {
        if (this.popup == null) {
            // Abfrage des PopUp Inhalts von der Datenbank mittels AJAX
            var ajaxRequest = new htmldb_Get(null, $('pFlowId').value,
                'APPLICATION_PROCESS=LoadLocCmpPopup', 0);
            ajaxRequest.add('P1010_IN_CMP_ID', feature.data.popupContentHTML);
            feature.data.popupContentHTML = ajaxRequest.get();

            // Erstellung und Anzeige des Pop-Ups
            this.popup = this.createPopup(this.closeBox);
            map.addPopup(this.popup);
            this.popup.show();
        } else {
            this.popup.toggle();
        }
        currentPopup = this.popup;
        OpenLayers.Event.stop(evt);
    };

    // Erstellung des Markers und Registrierung des Event-Handlers
    var marker = feature.createMarker();
    marker.events.register("mousedown", feature, markerClick);

    // Hinzufügen des Markers zum Marker-Layer
    markLayer.addMarker(marker);
}

```

Listing 11: Funktion zur Anzeige der Standortmarker und des PopUps

5.4 Integration in die Applikation

In diesem Kapitel wird erläutert, wie und an welchen Stellen der CRM-Applikation die in den letzten Kapiteln beschriebenen Erweiterungen integriert werden. Spezielle Karten-Seiten werden für die Business Objekte *Company* (der Kunde), *Opportunity* (die Geschäftschance) und *Installed Base* (die „installierte Basis“ an verkauften

Produkten) bereitgestellt. Zusätzlich wird die Möglichkeit angeboten, den Kundenstandort sowie den Standort einer verkauften Maschine entweder mit Hilfe der Geokodierung einer Adresse oder mittels manueller Setzens des Ortes auf der Karte zu bestimmen. Die Abbildung 30 zeigt rot umrandet die Platzierung der Links zu den einzelnen Karten-Seiten auf der Startseite – dem sogenannten „Dashboard“ – der Applikation in Form einer HTML Image Map.

Paul Jereb's Dashboard

My Dashboard | My Personal Info

Opportunities | Companies | Contacts | Installed Base

Recently Edited Opportunities

Opportunity Name	Opportunity Nr	SAP Q-Project Nr	Company Name
test_qol/va3	1713827.4711	-	Test Performanz Firma
test_qol/va3	1363928	-	Test Performanz Firma
test_qol/va3	1362891	-	Test Performanz Firma
test_qol/va3	1941798.4711	-	Test Performanz Firma

row(s) 1 - 5 of 34 Next >

Closing Soon | Follow up required | Update required | Worksheets for automatic IB creation

Closing Soon

No opportunities found for the specified period.
Show my opportunities that close within days

Maps

Image Map

COMPANY | OPPORTUNITY | INSTALLED BASE

News

No news available at the moment
Edit News

Quick Links

New Opportunity
New Contact
New Installed Base

Welcome to webCRM v. 3.5.6, Paul Jereb

Abbildung 30: HTML Image Map mit Links zu den einzelnen Karten-Seiten (Quelle: eigene Darstellung)

Auf der Bearbeitungsseite für einen Kunden, wird der Kundenstandort anhand einer interaktiven Karte angezeigt (siehe Abbildung 31). Unter der Karte werden zusätzlich noch Längen- und Breitengrade des Standortes angeführt und die Funktionalität angeboten den Standort zu verändern (wie in Abbildung 32 dargestellt). Wird die Adresse eines Kunden verändert, erscheint beim Speichern eine Abfrage ob der Standort auf der Karte ebenso geändert werden soll.

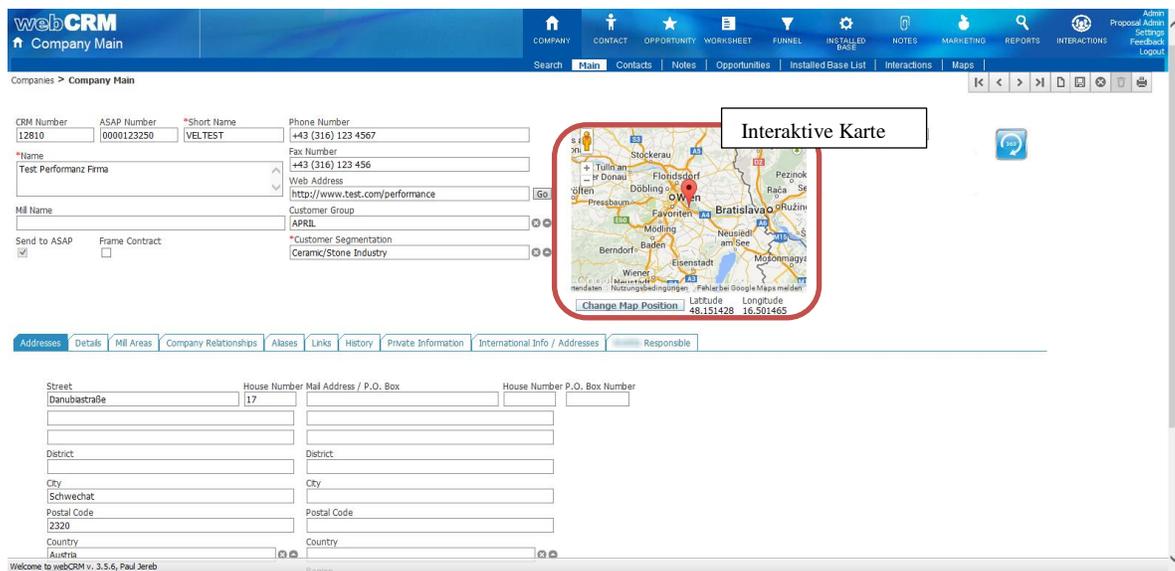


Abbildung 31: Kartendarstellung des Kundenstandorts (Quelle: eigene Darstellung)

Die untenstehende Abbildung 32 zeigt das Dialogfenster zur Bearbeitung des Kundenstandorts. Das Dialogfenster besteht aus zwei Teilen, der interaktiven Karte auf der linken Seite und der Eingabe- und Bedienfelder auf der rechten Seite. Der Standort kann auf einen der folgenden drei Wege gesetzt werden:

- Geokodierung der Adresse
- Eingabe der Koordinaten
- Setzen des Pins auf der interaktiven Karte

Durch Betätigung des Buttons „Update Map“ wird der gesetzte Standort gespeichert und auf der Seite zur Bearbeitung der Kundendaten die Karte entsprechend aktualisiert dargestellt.

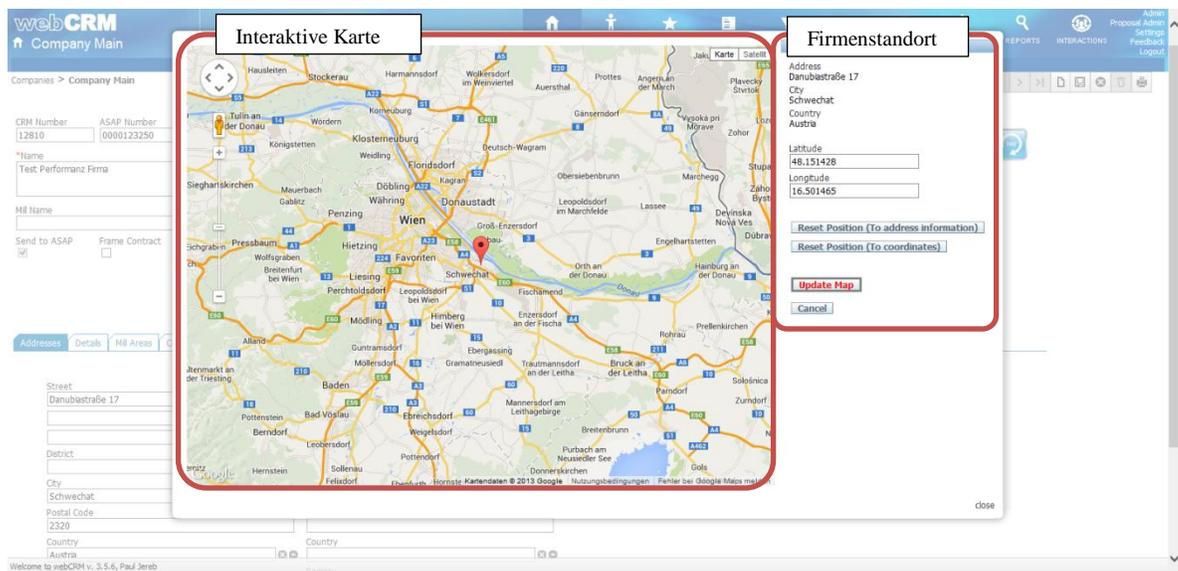


Abbildung 32: Dialogfenster zur Bearbeitung des Kundenstandorts (Quelle: eigene Darstellung)

Die Karten-Seiten für die eingangs erwähnten Business Objekte sind alle nach dem gleichen, in Abbildung 33 dargestellten, Schema aufgebaut und unterscheiden sich nur durch die Filtermöglichkeiten und die zum Einsatz kommenden Map-Overlays. Auf der linken Seite befindet sich der Kartenbereich mit Schaltflächen zum Pannen und Zoomen der Karte (was auch per Drag & Drop bzw. Scrollen mit der Maus funktioniert), einem Dialogfeld zur Änderung der Basiskarte und der angezeigten Map-Overlays sowie einer Übersichtskarte in der der aktuell dargestellte Kartenausschnitt ersichtlich ist. Die durch den Filter auf der rechten Seite eingeschränkte Ergebnismenge wird mit Hilfe von verschiedenfarbigen Pins (eine Farbe pro Map-Overlay) angezeigt. Per Mausklick auf einen der Pins werden in einem Pop-Up Detailinformationen mit Verlinkung zu den jeweiligen Teilbereichen der Applikation dargestellt (siehe Abbildung 34). Rechts oben befindet sich der Bereich „Map Control“ der folgende Funktionalitäten aufweist:

- **Update Map** – aktualisiert die Karte anhand der definierten Filterkriterien und zoomt auf das kleinste die Ergebnismenge umgebende Rechteck (die sog. Minimum Bounding Box).

- **Reset Map** – löscht alle auf der Karte angezeigten Pins und setzt das Zoomlevel zurück, sodass im Kartenbereich die gesamte Weltkarte dargestellt wird.
- **Find City** – Geokodiert den eingegebenen Städtenamen und bewegt die Karte an die entsprechende Stelle.

Zusätzlich wird in diesem Bereich sowohl die Anzahl der in der Ergebnismenge enthaltenen Elemente angezeigt, als auch die Anzahl der Elemente die aufgrund fehlender Standortinformationen nicht angezeigt werden können. Letzteres dient zur sukzessiven Verbesserung der Datenqualität. Abbildung 33 zeigt im Hinblick auf die Datenqualität auch, dass sich nicht alle Kunden mit einer österreichischen Adresse innerhalb der Staatsgrenzen befinden, was auf falsche Standortinformationen bei den betroffenen Datensätzen hinweist. Keine andere Darstellungsform als die Karte ermöglicht ein vergleichbar schnelles Erkennen dieser Fehler.

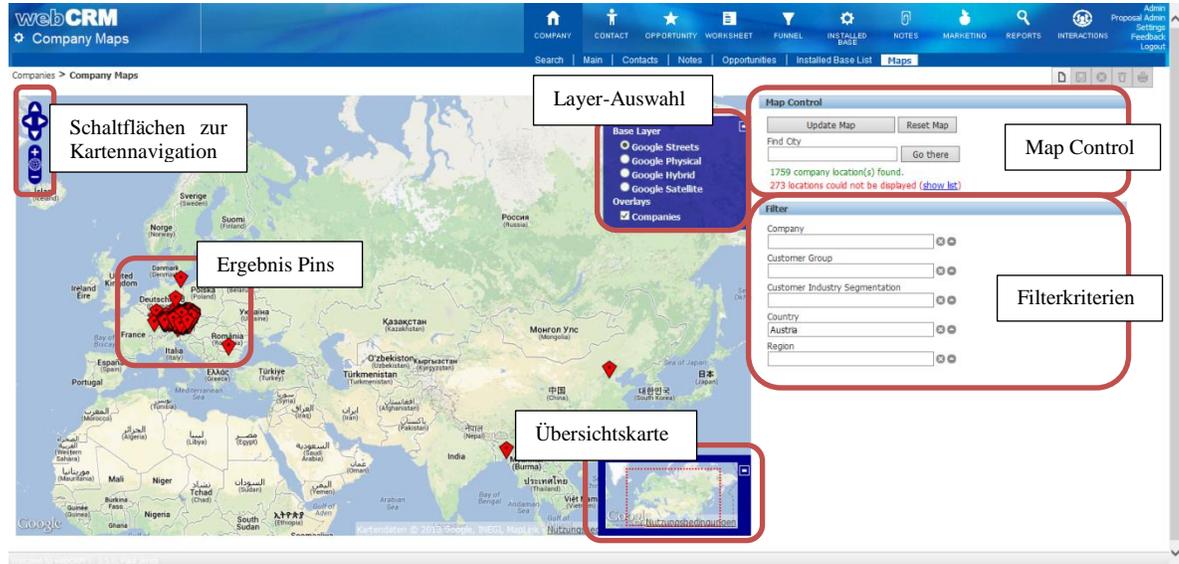


Abbildung 33: Anatomie einer Map-Page (Quelle: eigene Darstellung)

Die Abbildung auf der nächsten Seite zeigt die unterschiedlichen Pop-Ups mit Detailinformationen im jeweiligen Kontext und Verlinkungen zu den entsprechenden Stellen in der Applikation.

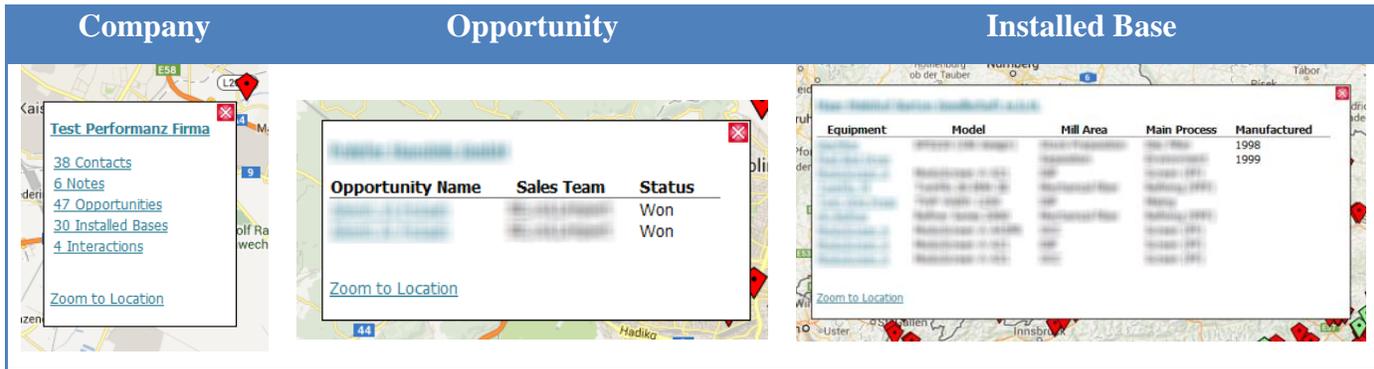


Abbildung 34: Pop-Ups mit Detailinformationen (Quelle: eigene Darstellung)

Das Kapitel abschließend soll anhand der folgenden Abbildung die Einbindung auf der Seite zur Bearbeitung der Installed Base gezeigt werden. Standardmäßig wird als *Machine Location* (der Maschinenstandort) die des *Final Customer* (des Endkunden) angezeigt.

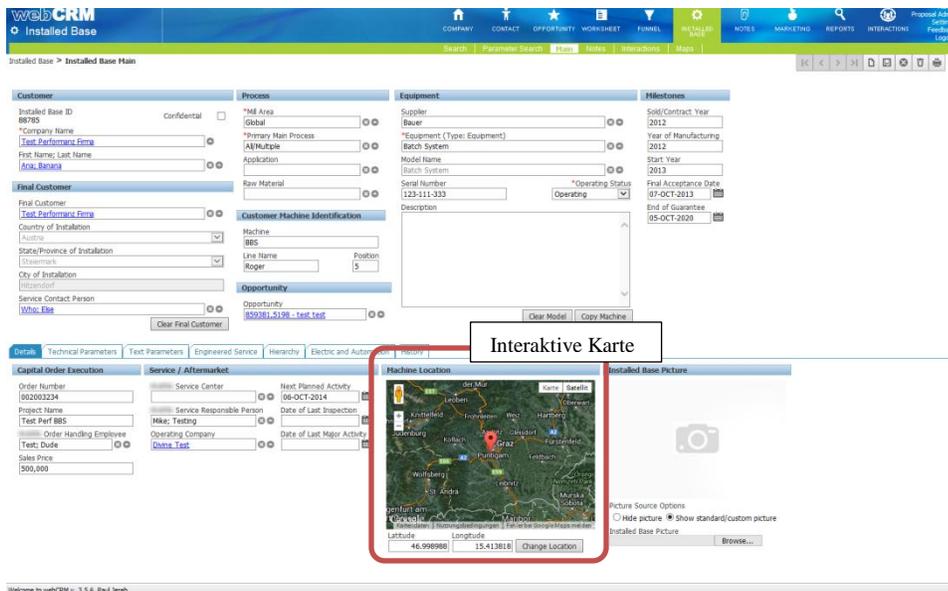


Abbildung 35: Standort einer Maschine (Quelle: eigenen Darstellung)

