

**Florian Schögl**

# **WITTER - Weather Twitter**

## **MASTERARBEIT**

zur Erlangung des akademischen Grades  
Diplom-Ingenieur

Masterstudium Geomatics Science



**Technische Universität Graz**

Betreuer und Begutachter:  
Dipl.-Ing. Dr.techn. Clemens Strauß  
Institut für Geoinformation

Graz, September 2011



# EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz am September 2011

.....

(Unterschrift)

# STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources

Graz, September 2011

.....

(signature)



# Kurzfassung

Die vorliegende Masterarbeit mit dem Titel WITTER – „Weather Twitter“ beinhaltet die Vorgangsweise bei der programmatischen Umsetzung einer WebGIS Applikation zur Einmeldung und Darstellung von aktuellen Wetterinformationen.

Der theoretische Teil der Arbeit beschäftigt sich mit Positionierungsmethoden von Mobiltelefonen und Desktop PCs, die im Rahmen von standortbezogenen Diensten (Location Based Services (LBS)) zur Verwendung kommen. Dabei wurde besonders die Nutzung der Positionsinformation in Verbindung mit einem Internetbrowser mithilfe der World Wide Web Consortium (W3C) Geolocation Application Programming Interface (API), einer Web-Schnittstelle zur Positionsbestimmung, untersucht. In weiterer Folge wird auf verschiedene Wetterdienste und Möglichkeiten der Datenbeschaffung eingegangen. Zu guter Letzt werden der Aufbau einer Dateninfrastruktur, sowie mögliche Lösungen zur Darstellung der Daten beleuchtet.

Der praktischen Teil der Arbeit beinhaltet die Umsetzung einer Webseite zum Einmelden und Darstellen von Wetterdaten, sowie die dahinterliegende Geodateninfrastruktur (GDI). Diese „Weather Twitter“ genannte WebGIS Applikation wurde mit besonderem Augenmerk auf Open-Source-Software (OSS) umgesetzt. Die Positionierung des Benutzers erfolgt dabei wenn möglich auf Basis der W3C Geolocation API Spezifikation. Desweiteren werden die Wetterdaten im Sinne der Interoperabilität einer iPhone/Android-Applikation über eine definierte Schnittstelle zur Weiterverarbeitung bereitgestellt.



# Abstract

This master thesis entitled WITTER – „Weather Twitter“ exemplifies a procedure of programmatic implementation of a WebGIS Application to twitter and display the current weather information.

The theoretical part of the thesis describes some methods of positioning mobile phones as well as personal computers using Location Based Services (LBS). In particular, it investigates positioning information in connection with an internet browser. By using the World Wide Web Consortium (W3C) Geolocation Application Programming Interface (API), which is a web interface, the position of a device can be identified. Furthermore it discusses several weather services and possibilities of obtaining data, the establishment of a data infrastructure and possible solutions for an optimal data display.

The practical part of the thesis includes the implementation of a website, to twitter and display weather data, as well as the underlying Spatial Data Infrastructure (SDI). This WebGIS Application called „Weather Twitter“ was implemented with focus on open source software. As far as possible, user positioning is based on the W3C Geolocation API specification. Furthermore, the weather data is provided to an iPhone/Android application for further processing due to interoperability via a defined interface.





# Inhaltsverzeichnis

Abbildungsverzeichnis	ix
Tabellenverzeichnis	xi
Listenverzeichnis	xiii
Abkürzungsverzeichnis	xv
<b>1 Einleitung</b>	<b>1</b>
1.1 Zielsetzung . . . . .	1
1.2 Motivation . . . . .	1
1.3 Aufbau . . . . .	2
<b>2 Allgemeine Grundlagen</b>	<b>3</b>
2.1 Location Based Services (LBS) . . . . .	3
2.1.1 Begriffsdefinition . . . . .	3
2.1.2 Einleitung . . . . .	4
2.1.3 Technologie . . . . .	5
2.1.4 Web 2.0 - Die neue Generation des World Wide Web . . . . .	6
2.2 Geodaten als Grundlage eines LBS . . . . .	7
2.2.1 Geodateninfrastruktur (GDI) . . . . .	7
2.2.2 Die OpenStreetMap (OSM) . . . . .	8
2.2.3 Datenformate . . . . .	8
2.2.4 Räumliche Interpolation von Geodaten . . . . .	10
2.3 Positionsbestimmung im Zusammenhang mit LBS . . . . .	11
2.3.1 Netzwerkgestützte Technologien . . . . .	12
2.3.2 Gerätebasierende Technologien . . . . .	17
2.4 Standardisierungsgremien und Interoperabilität . . . . .	19
2.4.1 Open Geospatial Consortium (OGC) . . . . .	19
2.4.2 Ausgewählte Standards des OGC . . . . .	20
2.4.3 World Wide Web Consortium (W3C) . . . . .	24
2.4.4 Webentwicklung und ausgewählte Standards des W3C . . . . .	24
2.5 Objektrelationale Datenbanken (ORDB) und die Structured Query Language (SQL) . . . . .	26
<b>3 Konzeptionelle Grundlagen</b>	<b>27</b>
3.1 Wetterdienste und Datenbeschaffung . . . . .	27
3.1.1 Wetterdienste in Österreich . . . . .	28
3.1.2 Wetterdienste international . . . . .	29
3.2 Analyse und Aufbereitung der verschiedenen Wetterdaten für die Steiermark . . . . .	35
3.2.1 Vergleich der Temperaturdaten der verschiedenen Anbieter . . . . .	35