In manchen Fällen unterscheidet sich der Maschinenstandort jedoch von dem des Endkunden. Ist dies der Fall, kann für den Standort eine separate Adresse bzw. separate Koordinaten mit Hilfe des in untenstehenden Abbildung 36 dargestellten Dialogfeldes definiert werden. Ähnlich dem in Abbildung 32 gezeigten Dialogfenster zur Bearbeitung des Firmenstandorts, teilt sich dieses Dialogfenster in einen Kartenbereich auf der linken und einen Bereich mit Eingabe- und Bedienelementen auf der rechten Seite des Fensters.

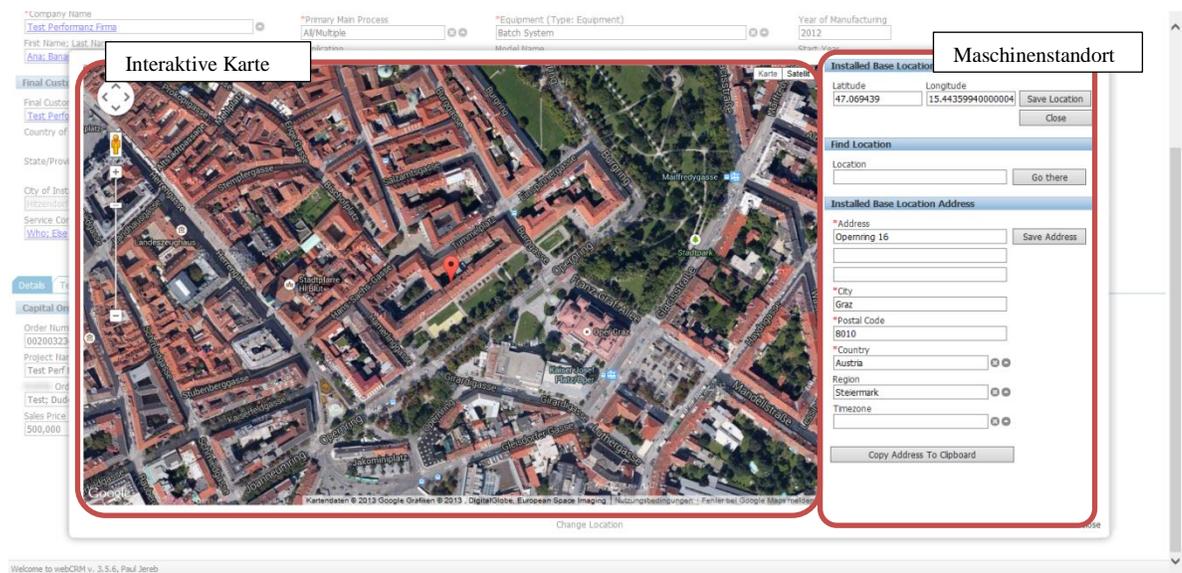


Abbildung 36: Dialogfenster zur Bearbeitung des Maschinenstandorts (Quelle: eigene Darstellung)

6 Zusammenschau und Ausblick

Die vorliegende Arbeit untersuchte die Integration von GIS-Technologien in CRM-Systeme – sogenannte geoCRM-Systeme – aus einer vornehmlich technischen Perspektive. Die in den einführenden Literaturüberblick einfließende Literaturrecherche ergab, dass im thematischen Umfeld in welchem die Arbeit angesiedelt ist, teilweise noch starker Forschungsbedarf gegeben ist. Die Recherche zeigte, dass sich die vorhandene Literatur in drei Kategorien gliedern lässt, welche sich auf unterschiedlichen Ebenen mit der Thematik befassen. Eine Kategorie beschäftigt sich – mit starkem Fokus auf das Geomarketing – allgemein mit dem Einsatz von GIS-Technologien in der Wirtschaft. In dieser Kategorie finden sich Schlagworte wie Business Geographics, Business GIS und eben Geomarketing. Die zweite Kategorie widmet sich ausgiebig mit den für die Integration von Standortdaten in betriebliche Informationssysteme sowie deren weiterfolgende Nutzung notwendigen, technischen Teilgebieten und Grundlagen. Wenige Werke der Kategorie in die auch diese Arbeit fällt, beschäftigen sich mit der Verbindung der Themenkomplexe Kundenbeziehungsmanagement und GIS-Technologien. Die Werke die es dennoch tun, weisen allesamt auf das außerordentliche Potential dieser Verbindung hin.

In einem ausführlichen Theorieteil werden die Technologien behandelt, die im Zuge einer Nutzung von Standortdaten zum Einsatz kommen. Dabei wird gezeigt, dass mittlerweile alle namhaften Datenbanksysteme, und vor allem auch die verwendete *Oracle* Datenbank, über die Voraussetzungen verfügen Geodaten zu speichern und optimiert abzufragen. Die Methode der Geokodierung wurde näher betrachtet und auf die Herausforderungen einer qualitativ hochwertigen eingegangen. Die zur Nutzung von Geokodierungsdiensten und zur interoperablen Bereitstellung von Geodaten innerhalb eines Unternehmens notwendigen Grundlagen im Bereich der M2M-Kommunikation mit Hilfe von Web Services notwendige Basis wurde geschaffen. Den Theorieteil abschließend wurde auf die zwei Web Mapping Architekturmodelle

eingegangen und jeweils zur Verfügung stehende proprietäre und Open-Source Komponenten vorgestellt.

Anschließend wurde in Zusammenarbeit mit Experten aus Verkauf und Management ein Anforderungsprofil entwickelt und auf potentielle Anwendungsgebiete eines geoCRM-Systems eingegangen. Dabei wurde versucht, aus den unterschiedlichen Blickwinkeln der Themenkomplexe an deren Schnittstelle sich geoCRM befindet, Beispiele für diese Anwendungsgebiete auszuarbeiten. Die Bearbeitung des Kapitels zeigte, dass Optimierungen in der operativen Nutzung des Systems vor allem durch die Geovisualisierung in Form von Standortkarten – dem sog. Business Mapping – erzielt werden können, was einerseits zu einer Steigerung der Datenqualität und andererseits zu einer effizientere Nutzung im Vergleich zur Darstellung in Tabellen- und Listenform führt. Auch lassen sich durch intelligente Aggregation von Informationen in einer interaktiven Karte – beispielsweise in Form von PopUps oder von mit den Elementen der Karte verbundener Tabellen – die andernfalls zur Informationsbeschaffung notwendigen Navigationsschritte innerhalb der CRM-Applikation drastisch minimieren. Zur Unterstützung der Entscheidungsfindung im strategischen Bereich sind vordergründig Visualisierungen hochaggrierter Daten in Form von thematischen Karten (Coroplethenkarten und Kartodiagrammen) von Interesse. Im Zuge der sog. Location Intelligence werden dabei auch Daten aus anderen betrieblichen Informationssystemen, wie z.B. ERP-Systeme, herangezogen. Im mobilen Umfeld gibt es Anwendungsgebiete in Form der klassischen Umfeldsuche, aber auch innovative Anwendungsgebiete unter der Nutzung des Geofencings zur Entwicklung von LBS, Check-In Funktionalitäten, der Routenplanung und in weiterer Folge auch des Salesforce-Managements.

Im Zuge der Prototypentwicklung wurde gezeigt, wie GIS-Technologien in ein bestehendes CRM-System integriert werden können. Dabei wurden auf Datenbankebene die notwendigen Voraussetzungen zur Speicherung von Standortinformationen geschaffen und eine Methode entwickelt, um bestehende

Adressdatensätze zu geokodieren. Zur Geovisualisierung unterschiedlicher Business Objekte wurde auf Basis der Open Source Bibliothek *OpenLayers* eine Web Mapping Komponente entwickelt und, neben kleineren Widgets zur Kundenstandortdarstellung und Geokodierung bei der Kundenerfassung, in die CRM-Applikation integriert.

Die Arbeit an dem Prototyp hat verdeutlicht, dass es, eine flexible webbasierte Entwicklungsumgebung und eine „geotaugliche“ Datenbank vorausgesetzt, mit vertretbarem Aufwand möglich ist, die datenbanktechnischen Grundlagen für ein geoCRM zu schaffen und Business Mapping Funktionalitäten in ein bestehendes CRM-System „nahtlos“ zu integrieren. Literatur und Dokumentation für die technischen Teilbereiche sind im notwendigen Detaillierungsgrad vorhanden. Die Erweiterung der Datenbank zur Speicherung von Geometriedaten war ohne Probleme innerhalb kürzester Zeit durchgeführt. Im Hinblick auf die Geokodierung wurde die Wichtigkeit qualitativ hochwertiger Adressdatensätze für akkurate Geokodierungsergebnisse deutlich. Die Rahmenbedingungen des Prototyps erlaubten allerdings einen großen Spielraum die Genauigkeit der Standortdaten betreffend. Ist dies nicht der Fall, wäre es nötig mit großem Aufwand die Adressdatensätze zu bereinigen und entsprechend der Voraussetzungen des Geokodierungsdienstes zu standardisieren. Wie im Kapitel 2.3.2 angemerkt, müsste für den produktiven Einsatz der entwickelten Lösung zur Geokodierung der vorhandenen Adressdaten, aufgrund der Nutzungsbestimmungen eine Alternative zum Geokodierungsdienst der *Google Maps* API verwendet werden. Bei der Verwendung der Open-Source Web Mapping Bibliothek *OpenLayers* wurde die Mächtigkeit und Flexibilität der Bibliothek im Hinblick auf die Anpassbarkeit und die Möglichkeiten zur Einbindung einer breiten Palette an Datenquellen deutlich. Das Zusammenspiel mit der Datenbank verlief ohne Probleme. Einzig vom „Look&Feel“ her wirkt *OpenLayers* schon etwas in die Jahre gekommen und es mangelt an Unterstützung mobiler Browsertechnologien.

Der Prototyp und die anschließende Expertenbefragung brachten auch eine Reihe an Verbesserungspotentialen und weitere mögliche Ausbaustufen einer geoCRM Lösung