3.2.2	Wahl der Grundlagendaten . . . . .	39
3.2.3	Wetterdaten aus den Benutzerwettermeldungen . . . . .	39
3.2.4	Aufbereitung der Daten . . . . .	39
3.3	Datenverwaltung für den „Weather Twitter“ . . . . .	41
3.3.1	PostgreSQL . . . . .	41
3.3.2	PostGIS . . . . .	41
3.4	Werkzeuge zur grafischen Aufbereitung der Wetterdaten . . . . .	43
3.4.1	MapServer . . . . .	43
3.4.2	GeoServer . . . . .	47
3.5	Grafische Darstellung der Wetterdaten im Webbrowser mithilfe von OpenLayers . . . . .	47
3.5.1	Aufbau eines OpenLayers-Dokumentes . . . . .	48
3.5.2	Objekte in OpenLayers . . . . .	48
3.6	W3C Geolocation API – Positionierung mittels Webbrowser . . . . .	53
3.6.1	Einleitung . . . . .	54
3.6.2	Vorteile und Nachteile der verschiedenen Technologien . . . . .	54
3.6.3	Entwicklung der W3C Geolocation API . . . . .	57
3.6.4	Die W3C Geolocation API . . . . .	60
3.6.5	Vergleich zwischen IP Geolocation und W3C Geolocation API . . . . .	63
<b>4</b>	<b>Der „Weather Twitter“</b>	<b>65</b>
4.1	Grundgedanken bzw. Konzept des „Weather Twitter“ . . . . .	65
4.2	Aufbau des Webdienstes . . . . .	65
4.2.1	Startseite . . . . .	65
4.2.2	Home . . . . .	67
4.2.3	Wetter mitteilen . . . . .	67
4.2.4	Wetterdetails . . . . .	68
4.3	Datenbankinfrastruktur mittels PostgreSQL . . . . .	69
4.3.1	Wettertweets . . . . .	69
4.3.2	Wettervorschauinformation . . . . .	70
4.3.3	Temperaturreaster . . . . .	70
4.4	Einbinden von Mapfiles in den Webdienst . . . . .	70
4.5	Einbindung von OpenLayers in den Webdienst . . . . .	71
4.6	Positionsbestimmung im „Weather Twitter“ . . . . .	71
4.6.1	Positionierung am Desktop PC . . . . .	71
4.6.2	Positionierung am Handy bzw. Smartphone . . . . .	71
<b>5</b>	<b>Anwendung und Einsatz des „Weather Twitter“</b>	<b>73</b>
5.1	Anwendung der Applikation . . . . .	73
5.2	Interoperabilität mit der mobilen Applikation „wettR“ . . . . .	73
5.2.1	Definition der Schnittstellen zwischen „Weather Twitter“ und der mobilen Applikation . . . . .	74
5.2.2	Darstellung der Daten in der mobilen Applikation . . . . .	75
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>79</b>
6.1	Ausbau der Applikation . . . . .	79
6.2	Inhalte und Design . . . . .	80
6.3	Kommerzielle Nutzung . . . . .	80

# Abbildungsverzeichnis

2.1	Klassisches LBS Dreieck . . . . .	6
2.2	Gegenüberstellung von Nearest Neighbor und Inverse Distance Weighting . . . . .	11
2.3	Verfeinerung der zellbasierten Ortung im Mobilfunk . . . . .	14
3.1	Rasterdarstellung der Temperaturen für die Steiermark von World Weather Online	36
3.2	Vergleich der Temperaturen von Graz und nahen Städten . . . . .	37
3.3	Temperaturzeitreihen für Murau und Schladming . . . . .	38
3.4	Temperaturverteilung als Ausgabebild des MapServers . . . . .	46
3.5	Blaue Punkte zeigen die Abdeckung der Steiermark mit WLAN-Zugangspunkten	56
3.6	Darstellung des Positionsergebnisses verschiedener Geolocationdienste . . . . .	64
4.1	Startseite von „Weather Twitter“ . . . . .	66
4.2	Tweet für Graz . . . . .	66
4.3	Übersichtsseite „Home“ des „Weather Twitter“ . . . . .	67
4.4	Seite um dem System sein Wetter mitzuteilen . . . . .	68
4.5	Detailseite von „Weather Twitter“ . . . . .	69
4.6	Entity Relationship Modell des "Weather Twitter" . . . . .	70
5.1	Wettersvorschlag für die aktuelle Position . . . . .	75
5.2	Übertragung des vorherrschenden Wetters . . . . .	76
5.3	Darstellungsmöglichkeiten von „wettR“ . . . . .	76



# Tabellenverzeichnis

2.1	Daten der Whois-Datenbank des Regional Internet Registrys (RIRs) für die IP-Adresse des Webservers des Institutes für Geoinformation der TU Graz . . . . .	18
2.2	Parameter der GetCapabilities-Anfrage . . . . .	20
2.3	Parameter der GetMap-Anfrage . . . . .	21
2.4	Basismethoden an Geometrieobjekten . . . . .	22
2.5	Testmethoden für räumliche Beziehungen von zwei Geometrieobjekten (A und B)	22
2.6	Methoden, die räumliche Analysen von Geometrieobjekten unterstützen . . . . .	23
3.1	Vor- und Nachteile verschiedener Positionierungsmethoden . . . . .	57



# Listings

3.1	Extensible Markup Language (XML)-Abfrage beim Dienst World Weather Online	31
3.2	XML-Wetterabfrage beim Google-Wetterdienst	32
3.3	XML-Wettervorhersage vom norwegischen Dienst yr.no	34
3.4	Grundelemente des Mapfiles (MAP-Objekt)	44
3.5	Grundelemente des Mapfiles (LAYER-Objekt)	45
3.6	Das Mapobjekt in OpenLayers	48
3.7	Layerobjekt in OpenLayers	49
3.8	Layerobjekt mit Temperaturverteilung der Steiermark in OpenLayers	49
3.9	Controlobjekt in OpenLayers	50
3.10	Controlobjekt Geolocate zur Bestimmung der Position	50
3.11	Stylemapobjekt in OpenLayers	53
3.12	Styleobjekt in OpenLayers	53
3.13	Google Gears Request	58
3.14	Google Gears Server Response	59
3.15	<i>Geolocation</i> Objekt	61
3.16	<i>Position</i> Objekt	62
3.17	Koordinateninterface <i>Coordinate</i>	62
3.18	<i>PositionError</i> Objekt	62
5.1	XML-Anfrage der mobilen Applikation an den „Weather Twitter“	74





# Abkürzungsverzeichnis

<b>ACG</b>	Austro Control Group
<b>AJAX</b>	Asynchronous JavaScript and XML
<b>AoA</b>	Angle of Arrival
<b>AP</b>	Access Point
<b>API</b>	Application Programming Interface
<b>ASCII</b>	American Standard Code for Information Interchange
<b>ASOS</b>	Automated Surface Observation Stations
<b>BSS</b>	Basic Service Set
<b>BTS</b>	Base Transceiver Station
<b>CDMA</b>	Code Division Multiple Access
<b>CGI</b>	Cell Global Identity
<b>CGM</b>	Computer Graphics Metafile
<b>CoO</b>	Cell of Origin
<b>CSS</b>	Cascading Style Sheets
<b>CSV</b>	Comma-Separated Values
<b>DoD</b>	Department of Defense
<b>DSSS</b>	Direct Sequence Spread Spectrum
<b>E-OTD</b>	Enhanced Observed Time Difference
<b>EPSG</b>	European Petroleum Survey Group
<b>ERM</b>	Entity Relationship Modell
<b>ESRI</b>	Environmental Systems Research Institute
<b>EWKB</b>	extended Well Known Binary
<b>EWKT</b>	extended Well Known Text
<b>FHSS</b>	Frequency Hopping Spread Spectrum
<b>GDI</b>	Geodateninfrastruktur
<b>GIF</b>	Graphics Interchange Format
<b>GIS</b>	Geoinformationssystem
<b>GLONASS</b>	Globalnaja Nawigazionnaja Sputnikowaja Sistema
<b>GML</b>	Geography Markup Language
<b>GNSS</b>	Global Navigation Satellite System
<b>GPS</b>	Global Positioning System