zum Vorschein. Die von manchen Geokodierungsdiensten als Bestandteil des Ergebnisses zurückgelieferten, standardisierten Adressen können zur Verbesserung der Adressdatenqualität verwendet werden. Sehr schnell wurde das Problem des sog. Clusterings ersichtlich (siehe Kapitel 2.4.2) welches in *OpenLayers* noch nicht zufriedenstellen gelöst ist, vor allem im Vergleich zu den sehr eleganten Lösungen für die *Google Maps* API. Auch scheint die Erweiterung des geoCRM um eine Routingfunktionalität sich deutlich einfacher mit Hilfe der *Google Maps* API realisieren zu lassen. Trotz dieser Vorteile, spricht die Herstellerunabhängigkeit weiterhin für die Verwendung von *OpenLayers* für die Entwicklung einer Web Mapping Komponente. Ein großer Nachteil der entwickelten Lösung ist die mangelnde Interoperabilität hinsichtlich der Verwendung offener Standards für Abfrage und Bereitstellung der Geodaten, sowie der Wiederverwertbarkeit der Web Mapping Komponente z.B. im mobilen Umfeld bzw. der erzeugten Karten in anderen Applikationen oder im Reporting. Zusammen mit der beim Architekturmodell des clientseitigen Web Mappings fehlenden Möglichkeit, Karten zu Exportieren bzw. zu Drucken, ist das ein Argument für die intensivere Betrachtung des Einsatzes von serverseitigem Web Mapping.

Die Arbeit abschließen und ein Fazit ziehend kann gesagt werden, dass die technischen Voraussetzungen für ein geoCRM in einem mehr als ausreichenden Reifestadium vorhanden sind. Die Expertenbefragungen ergaben, dass ein geoCRM durch die Geovisualisierung der Kunden- und Maschinenstandorte sowohl die tägliche Arbeit der Verkaufsmitarbeiter mit dem System optimiert als auch die Entscheidungsfindungsprozesse des Managements beschleunigt. Was fehlt ist die interdisziplinäre Abstimmung zwischen Wirtschaft und Geographie, um den Anforderungen an GIS-Technologien im Kundenbeziehungsmanagement mit den heute bestehenden technischen Möglichkeiten gerecht werden zu können.

7 Quellenverzeichnis

AMELUNXEN, C. (2009): An approach to geocoding based on volunteered spatial data. Salzburg.

AMELUNXEN, C. (2010): An Approach to geocoding based on volunteered Spatial Data. 4 S.

AZAZ, L. (2011): The use of Geographic Information Systems (GIS) in Business.

BAKSHI, R.; KNOBLOCK, C. A. und THAKKAR, S. (2004): Exploiting Online Sources to Accurately Geocode Addresses. Los Angeles, CA. 10 S.

BALLATORE, A.; TAHIR, A.; MCARDLE, G. und BERTOLOTTO, M. (2011): A Comparison of Open Source Geospatial Technologies for Web Mapping. International Journal of Web Engineering and Technology 6 (4): 354.

BEHR, F.-J. (2010): Geocoding: Fundamentals, Techniques, Commercial and Open Services. In: BEHR, F.-J. (Hrsg.): Applied Geoinformatics for Society and Environment. Hochschule für Technik. Stuttgart: 111–122.

BILL, R. und ZEHNER, M. L. (2001): Geokodierung. <http://www.geoinformatik.uni-rostock.de/einzel.asp?ID=760> (Zugriff: 5.10.2013).

BRIDWELL, S. und MILLER, H. (2011): Location-Based Services. In: LIU, L. u. ÖZSU, M. (Hrsg.): Encyclopedia of Database Systems: SpringerReference. Springer-Verlag. Berlin/Heidelberg.

BRINEY, A. (2013): Business Geographics. <http://geography.about.com/od/geographicstechnology/a/businessgeog.htm> (Zugriff: 2.11.2013).

BRINKHOFF, T. (2010): Geodatenbanksysteme als Basis für Geoweb-Anwendungen. In: REHRL, K. (Hrsg.): Geoweb. dpunkt-Verl. Heidelberg: 29–38.

BRINKHOFF, T. (2012): Geodatenbanksysteme in Theorie und Praxis. Wichmann. Heidelberg.

CHAPPELL, D. (2009): SOAP vs. REST: Complements or Competitors? Palm Springs, CA. 42 S.

CZERANKA, M. (2001): Business Geographics und Geomarketing als Schlüssel zur unternehmenseigenen Schatztruhe. In: STROBL, J. u. FALLY, M. (Hrsg.): Business Geographics. Wichmann. Heidelberg: 1–10.

DICKINSON, A. J. (2013): MyCRM Company Blog | Using Maps to find your way in CRM 2013! - EMAP for #CRM2013 RELEASED. <http://blog.mycrmgroup.com/post/2013/11/05/Using-Maps-to-find-your-way-in-CRM-2013!-EMAP-2013-RELEASED.aspx> (Zugriff: 11.01.2014).

- DING, J.; GRAVANO, L. und SHIVAKUMAR, N. (2000): Computing Geographical Scopes of Web Resources. In: EL ABBADI, A. (Hrsg.): Marking the millennium. Morgan Kaufmann. Orlando, FL: 545–556.
- DIOMIN, Y. (2010): An Introduction to Geocoding. <http://www.drdoobs.com/database/an-introduction-to-geocoding/228500287#> (Zugriff: 6.10.2013).
- DOUGLAS, B. (2008): Achieving business success with GIS. Wiley. Chichester. Online-Ressource.
- ESRI (2007): Geospatial Service-Oriented Architecture (SOA). <http://www.esri.com/library/whitepapers/pdfs/geospatial-soa.pdf> (Zugriff: 15.08.2013).
- ESRI (2014): Esri's Location Platform brings the Power of Place to Microsoft Dynamics CRM. <http://www.esri.com/esri-news/releases/14-1qtr/esri-location-platform-brings-the-power-of-place-to-microsoft-dynamics-crm> (Zugriff: 17.01.2014).
- FÄRBER, K.-D. (2012): Business GIS. AV Akadmeikerverlag. Saarbrücken. V, 111 S.
- FARLEY, J. (2004): Location-enabled Platform Technology. GIM International.
- FERNANDES, A. I.; GOULAO, M. und RODRIGUES, A. (2013): A Comparison of Maps Application Programming Interfaces. 13 S.
- FIELDING, R. T. (2000): Architectural Styles and the Design of Network-based Software Architectures. Irvine. 180 S.
- FLEISCHMANN, R. und KAISER, K. (2001): Oracle Spatial Option - Theorie und Praxis. In: STROBL, J. u. FALLY, M. (Hrsg.): Business Geographics. Wichmann. Heidelberg: 160–172.
- FLORCZYK, A. J.; LÓPEZ-PELLICER, F. J.; GAYAN, D.; RODRIGO-CARDIEL, P.; LATRE, M. A. und NOGUERAS-ISO, J. (2011): Compound Geocoder: get the right position. 11 S.
- GOLDBERG, D. W. (2008): A Geocoding Best Practices Guide. 287 S.
- GOODCHILD, M. F. (2007): Citizens as Voluntary Sensors: Spatial Data Infrastructure in the World of Web 2.0. International Journal of Spatial Data Infrastructures Research (2): 24–32.
- Google (2009): Countries currently supported by the Google Maps APIs geocoder. http://gmaps-samples.googlecode.com/svn/trunk/mapcoverage_filtered.html (Zugriff: 17.11.2013).
- Google (2013a): Google Maps API. <https://developers.google.com/maps/> (Zugriff: 23.11.2013).
- Google (2013b): Google Maps/Google Earth APIs Terms of Service. <https://developers.google.com/maps/terms> (Zugriff: 23.11.2013).

- Google (2013c): The Google Geocoding API.
<https://developers.google.com/maps/documentation/geocoding/> (Zugriff: 23.11.2013).
- GRIMSHAW, D. J. (2000): Bringing geographical information systems into business. John Wiley. New York. xiii, 346.
- HAKE, G.; GRÜNREICH, D. und MENG, L. (2002): Kartographie. 604 S.
- HAZZARD, E. (2011): Openlayers 2.10 beginner's guide. Birmingham, U. K. xii, 351.
- HERTER, M. (2008): Definition des Begriffes "Geomarketing". In: HERTER, M. u. MÜHLBAUER, K.-H. (Hrsg.): Handbuch Geomarketing. Wichmann. Heidelberg [u.a.]: 5–7.
- HERTER, M. und MÜHLBAUER, K.-H. Hrsg. (2008): Handbuch Geomarketing. Wichmann. Heidelberg [u.a.]. x, 366 S.