**GPX** GPS Exchange Format  
**GSM** Global System for Mobile Communications  
**HTML** Hypertext Markup Language  
**HTTP** Hypertext Transfer Protocol  
**IANA** Internet Assigned Numbers Authority  
**IDW** Inverse Distance Weighting  
**IEEE** Institute of Electrical and Electronics Engineers  
**INSPIRE** Infrastructure for Spatial Information in the European Community  
**ISM** Industrial, Scientific and Medical  
**ISO** International Organization for Standardization  
**IP** Internet Protocol  
**ISP** Internet Service Provider  
**JPEG** Joint Photographic Experts Group  
**JSON** JavaScript Object Notation-Format  
**KML** Keyhole Markup Language  
**LBS** Location Based Services  
**LDS** Location Dependent Services  
**LIR** Local Internet Registry  
**LMU** Location Measurement Unit  
**MADIS** Meteorological Assimilation Data Ingest System  
**MLC** Mobile Location Center  
**MS** Mobile Subscriber  
**NASA** National Aeronautics and Space Administration  
**NAVSTAR** Navigation Satellite Timing and Ranging  
**NLP** Network Location Provider  
**NOAA** National Oceanic and Atmospheric Administration  
**NSS** Network and Switching Subsystem  
**OGC** Open Geospatial Consortium  
**ORDB** Objektrelationale Datenbank  
**ORDBMS** Objektrelationales Datenbank Managementsystem  
**ORF** Österreichischer Rundfunk  
**OSM** OpenStreetMap  
**OSS** Open-Source-Software  
**OSS** Operation Subsystem  
**PDA** Personal Digital Assistant  
**PHP** PHP: Hypertext Preprocessor  
**PNC** Pseudo Noise Code  
**PNG** Portable Network Graphics  
**PRN** Pseudo Random Noise

**PWS** Private Weather Station  
**RIPE NCC** Réseaux IP Européens Network Coordination Centre  
**RIR** Regional Internet Registry  
**RSS** Radio Subsystem  
**RSS** Received Signal Strength  
**RSS** Really Simple Syndication  
**SDI** Spatial Data Infrastructure  
**SFA** Simple Feature Access  
**SGML** Standard Generalized Markup Language  
**SLD** Styled Layer Descriptor  
**SQL** Structured Query Language  
**SRID** Spatial Reference System Identifier  
**SVG** Scalable Vector Graphics  
**TA** Timing Advance  
**TDoA** Time Difference of Arrival  
**ToA** Time of Arrival  
**TOPP** The Open Planning Project  
**UMN** University of Minnesota  
**UMTS** Universal Mobile Telecommunications System  
**UNESCO** United Nations Educational, Scientific and Cultural Organization  
**URL** Uniform Resource Locator  
**W3C** World Wide Web Consortium  
**WCS** Web Coverage Service  
**WFS** Web Feature Service  
**WGS84** World Geodetic System 1984  
**Wi-Fi** Wireless-Fidelity  
**WKB** Well Known Binary  
**WKT** Well Known Text  
**WLAN** Wireless Local Area Network  
**WMS** Web Map Service  
**WMO** World Meteorological Organization  
**WWW** World Wide Web  
**X3D** Extensible 3D  
**XHTML** Extensible Hypertext Markup Language  
**XML** Extensible Markup Language  
**ZAMG** Zentralanstalt für Meteorologie und Geodynamik



# Kapitel 1

## Einleitung

### 1.1 Zielsetzung

Der Inhalt der Masterarbeit „WITTER – Weather Twitter“ umfasst eine Recherchetätigkeit über Positionierungsmethoden von Mobiltelefonen und über internetbasierte Dienstleistungen, die Wetterinformation serviceorientiert bereitstellen.

Dabei spielt im Besonderen die Positionierung des Anwenders, sei es durch den Zugang über ein mobiles Gerät, oder einen Desktop Rechner eine wichtige Rolle. In diesem Zusammenhang wird vor allem die Positionierung in Verbindung mit einem Internetbrowser untersucht, die sich mithilfe der W3C Geolocation API realisieren lässt.

Die von verschiedenen Dienstleistern angebotene Wetterinformation wird auf ihre Genauigkeit und Zuverlässigkeit untersucht, um die Möglichkeit der Datenweiterverarbeitung abzuschätzen.

Weiters umfasst die Masterarbeit einen praktischen Teil, der im großen und ganzen vier wichtige Komponenten umfasst:

- Bereitstellung der Serviceinfrastruktur zur Verwaltung und Zusammenführung von Wetterdaten aus unterschiedlichen Quellen.
- Website zum Einmelden von Wetterdaten.
- Website zur Abfrage eines kontextsensitiven Wetterberichts.
- Bereitstellung einer Schnittstelle für die iPhone und Android Wetterapplikation „wettR“

### 1.2 Motivation

Wie ist das Wetter gerade? – Eine Frage die sich vermutlich jeder Mensch irgendwann schon einmal gestellt hat. Egal ob im Fernsehen, in der Zeitung oder im Internet, der Wetterbericht ist stets ein Thema. Wie wird's am Wochenende? – Die nächste Frage die beinahe jeden brennend interessiert. Wie wichtig das Thema Wetter für uns Menschen auch im Zusammenhang mit dem Internet ist, zeigt die Eingabe des Begriffs „Wetter“ in die Internet-Suchmaschine Google. Der