- HILLMANN-KÖSTER, B. (2010): Revolutionierung von CRM durch Einbeziehung georeferenzierter Informationen. *ew* 109 (20): 60–63.
- HONGJIAN, Z.; FEI, X. und CHUNCIU, W. (2011): Research on the Application of GIS Technology in Spatial Mode Customer Relationship Management System. *Journal on Innovation and Sustainability* 2 (2): 34–37.
- HUTCHINSON, M. und VEENENDALL, B. (o.J.): Towards using Intelligence to move from Geocoding to Geolocating. 13 S.
- HUTCHINSON, M. J. und VEENENDAAL, B. (2013): An agent-based framework for intelligent geocoding. *Applied Geomatics* 5 (1): 33–44.
- IACOVELLA, S. und YOUNGBLOOD, B. (2013): GeoServer beginner's guide. Packt Publishing. Birmingham, U.K. vii, 326.
- JANSEN, M. und ADAMS, T. (2010): OpenLayers. Open Source Press. München. 344 S.
- KAUER, J. (2011): Aufbruch in die Geoinformationsgesellschaft mit Microsoft Bing Maps. Wichmann, H. Berlin. XI, 169 S. ;
- KOTHURI, R.; GODFRIND, A. und BEINAT, E. (2007): Pro Oracle Spatial for Oracle database 11g. Springer-Verlag New York. Berkeley, CA, New York, NY. 787 S.
- LEUBER, W.; HIPPER, H. und WILDE, K. D. (2011): CRM - Grundlagen, Konzepte und Prozesse. In: HIPPER, H.; HUBRICH, B. u. WILDE, K. D. (Hrsg.): Grundlagen des CRM. Gabler. Wiesbaden: 15–55.
- LUCKE, J. von und GEIGER, C. (2010): Open Government Data - Frei verfügbare Daten des öffentlichen Sektors. 37 S.
- MAGUIRE, D. J.; SMITH, R. und KOUYOUMJIAN, V. (2008): The business benefits of GIS. ESRI Press. Redlands, Calif. ix, 243.
- MARTIN, W. (2012): Location Intelligence. 11 S.

- MCNEIL, B. (2013): Five reasons CRM (Customer Relationship Management) needs Business Mapping Tools. <http://www.directionsmag.com/articles/five-reasons-crm-customer-relationship-management-needs-business-mappi/358086> (Zugriff: 8.11.2013).
- MELZER, I. (2010): Service-orientierte Architekturen mit Web Services. Spektrum Akad. Verl. Heidelberg. 381 S.
- Microsoft (2013): Enhanced Bing Maps experience in CRM 2013. <http://blogs.msdn.com/b/crm/archive/2013/10/22/enhanced-bing-maps-experience-in-crm-2013.aspx> (Zugriff: 11.01.2014).
- MURRAY, C. (2013a): Oracle Spatial. http://docs.oracle.com/cd/E11882_01/appdev.112/e11830.pdf (Zugriff: 19.10.2013).
- MURRAY, C. (2013b): Oracle Spatial and Graph. http://docs.oracle.com/cd/E16655_01/appdev.121/e17896.pdf (Zugriff: 19.10.2013).
- NEUMANN, A. (2011): Web Mapping and Web Cartography. In: SHEKHAR, S. u. XIONG, H. (Hrsg.): Encyclopedia of GIS: SpringerReference. Springer-Verlag. Berlin/Heidelberg.
- NITSCHKE, M. (2008): Customer Relationship Management (CRM). In: HERTER, M. u. MÜHLBAUER, K.-H. (Hrsg.): Handbuch Geomarketing. Wichmann. Heidelberg [u.a.]: 216–222.
- Open Geospatial Consortium (2005): Web Feature Service Implementation Specification. 131 S.
- Open Geospatial Consortium (2006): Web Map Server Implementation Specification. 85 S.
- Oracle (o. J.): Application Express Architecture. <http://www.oracle.com/technetwork/developer-tools/apex/apex-arch-086399.html> (Zugriff: 1.12.2013).
- PANIAN, Z. (2012): A New Dimension of Business Intelligence: Location-based Intelligence. <https://waset.org/journals/waset/v63/v63-122.pdf> (Zugriff: 28.07.2013).
- PETERSON, M. P. (2012a): Online Mapping with APIs. In: PETERSON, M. P. (Hrsg.): Online Maps with APIs and WebServices. Springer. Berlin, New York: 3–12.
- PETERSON, M. P. Hrsg. (2012b): Online Maps with APIs and WebServices. Springer. Berlin, New York. 1 online resource (ix, 318).
- PLANK, P. (2004): Komparative Analyse von GIS-Funktionalitäten bei CRM-Lösungen. Graz.
- REHRL, K. Hrsg. (2010a): Geoweb. dpunkt-Verl. Heidelberg. 128 S.
- REHRL, K. (2010b): Geoweb verbindet WWW und physische Welt. In: REHRL, K. (Hrsg.): Geoweb. dpunkt-Verl. Heidelberg: 6–18.

- ROONGPIBOONSOPIT, D. und KARIMI, H. A. (2010): Quality assessment of online street and rooftop geocoding services.
<http://www.thefreelibrary.com/Quality+assessment+of+online+street+and+rooftop+geocoding+services.-a0243797905> (Zugriff: 8.11.2013).
- SANTOS, W. (2012): 56 Geocoding APIs: Geocoder, Google and MapLarge.
<http://blog.programmableweb.com/2012/07/25/56-geocoding-apis-geocoder-google-and-maplarge/> (Zugriff: 19.10.2013).
- SCHMIDT, M. und WEISER, P. (2012): Web Mapping Services: Development and Trends. In: PETERSON, M. P. (Hrsg.): Online Maps with APIs and WebServices. Springer. Berlin, New York: 13–22.
- SCHÜSSLER, F. (2006): Geomarketing. Tectum-Verl. Marburg. X, 235 S.
- SHNEIDERMAN, B. (1991): Touch screens now offer compelling uses. IEEE Software 8 (2): 93–94.
- STROBL, J. und FALLY, M. Hrsg. (2001): Business Geographics. Wichmann. Heidelberg. VIII, 174 S.
- SVENNERBERG, G. (2010): Beginning Google Maps API 3. Apress. [New York]. 1 online resource (xvi, 310).
- SWADENER, D. (2013): Oracle Application Express SQL Workshop Guide.
http://docs.oracle.com/cd/E37097_01/doc/doc.42/e35128.pdf (Zugriff: 27.10.2013).
- SWIFT, J. N.; GOLDBERG, D. W. und WILSON, J. P. (2008): Geocoding Best Practices: Review of Eight Commonly Used Geocoding Systems. Los Angeles, CA. 55 S.
- TAMU (2013): List of Online Geocoding Systems.
<http://geoservices.tamu.edu/Services/Geocode/OtherGeocoders/> (Zugriff: 23.11.2013).
- TAPPERT, W. (2007): Geomarketing in der Praxis. Harzer. Karlsruhe. 180 S.
- TILKOV, S. (2011): REST und HTTP. dpunkt. Heidelberg. 248 S.
- TSAI, J. (2010): Here, there, and everywhere: knowing where consumers are can reveal a lot about who they are. Customer relationship management (Malibu, Calif.) 14 (1): 38.
- UFFMANN, M. (2011): Der Standort-Faktor: Revolutionierung von CRM durch Einbeziehung von georeferenzierten Informationen. Geomatik Schweiz (5): 198–200.
- WAGNER, D.; ZLOTNIKOVA, R. und BEHR, F.-J. (2009): XML-Based and Other Georelated Encodings: Overview of Main Existing Geocoding Formats. In: BEHR, F.-J. (Hrsg.): Applied Geoinformatics for Society and Environment. Hochschule für Technik. Stuttgart: 196–202.
- WANG, J. (2011): Building Oracle XML DB applications. McGraw-Hill. New York. xvi, 396.

WHITE, S. (2005): Tame the Beast by Matching Similar Strings.
http://www.catalysoft.com/articles/MatchingSimilarStrings.html?article=Tame_the_Beast_by_Matching_Similar_Strings_14 (Zugriff: 12.10.2013).

Wikipedia (2013): Georeferenzierung.
<http://de.wikipedia.org/w/index.php?oldid=117539172> (Zugriff: 5.10.2013).

WINKELMANN, P. (2012): Vertriebskonzeption und Vertriebssteuerung. Vahlen.
München. XXIII, 758 S.

WINSLOW, J. und CATTINI, M. (2007): Leveraging the Power of Location Intelligence to Enhance Business Intelligence.
http://resource.mapinfo.com/static/files/document/LIforBIWPA_Final.pdf (Zugriff: 28.07.2013).

WOLFE, D. und MOON, G. (2011): Location Intelligence. In: SHEKHAR, S. u. XIONG, H. (Hrsg.): Encyclopedia of GIS: SpringerReference. Springer-Verlag.
Berlin/Heidelberg.

Anhang A

Anfrage

<https://maps.googleapis.com/maps/api/geocode/json?address=Stattegger+Straße+18+A-8010+Graz&sensor=false>

Ergebnis