Begriff liefert 168 Millionen Ergebnisse. Der englische Begriff „weather“ gibt sogar 1,23 Milliarden Ergebnisse aus (Stand: Juni 2011). Warum sollte man dieses gemeinsame Interesse von vielen Menschen nicht nutzen, um eine Applikation zum gegenseitigen Informationsaustausch zu erstellen — den „Weather Twitter“. Man teilt dem System das aktuelle Wetter bzw. das Wetter in naher Zukunft mit, und kann gleichzeitig Informationen von anderen Personen zu deren Wetter abrufen. Neben dem aktuellen Wetter ist eine absolut wesentliche Information der Standpunkt der Person, die dem System die Wetterinformation mitteilt. Diese Positionsinformation kann auf diverse verschiedene Arten gewonnen werden, sei es über einen ins Mobiltelefon oder PDA integrierter GPS-Empfänger oder über die IP-Adresse eines Desktop Rechners. Diese Methoden werden in dieser Arbeit ebenso untersucht, wie diverse Wetterdienstleister, die ihre Daten im Internet in verschiedenen Formaten kostenlos zur Weiterverarbeitung und Darstellung anbieten. Somit werden die notwendigen Bausteine untersucht, um mittels OSS eine Web 2.0 – basierte LBS Dienstleistungswebsite zu erstellen.

### **1.3 Aufbau**

Die Masterarbeit „WITTER – Weather Twitter“ gliedert sich in sechs Kapitel. Nach dem ersten Kapitel, der Einleitung wird im zweiten Kapitel „Allgemeine Grundlagen“ der Grundstein für die darauffolgenden Abhandlungen gelegt. Das Kapitel beschäftigt sich mit einer allgemeinen Erläuterung der Aufgabenstellung, sowie den allgemeinen Grundlagen bei der Erstellung einer WebGIS-Anwendung. In Kapitel 3, den „Konzeptionellen Grundlagen“, wird die Herangehensweise an die Errichtung eines WebGIS betrachtet, wobei der Focus besonders auf den Anbietern von Wetterinformation, sowie der Positionierung mittels Geolocation liegt. In Kapitel 4 wird die praktische Umsetzung des „Weather Twitter“ näher vorgestellt. Dabei wird der Aufbau und die Gestaltung der einzelnen Webseiten erläutert. Kapitel 5 zeigt mögliche Anwendungsbereiche für den „Weather Twitter“, wobei vor allem die Interoperabilität mit der iPhone bzw. Android Applikation „wettR“ im Vordergrund steht. Schließlich und endlich werden im letzten Kapitel der Masterarbeit die durchgeführten Schritte zusammengefasst und weiterführende Aspekte in Form eines Ausblickes beleuchtet.

## Kapitel 2

# Allgemeine Grundlagen

In diesem Kapitel werden dem Leser einführende Informationen zu Location Based Services (LBS) sowie webbasierten Geoinformationssystemen (GIS) vermittelt. Außerdem werden die Standardisierungseinrichtungen Open Geospatial Consortium (OGC) und World Wide Web Consortium (W3C) beleuchtet, die sich zum Ziel gesetzt haben, die Interoperabilität bei Anwendungen mit GI-Bezug bzw. bei Webanwendungen größtmöglich zu gewährleisten. Des Weiteren wird in diesem Kapitel auf ausgewählte Standards und Technologien eingegangen, die im Rahmen der Masterarbeit verwendet wurden.

## 2.1 Location Based Services (LBS)

### 2.1.1 Begriffsdefinition

Unter LBS oder auch Location Dependent Services (LDS), versteht man mobile Dienste, die unter Zuhilfenahme von positionsabhängigen Daten dem Endbenutzer selektive Informationen bereitstellen oder Dienste anderer Art erbringen. Im Nachhinein werden einige Definitionen des Begriffes LBS von verschiedenen Quellen angeführt.

Die International Organization for Standardization (ISO) definiert die Begriffe LBS und LDS in ihrer Norm 19133 wie folgt:

„Location Based Services – LBS – service whose return or other property is dependent on the location of the client requesting the service or of some other thing, object or person.“

„Location Dependent Services – LDS – service whose availability is dependent upon the location of the client.“ (ISO (Zuletzt besucht: 27.April 2011))

Folgende Definition zu LBS liefert das OGC:

„A wireless-IP service that uses geographic information to serve a mobile user. Any application service that exploits the position of a mobile terminal“ (OpenGIS-Location-Service (Zuletzt besucht: 27.April 2011))

Eine weitere Definition von GISWIKI:

„Standortbasierte Dienste oder Location Based Services (LBS) bezeichnen Informationsangebote, welche den aktuellen Standort der Person bei der Informationsauswahl berücksichtigen. Weiterführend können auch individuelle Interessen der einzelnen Personen und zeitliche Aspekt berücksichtigt werden.“ (GISWIKI-LBS (Zuletzt besucht: 27.April 2011))