```
{
  "results" : [
    {
      "address_components" : [
        {
          "long_name" : "18",
          "short_name" : "18",
          "types" : [ "street_number" ]
        },
        {
          "long_name" : "Stattegger Straße",
          "short_name" : "Stattegger Str.",
          "types" : [ "route" ]
        },
        {
          "long_name" : "Andritz",
          "short_name" : "Andritz",
          "types" : [ "sublocality", "political" ]
        },
        {
          "long_name" : "Graz",
          "short_name" : "Graz",
          "types" : [ "locality", "political" ]
        },
        {
          "long_name" : "Graz",
          "short_name" : "Graz",
          "types" : [ "administrative_area_level_2", "political" ]
        },
        {
          "long_name" : "Styria",
          "short_name" : "Stmk.",
          "types" : [ "administrative_area_level_1", "political" ]
        },
        {
          "long_name" : "Austria",
          "short_name" : "AT",
          "types" : [ "country", "political" ]
        },
        {
          "long_name" : "8045",
          "short_name" : "8045",
          "types" : [ "postal_code" ]
        }
      ],
      "formatted_address" : "Stattegger Straße 18, 8045 Graz, Austria",
      "geometry" : {
        "location" : {
          "lat" : 47.1083629,
          "lng" : 15.4239255
        },
        "location_type" : "ROOFTOP",
        "viewport" : {
          "northeast" : {
            "lat" : 47.10971188029149,
```

```

        "lng" : 15.4252744802915
      },
      "southwest" : {
        "lat" : 47.10701391970849,
        "lng" : 15.4225765197085
      }
    }
  },
  "partial_match" : true,
  "types" : [ "street_address" ]
}
],
"status" : "OK"
}

```

Anfrage

<https://maps.googleapis.com/maps/api/geocode/xml?address=Stattegger+Straße+18+A-8010+Graz&sensor=false>

Ergebnis

```

<?xml version="1.0" encoding="UTF-8"?>
<GeocodeResponse>
  <status>OK</status>
  <result>
    <type>street_address</type>
    <formatted_address>Stattegger Straße 18, 8045 Graz, Austria</formatted_address>
    <address_component>
      <long_name>18</long_name>
      <short_name>18</short_name>
      <type>street_number</type>
    </address_component>
    <address_component>
      <long_name>Stattegger Straße</long_name>
      <short_name>Stattegger Str.</short_name>
      <type>route</type>
    </address_component>
    <address_component>
      <long_name>Andritz</long_name>
      <short_name>Andritz</short_name>
      <type>sublocality</type>
      <type>political</type>
    </address_component>
    <address_component>
      <long_name>Graz</long_name>
      <short_name>Graz</short_name>
      <type>locality</type>
      <type>political</type>
    </address_component>
    <address_component>
      <long_name>Graz</long_name>
      <short_name>Graz</short_name>
      <type>administrative_area_level_2</type>
      <type>political</type>
    </address_component>
    <address_component>
      <long_name>Styria</long_name>
      <short_name>Stmk.</short_name>
      <type>administrative_area_level_1</type>
      <type>political</type>
    </address_component>
  </address_component>

```

```
<long_name>Austria</long_name>
<short_name>AT</short_name>
<type>country</type>
<type>political</type>
</address_component>
<address_component>
  <long_name>8045</long_name>
  <short_name>8045</short_name>
  <type>postal_code</type>
</address_component>
<geometry>
  <location>
    <lat>47.1083629</lat>
    <lng>15.4239255</lng>
  </location>
  <location_type>ROOFTOP</location_type>
  <viewport>
    <southwest>
      <lat>47.1070139</lat>
      <lng>15.4225765</lng>
    </southwest>
    <northeast>
      <lat>47.1097119</lat>
      <lng>15.4252745</lng>
    </northeast>
  </viewport>
</geometry>
<partial_match>true</partial_match>
</result>
</GeocodeResponse>
```

Anhang B

Auflistung der möglichen Werte für SDO_GTYPE

Wert von SDO_GTYPE	Geometrietypp nach OGC-Spezifikation	Beschreibung
D000	UNKNOWN-GEOMETRY	Unbekannte Geometrie
D001	POINT	Punkt
D002	LINE or CURVE	Linienzug
D003	POLYGON or SURFACE	Einfaches Polygon, ggf. mit Löchern
D004	COLLECTION	Sammlung von unterschiedlichen Geometrien
D005	MULTIPOINT	Sammlung von Punkten
D006	MULTILINE	Sammlung von Linienzügen
D007	MULTIPOLYGON	Sammlung von überlappungsfreien Polygonen
D008	SOLID	Oberfläche
D009	MULTISOLID	Sammlung von überlappungsfreien Oberflächen

Anhang C

PL/SQL Block zur Geokodierung von Adressdatensätzen

```
SET DEFINE OFF;

DECLARE
    rec_cmpnt          temp_cmp_geocoding%ROWTYPE;
    v_url              VARCHAR2 (32767);
    v_http_request     UTL_HTTP.req;
    v_http_response    UTL_HTTP.resp;
    v_text             VARCHAR2 (32767);
    v_resp             CLOB;
    v_address          VARCHAR2 (32767);
BEGIN
    -- loop through company addresses
    FOR rec IN (SELECT adr.*, CNY.SHORTDESCRIPTION, RGN.NAME
                FROM addresses adr
                INNER JOIN countries cny ON adr.cny_code = cny.cny_code
                LEFT OUTER JOIN regions rgn ON adr.rgn_id = rgn.rgn_id
                WHERE adr.pa_id IN (SELECT cmpd_id
                                    FROM company_details
                                    WHERE latitude IS NULL
                                      AND longitude IS NULL)
                AND adrt_code = 'STREET')
    LOOP
        DBMS_LOB.createtemporary (v_resp, FALSE);

        -- decide which columns to include in the geocoding request
        IF rec.city IS NOT NULL AND INSTR (LOWER (rec.city), 'none') = 0
        THEN
            v_address := rec.city;
        END IF;

        IF rec.postalcode IS NOT NULL AND INSTR (LOWER (rec.postalcode), 'none')
        = 0
        THEN
            v_address := v_address || '+' || TRIM (rec.postalcode);
        END IF;

        IF rec.shortdescription IS NOT NULL
        THEN
            v_address := v_address || '+' || TRIM (rec.shortdescription);
        END IF;

        IF rec.name IS NOT NULL
        THEN
            v_address := v_address || '+' || TRIM (rec.name);
        END IF;

        -- try only to use address lines that can be geocoded
        IF TRIM (rec.addrline_1) IS NOT NULL
        AND INSTR (LOWER (rec.addrline_1), 'box') = 0
        AND INSTR (LOWER (rec.addrline_1), 'none') = 0
        AND INSTR (LOWER (rec.addrline_1), 'postfach') = 0
        AND INSTR (rec.addrline_1, 'PL') = 0
```

```

AND INSTR (rec.addrline_1, 'BP') = 0
AND INSTR (rec.addrline_1, 'CP') = 0
AND INSTR (rec.addrline_1, 'CR') = 0
AND INSTR (rec.addrline_1, 'P.O.') = 0
AND INSTR (LOWER (rec.addrline_1), 'floor') = 0
AND TRIM (rec.addrline_1) != '.'
AND INSTR (LOWER (rec.addrline_1), 'postale') = 0
AND INSTR (LOWER (rec.addrline_1), 'bag') = 0
AND INSTR (rec.addrline_1, 'Attn:') = 0
AND INSTR (rec.addrline_1, 'C/') = 0
THEN
    v_address := v_address || '+' || TRIM (rec.addrline_1);
END IF;

-- to meet google geocode api requirements, replace space with plus sign
v_address := REPLACE (v_address, ' ', '+');

-- initialise record
rec_cmpt := null;
rec_cmpt.cmp_id := rec.pa_id;
rec_cmpt.adr_string := v_address;

BEGIN
    -- prepare URL for HTTP request
    v_url := 'http://maps.google.com/maps/api/geocode/xml?address='
            || v_address
            || '&sensor=false';

    -- Make a HTTP request and get the response.
    l_http_request := UTL_HTTP.begin_request (v_url);
    l_http_response := UTL_HTTP.get_response (v_http_request);

    -- copy the response into the CLOB
    BEGIN
        LOOP
            UTL_HTTP.read_text (v_http_response, v_text, 32766);
            DBMS_LOB.writeappend (v_resp, LENGTH (v_text), v_text);
        END LOOP;
    EXCEPTION WHEN UTL_HTTP.end_of_body THEN
        UTL_HTTP.end_response (l_http_response);
    END;

    -- save the response
    rec_cmpt.resp_clob := v_resp;

    -- extract latitude and longitude from geocoding response
    SELECT EXTRACTVALUE (XMLTYPE (v_resp),
'/GeocodeResponse/result[1]/geometry/location/lat'),
        EXTRACTVALUE (XMLTYPE (v_resp),
        '/GeocodeResponse/result[1]/geometry/location/lng')
    INTO rec_cmpt.lat, rec_cmpt.lng
    FROM DUAL;

    EXCEPTION WHEN OTHERS THEN
        -- something went wrong, save error message
        rec_cmpt.errmsg := SQLERRM;
    END;

    -- clean up and wait a random time
    DBMS_LOB.freeTemporary (v_resp);

```

```

DBMS_LOCK.sleep (DBMS_RANDOM.VALUE (2, 10));

-- insert result into temporary table
INSERT INTO temp_cmp_geocoding VALUES rec_cmpt;

-- update company_details location columns
UPDATE company_details
SET location = SDO_GEOMETRY(2001,
                           8307,
                           SDO_POINT_TYPE(rec_cmpt.lat,   rec_cmpt.lng,
NULL),
                           NULL,
                           NULL)
WHERE cmpd_id = rec_cmpt.cmp_id;
END LOOP;