Eine sehr gute Definition liefert Jörg Blankenbach in seiner Dissertation „Handbuch der mobilen Geoinformation“:

„Standortbezogene Dienste (Location Based Services , LBS) sind mobile Mehrwertdienste, die dem Anwender mit den Mitteln der mobilen IT selektive Informationen oder Dienste auf Grundlage einer Lokalisierung in Echtzeit zur Verfügung stellen.“ (Blankenbach (2007))

### 2.1.2 Einleitung

In den letzten Jahren haben sich Location Based Services in einer breiten Öffentlichkeit etabliert. Neben den schon länger bekannten Anwendungen der Navigation (Positionsbestimmung, Routenplanung, Zielführung), des Flottenmanagements oder bestimmten Informationsdiensten (Gelbe Seiten, Veranstaltungen, Sehenswürdigkeiten) erfreuen sich vor allem LBS im Unterhaltungsbereich einer immer größeren Popularität. Hauptgrund dafür sind Veränderungen in der Mobilfunklandschaft. Sowohl die Mobilfunkbetreiber, wie auch die Mobilfunkkunden stehen in ständiger Weiterentwicklung. Die meisten Mobilfunkbetreiber bieten Verträge mit sehr großen Datenpaketen, zu immer günstigeren Preisen an, wodurch immer mehr Kunden auf multimediale Endgeräte zurückgreifen. So wurden laut Statistik Austria<sup>1</sup> im Jahr 2010 erstmals mehr Internetzugriffe via tragbarer Computer als per Desktopcomputer gezählt. Außerdem stieg die Nutzung von internetfähigen Mobiltelefonen vom Jahr 2005 bis ins Jahr 2010 von 2% auf 21% an. Darüberhinaus sind schon sehr viele Endgeräte am Markt, die sich sowohl dem Global Positioning System (GPS), als auch dem Wireless Local Area Network (WLAN) bedienen. Nicht zuletzt haben soziale Netzwerke wie Facebook oder Twitter mit eigenen Applikationen für Mobiltelefone den Boom rund um das mobile Internet weiter verstärkt. Neuerdings spielen Anwendungen, die den aktuellen Standort der Handynutzer verarbeiten eine immer größere Rolle. Im Jahr 2007 startete der Dienst „gowalla.com“, zwei Jahre später „foursquare.com“. Beide Dienste können als sogenannte standortbasierte soziale Netzwerke bezeichnet werden, die hauptsächlich für Mobiltelefone und Smartphones konzipiert sind. Dabei teilt der Nutzer seinen aktuellen Standort mit einem GPS-fähigen Gerät dem Dienst und somit seinen Freunden mit. „Foursquare.com“ lockt seine User durch bestimmte Belohnungen, so wird ein Nutzer zum Bürgermeister eines Ortes, wenn er an einem Standort die meisten Check-ins erreicht. Facebook reagierte im August 2010

<sup>1</sup>Statistik zum Internetzugang in Österreich [http://www.statistik.at/web\\_de/statistiken/informationsgesellschaft/ikt-einsatz\\_in\\_haushalten/index.html](http://www.statistik.at/web_de/statistiken/informationsgesellschaft/ikt-einsatz_in_haushalten/index.html)



mit der Einführung der Zusatzfunktion Facebook Places, um Usern die Möglichkeit zu geben anderen Usern mitzuteilen, wo sie sich gerade befinden. Diese Applikationen verfügen über ein enormes Marketingpotenzial<sup>2</sup> und erobern nach den USA 2010 nun auch Europa.

Neben der Möglichkeit, den Benutzer aufzufordern seinen eigenen Standort mitzuteilen, sind jedoch auch Möglichkeiten von Interesse seinen Standort automatisch abfragen zu können. Um dabei nicht von verschiedenen Softwarelösungen für das jeweilige Endgerät, oder deren Hardwarekomponenten abhängig zu sein, wird versucht die Position der User über den Webbrowser, wie zum Beispiel Firefox oder den Internet Explorer zu ermitteln. Dieses Ziel verfolgt die sogenannte Geolocation API, eine Schnittstelle zur Positionsermittlung, die in Abschnitt 3.6.1 näher beschrieben wird. Im Rahmen dieser Arbeit entstand ein LBS zur Einmeldung von Wetterdaten, der „Weather Twitter“, bei dessen Umsetzung auch die Anwendung der W3C Geolocation API untersucht wurde.

### 2.1.3 Technologie

Um standortbezogene Dienste anbieten zu können, müssen mehrere Akteure zusammenspielen:

- Das Endgerät z.B. Mobiltelefon oder Personal Digital Assistant (PDA).
- Der Positionsermittler, der die Ortung des Zielobjekts übernimmt. Entweder über ein endgerätebasierendes Verfahren (GPS), oder die netzbasierte Ortung (WLAN, Cell-ID, Internet Protocol (IP)).
- Ein Ortsbestimmer, der die Position mit einer Lokation (Adresse) in Verbindung bringt.
- Ein Dienstanbieter, der die Positionsinformation nutzt, um sie in seinem Dienst weiter zu verarbeiten.
- Sowie der Dienstanutzer, der per Mobiltelefon, PDA oder PC Anfragen an den Dienst stellt.