COMMIT;
END;

```

Anhang D

PL/SQL Package Body zur Datenaufbereitung für das Web Mapping (gekürzt)

```
CREATE OR REPLACE PACKAGE BODY WEBCRM.WEBCRM MAPS AS
  FUNCTION GET_CMP_NOLOCATION_SQL (p_company VARCHAR2 DEFAULT NULL,
    p_customergroup VARCHAR2 DEFAULT NULL,
    p_industry VARCHAR2 DEFAULT NULL,
    p_country VARCHAR2 DEFAULT NULL,
    p_region VARCHAR2 DEFAULT NULL) RETURN CLOB
  IS
    v_sql          VARCHAR2 (32000);
  BEGIN
    v_sql :=
    'select cmp.cmp_id, replace(replace(cmp.name,chr(10),' ' '),chr(13),' ' ') name
    from companies cmp
    inner join addresses adr on cmp.cmp_id = adr.pa_id and adr.adrt_code = ''STREET''
    left outer join company_details cmpd on cmp.cmp_id = cmpd.cmpd_id
    where cmpd.longitude is null AND cmpd.latitude is null
    and 1=1';

    IF p_company IS NOT NULL THEN
      v_sql := v_sql || ' AND cmp.cmp_id = '' ' || p_company || ''';
    END IF;

    IF p_customergroup IS NOT NULL THEN
      v_sql := v_sql || ' AND cmp.cgr_code = '' ' || p_customergroup || ''';
    END IF;

    IF p_industry IS NOT NULL THEN
      v_sql := v_sql || ' AND cmp.isg_code = '' ' || p_industry || ''';
    END IF;

    IF p_country IS NOT NULL THEN
      v_sql := v_sql || ' AND adr.cny_code = '' ' || p_country || ''';
    END IF;

    IF p_region IS NOT NULL THEN
      v_sql := v_sql || ' AND adr.rgn_id= '' ' || p_region || ''';
    END IF;

    RETURN v_sql;
  END GET_CMP_NOLOCATION_SQL;

  PROCEDURE GET_CMP_POUPTEXT(in_cmp_id    NUMBER)
  IS
    n_contacts number;
    n_notes    number;
    n_opportunities number;
    n_installedbases number;
    n_interactions number;
    v_ret      CLOB;
    v_apexHost VARCHAR2 (200);
    v_cmp_name companies.name%type;
    v_lon      company_details.longitude%type;
    v_lat      company_details.latitude%type;
  BEGIN
    SELECT VALUE INTO v_apexHost
    FROM dbadmin_apex_parameters
    WHERE key = 'HOST_NAME';

    SELECT cmp.name, cmpd.longitude, cmpd.latitude INTO v_cmp_name, v_lon, v_lat
    FROM companies cmp
```

```

LEFT OUTER JOIN company_details cmpd ON cmpd.cmpd_id = cmp.cmp_id
WHERE cmp.cmp_id = in_cmp_id;

SELECT count(*) INTO n_contacts FROM contacts WHERE cmp_id = in_cmp_id;
SELECT count(*) INTO n_notes FROM company notes WHERE cmp_id = in_cmp_id;
SELECT count(*) INTO n_opportunities FROM opportunities
WHERE cmp_id = in_cmp_id OR cmp_id_final = in_cmp_id;
SELECT count(*) INTO n_installedbases FROM references WHERE cmp_id = in_cmp_id;
SELECT count(*) INTO n_interactions FROM customer_interactions WHERE cmp_id =
in_cmp_id;

v_ret := '<h2><a
href="||v_apexHost||/f?p=120:12::NO:12:P12_CMP_ID:'||in_cmp_id||'"
target=_blank>||v_cmp_name||</a></h2><p>';
v_ret := v_ret || HTF.TABLEOPEN || HTF.tablerowopen || HTF.tabledata(cvalue =>
HTF.ANCHOR(v_apexHost || '/f?p=120:1050::NO:1050:P12_CMP_ID: ' || TO_CHAR(in_cmp_id),
to_char(nvl(n_contacts, 0)) || ' Contacts', NULL, 'target=_blank'), cnowrap =>
'NOWRAP') || HTF.tablerowclose
|| HTF.tablerowopen || HTF.tabledata(cvalue => HTF.ANCHOR(v_apexHost ||
'/f?p=120:1060::NO:1060:P12_CMP_ID: ' || TO_CHAR(in_cmp_id), to_char(nvl(n_notes, 0))
|| ' Notes', NULL, 'target=_blank'), cnowrap => 'NOWRAP') || HTF.tablerowclose
|| HTF.tablerowopen || HTF.tabledata(cvalue => HTF.ANCHOR(v_apexHost ||
'/f?p=120:1070::NO:1070:P12_CMP_ID: ' || TO_CHAR(in_cmp_id),
to_char(nvl(n_opportunities, 0)) || ' Opportunities', NULL, 'target=_blank'), cnowrap
=> 'NOWRAP') || HTF.tablerowclose
|| HTF.tablerowopen || HTF.tabledata(cvalue => HTF.ANCHOR(v_apexHost ||
'/f?p=120:1090::NO:1090:P12_CMP_ID: ' || TO_CHAR(in_cmp_id),
to_char(nvl(n_installedbases, 0)) || ' Installed Bases', NULL, 'target=_blank'),
cnowrap => 'NOWRAP') || HTF.tablerowclose
|| HTF.tablerowopen || HTF.tabledata(cvalue => HTF.ANCHOR(v_apexHost ||
'/f?p=120:1120::NO:1120:P12_CMP_ID: ' || TO_CHAR(in_cmp_id),
to_char(nvl(n_interactions, 0)) || ' Interactions', NULL, 'target=_blank'), cnowrap
=> 'NOWRAP') || HTF.tablerowclose
|| HTF.TABLECLOSE;
v_ret := v_ret || '<br><br><a
href=javascript:zoomToLocation('||v_lon||','||v_lat||')>Zoom to Location</a></p>';

htp.p(v_ret);
END GET_CMP_POPUPTEXT;

PROCEDURE GET_CMP_LOCATION COUNT(p company VARCHAR2 DEFAULT NULL,
                                p_customergroup VARCHAR2 DEFAULT NULL,
                                p_industry VARCHAR2 DEFAULT NULL,
                                p_country VARCHAR2 DEFAULT NULL,
                                p_region VARCHAR2 DEFAULT NULL)
IS
n count number;
v_sql VARCHAR2 (30000);
BEGIN
v_sql := 'select count(*)
from companies cmp
inner join addresses adr on cmp.cmp_id = adr.pa_id and adrt_code = ''STREET''
left outer join company_details cmpd on cmp.cmp_id = cmpd.cmpd_id
where CMPD.LATITUDE is not null and cmpd.longitude is not null';

IF p_company IS NOT NULL THEN
v_sql := v_sql || ' AND cmp.cmp_id = '' ' || p_company || ''';
END IF;

IF p_customergroup IS NOT NULL THEN
v_sql := v_sql || ' AND cmp.cgr_code = '' ' || p_customergroup || ''';
END IF;

IF p_industry IS NOT NULL THEN
v_sql := v_sql || ' AND cmp.isg_code = '' ' || p_industry || ''';
END IF;

IF p_country IS NOT NULL THEN
v_sql := v_sql || ' AND adr.cny_code = '' ' || p_country || ''';

```

```

END IF;

IF p_region IS NOT NULL THEN
    v_sql := v_sql || ' AND adr.rgn_id= '' || p_region || ''';
END IF;

EXECUTE IMMEDIATE v_sql INTO n_count;
HTP.p (n_count);

EXCEPTION WHEN OTHERS THEN
    HTP.p (0);
END GET_CMP_LOCATION_COUNT;

PROCEDURE GET_CMP_NOLOCATION_COUNT(p_company VARCHAR2 DEFAULT NULL,
                                  p_customergroup VARCHAR2 DEFAULT NULL,
                                  p_industry VARCHAR2 DEFAULT NULL,
                                  p_country VARCHAR2 DEFAULT NULL,
                                  p_region VARCHAR2 DEFAULT NULL)
IS
    n_count number;
    v_sql VARCHAR2 (30000);
BEGIN
    v_sql := 'select count(*)
              from companies cmp
              inner join addresses adr on cmp.cmp_id = adr.pa_id and adrt_code = ''STREET''
              left outer join company_details cmpd on cmp.cmp_id = cmpd.cmpd_id
              where CMPD.LATITUDE is null and cmpd.longitude is null';

    IF p_company IS NOT NULL THEN
        v_sql := v_sql || ' AND cmp.cmp_id = '' || p_company || ''';
    END IF;

    IF p_customergroup IS NOT NULL THEN
        v_sql := v_sql || ' AND cmp.cgr_code = '' || p_customergroup || ''';
    END IF;

    IF p_industry IS NOT NULL THEN
        v_sql := v_sql || ' AND cmp.isg_code = '' || p_industry || ''';
    END IF;

    IF p_country IS NOT NULL THEN
        v_sql := v_sql || ' AND adr.cny_code = '' || p_country || ''';
    END IF;

    IF p_region IS NOT NULL THEN
        v_sql := v_sql || ' AND adr.rgn_id= '' || p_region || ''';
    END IF;

    EXECUTE IMMEDIATE v_sql INTO n_count;
    HTP.p (n_count);

    EXCEPTION WHEN OTHERS THEN
        HTP.p (0);
    END GET_CMP_NOLOCATION_COUNT;

PROCEDURE GET_CMP_JSON(p_company VARCHAR2 DEFAULT NULL,
                       p_customergroup VARCHAR2 DEFAULT NULL,
                       p_industry VARCHAR2 DEFAULT NULL,
                       p_country VARCHAR2 DEFAULT NULL,
                       p_region VARCHAR2 DEFAULT NULL)
IS
    v_sql VARCHAR2 (30000);
    b_allnull BOOLEAN := TRUE;
BEGIN
    v_sql := 'SELECT to_char(round(cmpd.longitude,6), ''990.999999'') LON,
                  to_char(round(cmpd.latitude,6),''990.999999'') LAT,
                  CMP.CMP_ID COMP_ID
              FROM companies cmp
              INNER JOIN company_details cmpd ON cmp.cmp_id = cmpd.cmpd_id

```

```

        INNER JOIN addresses adr ON adr.pa_id = cmp.cmp_id AND adr.adrt_code =
        'STREET'
        WHERE cmpd.longitude is not null AND cmpd.latitude is not null AND 1=1';

    IF p_company IS NOT NULL THEN
        v_sql := v_sql || ' AND cmp.cmp_id = ''' || p_company || '''';
        b_allnull := FALSE;
    END IF;

    IF p_customer_group IS NOT NULL THEN
        v_sql := v_sql || ' AND cmp.cgr_code = ''' || p_customer_group || '''';
        b_allnull := FALSE;
    END IF;

    IF p_industry IS NOT NULL THEN
        v_sql := v_sql || ' AND cmp.isg_code = ''' || p_industry || '''';
        b_allnull := FALSE;
    END IF;

    IF p_country IS NOT NULL THEN
        v_sql := v_sql || ' AND adr.cny_code = ''' || p_country || '''';
        b_allnull := FALSE;
    END IF;

    IF p_region IS NOT NULL THEN
        v_sql := v_sql || ' AND adr.rgn_id= ''' || p_region || '''';
        b_allnull := FALSE;
    END IF;

    IF b_allnull = TRUE THEN
        v_sql := v_sql || ' AND 1 = 2';
    END IF;

    APEX_UTIL.json_from_sql (v_sql);

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        NULL;
    WHEN OTHERS THEN
        HTP.p ('{"row":[]}');
END GET_CMP_JSON;

END WEBCRM_MAPS;
/

```

Anhang E

JavaScript-Datei für die Geokomponente (gekürzt)

```
var map, markers;
var projWGS84 = new OpenLayers.Projection("EPSG:4326");
var projMercator = new OpenLayers.Projection("EPSG:900913");
var size = new OpenLayers.Size(21,25);
var offset = new OpenLayers.Pixel(-(size.w/2), -size.h);

function init() {
    var maxExt = new OpenLayers.Bounds(-20037508.34,-
    20037508.34,20037508.34,20037508.34);

    var options = {
        projection: projMercator,
        displayProjection: projWGS84,
        numZoomLevels: 21,
        maxExtent: maxExt,
        maxResolution: 156543.0339,
        units: 'm'
    };

    map = new OpenLayers.Map('map', options);

    var gmap = new OpenLayers.Layer.Google(
        "Google Streets",
        {numZoomLevels: 20, sphericalMercator: true, maxExtent: maxExt}
    );
    var gphy = new OpenLayers.Layer.Google(
        "Google Physical",
        {type: google.maps.MapTypeId.TERRAIN, sphericalMercator: true, maxExtent: maxExt}
    );
    var ghyb = new OpenLayers.Layer.Google(
        "Google Hybrid",
        {type: google.maps.MapTypeId.HYBRID, sphericalMercator: true, maxExtent: maxExt}
    );
    var gsat = new OpenLayers.Layer.Google(
        "Google Satellite",
        {type: google.maps.MapTypeId.SATELLITE, sphericalMercator: true, maxExtent:
maxExt}
    );

    map.addLayers([gmap, gphy, ghyb, gsat]);

    map.addControl(new OpenLayers.Control.LayerSwitcher());
    map.addControl(new OpenLayers.Control.OverviewMap({maximized: true}));

    map.setCenter(new OpenLayers.LonLat(0, 0).transform(projWGS84,projMercator), 2);

    markers = new OpenLayers.Layer.Markers("Companies");
    map.addLayer(markers);
}

function ajaxRequestAdd(request)
{
    request.add('P1010_COMPANY', $v('P1010_COMPANY'));
    request.add('P1010_COUNTRY', $v('P1010_COUNTRY'));
    request.add('P1010_REGION', $v('P1010_REGION'));
    request.add('P1010_CUSTOMERGROUP', $v('P1010_CUSTOMERGROUP'));
    request.add('P1010_INDUSTRY', $v('P1010_INDUSTRY'));
}
```

```

function loadLocationCount ()
{
    var retCOUNT;
    var req;

    req = new htmldb_Get (null,
    $('pFlowId').value, 'APPLICATION_PROCESS=loadLocCmpCnt', 0);
    ajaxRequestAdd (req);

    retCOUNT = req.get ();
    req = null;
    return retCOUNT;
}

function loadNoLocationCount ()
{
    var retCOUNT;
    var req;

    req = new htmldb_Get (null,
    $('pFlowId').value, 'APPLICATION_PROCESS=loadNoLocCmpCnt', 0);
    ajaxRequestAdd (req);

    retCOUNT = req.get ();
    req = null;
    return retCOUNT;
}

function loadLocationData ()
{
    var retJSON;
    var req;

    req = new htmldb_Get (null,
    $('pFlowId').value, 'APPLICATION_PROCESS=loadLocCmpJSON', 0);
    ajaxRequestAdd (req);

    retJSON = req.get ();
    req = null;
    return retJSON;
}

function addMapMarkers (seJSON)
{
    var json = jQuery.parseJSON (seJSON);

    markers.clearMarkers ();
    clearPopups ();

    for (var i = 0; i < jsonObj.row.length; i++) {
        ll = new OpenLayers.LonLat (jsonObj.row[i].LON,
        jsonObj.row[i].LAT).transform (projWGS84, projMercator);
        compID = jsonObj.row[i].COMP_ID;
        addMarker (ll, compID, markers);
    }

    map.zoomToExtent (markers.getDataExtent ());
}

function addMarker (ll, compID, markLayer)
{
    var feature = new OpenLayers.Feature (markLayer, ll);
    feature.closeBox = true;
    feature.popupClass = OpenLayers.Class (OpenLayers.Popup.Anchored, {autoSize: true});
    feature.data.compID = compID;
    feature.data.overflow = "auto";
    feature.data.icon = new OpenLayers.Icon ('/c/maps/img/marker.png', size, offset);

    var marker = feature.createMarker ();
}

```

```

var markerClick = function (evt) {
    if (this.popup == null) {
        var req = new htmldb_Get(null,
            $('pFlowId').value, 'APPLICATION_PROCESS=LoadLocCmpPopup', 0);
        ajaxRequestAdd(req);
        req.add('P1010_IN_CMP_ID', feature.data.compID);
        feature.data.popupContentHTML = ajaxRequest.get();
        this.popup = this.createPopup(this.closeBox);
        map.addPopup(this.popup);
        this.popup.show();
    } else {
        this.popup.toggle();
    }
    currentPopup = this.popup;
    OpenLayers.Event.stop(evt);
};

marker.events.register("mousedown", feature, markerClick);
marklayer.addMarker(marker);
}

function resetMapFilter()
{
    $s('P1010_COMPANY', '');
    $s('P1010_COUNTRY', '');
    $s('P1010_REGION', '');
    $s('P1010_CUSTOMERGROU', '');
    $s('P1010_INDUSTRY', '');
    markers.clearMarkers();
    clearPopups();
    $(".pseudoButtonInactive").trigger("click");
    map.setCenter(new OpenLayers.LonLat(0, 0).transform(projWGS84, projMercator), 2);
}

function resetMap()
{
    markers.clearMarkers();
    clearPopups();
    map.setCenter(new OpenLayers.LonLat(0, 0).transform(projWGS84, projMercator), 2);
}

function clearPopups()
{
    while( map.popups.length ) {
        map.removePopup(map.popups[0]);
    }
}

function zoomToLocation(lon, lat)
{
    ll = new OpenLayers.LonLat(lon, lat).transform(projWGS84, projMercator);
    map.setCenter(new OpenLayers.LonLat(ll.lon, ll.lat), 9);
}

function geocodeAddress(addressString)
{
    var geocoder = new google.maps.Geocoder();
    var geocoderRequest = { address: addressString };

    geocoder.geocode(geocoderRequest, function(results, status) {
        if (status == google.maps.GeocoderStatus.OK) {
            map.setCenter(new OpenLayers.LonLat(results[0].geometry.location.lng(),
                results[0].geometry.location.lat()).transform(projWGS84, projMercator), 9);
        } else {
            alert ("The city you entered couldn't be found on this planet.");
        }
    });
}

```