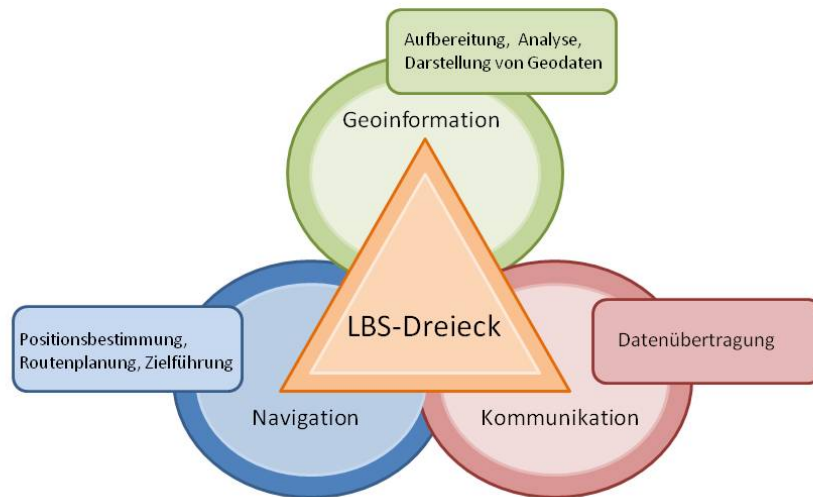
Diese Akteure bedienen sich der Technologien, die Location Based Services zu Grunde liegen, und sich in drei Hauptkomponenten unterteilen lassen:

- *Geoinformation*: Aufbereitung, Analyse und Darstellung von Geodaten.
- *Navigation*: Bestimmung und Verfolgung der Position eines mobilen Endgerätes oder einer Person.
- *Kommunikation*: Übertragung der Daten mittels Internet oder drahtloser Kommunikationstechnik (Funktechnik).

Diese drei Bestandteile eines LBS lassen sich auch über das klassische LBS Dreieck (siehe Abbildung 2.1) darstellen.

---

<sup>2</sup>Marketing mit Foursquare, Facebook Places, Gowalla & Co <http://www.salzburg.com/online/salzburg/salzbürger+wirtschaft/Marketing-mit-Foursquare-Facebook-Places-Gowalla-Co.html?article=eGmM0I8V5iw2HiXseQY835UBgAxxjhVtrejbjbs8&img=&text=&mode=>



**Abbildung 2.1** – Klassisches LBS Dreieck

Der Aspekt Geoinformation wird in Abschnitt 2.2 detaillierter beleuchtet. Die nähere Betrachtung der Punkte Navigation und Kommunikation folgt in Abschnitt 2.3.

#### 2.1.4 Web 2.0 - Die neue Generation des World Wide Web

Eng im Zusammenhang mit Location Based Services steht auch die Weiterentwicklung des World Wide Web. Die Welt des Internet hat sich in den letzten Jahren immer mehr in einen interaktiven Raum gewandelt. Noch vor einem Jahrzehnt standen wenige Bearbeiter und Veröffentlicher von Information vielen Nutzern gegenüber. Heute wird Information jedoch von sehr vielen Nutzern in das Internet eingespeist. Soziale Netzwerke, wie Facebook, Myspace oder Studivz, sowie verschiedene Blogs und Twitter sind nur einige Beispiele für das rasante Wachstum von Websites mit großen Mengen an von Benutzern erzeugtem Inhalt, dem „User Generated Content“.

Eine immer größere Rolle bei der Erzeugung und Darstellung von Information im Internet spielt dabei auch die Positionierung und Geoinformation. Wo befindet sich der User, der gerade meine Webseite besucht? Welche Daten kann ich vom User bekommen?

Dabei kommen immer öfter auch sogenannte Geobrowser wie Google Earth oder Bing Maps 3D zum Einsatz. Zudem sind viele Webseiten bereits mit diversen Erweiterungen, wie Google Maps, Bing Maps oder OpenStreetMap (OSM) ausgestattet, um den Nutzern Information mittels Kartenmaterial näherzubringen.

## 2.2 Geodaten als Grundlage eines LBS

Der Bereich Geoinformation stellt eine der drei großen Säulen von LBS dar. Dieser Teilbereich umfasst alle Schritte von der Datenauswahl über die Datenaufbereitung, bis zur Analyse, Verwaltung und Darstellung der Geodaten. Diese Daten können als Grundlage bei der Erstellung einer entsprechenden GI-Anwendung betrachtet werden. Geodaten stellen vereinfachte Abbilder von Objekten der realen Welt dar. Sie können wie folgt eingeteilt werden (vgl. Bartelme (2005)):

- *Rohdaten*: Über diverse Sensoren und Messgeräte direkt registrierte Daten.
- *Interpretierte Daten*: Sind infolge eines bestimmten Hintergrundwissens klassifiziert und mit einer Bedeutung hinterlegt.
- *Symbolisierte Daten*: Nach kartografischen Konventionen erzeugte Symbole.
- *Strukturierte Daten*: Daten, die nach bestimmten Anwendungsanforderungen zu vielfältigen Strukturen zusammengefasst wurden.

Nun stellt sich die Frage, welche Daten für eine entsprechende Anwendung geeignet sind, woher man die Daten bekommt, in welchen Datenformaten die Daten vorliegen können, und inwieweit eine entsprechende Weiterverarbeitung der Daten erfolgen kann.

### 2.2.1 Geodateninfrastruktur (GDI)

Um den Zugang zu Geodaten verschiedenster Quellen einfacher zu gestalten, wurde im Jahr 2007 von der europäischen Kommission eine Richtlinie verabschiedet um die Mitgliedsstaaten der Europäischen Union zu verpflichten stufenweise interoperable Geobasisdaten, sowie Geofachdaten zur Verfügung zu stellen – Das Projekt Infrastructure for Spatial Information in the European Community (INSPIRE)<sup>3</sup>.

In Österreich wird diese Richtlinie stufenweise umgesetzt. Ein in diesem Zusammenhang wichtiges Beispiel stellt das Geoportal „Geoland“<sup>4</sup> dar. Dabei handelt es sich um einen Geodatenverbund der von den österreichischen Bundesländern aufgebauten Geoinformationsabteilungen. Gemeinsam ermöglichen die einzelnen Länder einen offenen, freien und österreichweiten Zugriff auf Geodaten und Geo-Services. Dabei stehen die Services auch als OGC-konformer Web Map Service (WMS) zur Verfügung. Für den Aufbau des „Weather Twitter“ wurde unter anderem auf frei zur Verfügung stehende Shapefiles des GIS-Steiermark zurückgegriffen. Eine weitere Möglichkeit auf Geodaten zuzugreifen bietet das Portal „Geofabrik“<sup>5</sup>. Hierbei werden Möglichkeiten aufgezeigt, wie man OSM-Daten für bestimmte Zwecke einsetzen kann.

---

<sup>3</sup><http://www.inspire.gv.at>

<sup>4</sup>[http://www.geoland.at/images/stories/Geoland/presse/folder\\_geoland\\_2007.pdf](http://www.geoland.at/images/stories/Geoland/presse/folder_geoland_2007.pdf)

<sup>5</sup><http://www.geofabrik.de>

### 2.2.2 Die OpenStreetMap (OSM)

Nachdem Geoinformationen heutzutage noch immer selten frei erhältlich sind, und der weltweite Aufbau einer GDI sicher noch einiges an Arbeit und Zeit in Anspruch nehmen wird, wurde eine Möglichkeit gesucht, um weltweite Geodaten frei zur Verfügung stellen zu können. Mit der OpenStreetMap<sup>6</sup> wurde im Jahr 2004 ein Projekt ins Leben gerufen, welches das Ziel verfolgt, eine kostenlose und freie Weltkarte zu erstellen. Um diese Ziel zu realisieren, sammelt eine weltweite Community an Freiwilligen (auch *User* oder *Mapper* genannt) Daten über Straßen, Eisenbahnlinien, Flüsse, Wälder, Häuser und andere für Landkarten typische Elemente.

Aufgenommen werden diese Daten hauptsächlich mittels GPS-Empfängern, mit denen bestimmte Routen abgefahren werden. Die dabei erhaltenen GPS-Tracks im GPS Exchange Format (GPX)-Format bilden meist die Basis der kartographischen Darstellung. Doch auch Luftbilder, die zum Abzeichnen verwendet werden und lokales Wissen der Bearbeiter dienen häufig als Datengrundlagen für die Erstellung der OpenStreetMap (OSM). Um die aufgezeichneten Rohdaten weiterverarbeiten zu können, werden die Daten mit Zusatzinformationen (z.B. Namen, Straßentyp etc.) versehen. Je nach Bedarf können aus diesen Informationen verschiedene Karten generiert werden. So stehen auf der Webseite von OSM, bzw. anderen Seiten, Karten verschiedener Renderer und Stile zur Verfügung. Bekannte Renderer sind Mapnik und Osmarender. Zusätzlich stellt OSM auch diverse Spezialkarten (Fahrradkarte, Seekarte, Öffentlicher Verkehr, Ski- und Pistenkarte etc.) bereit.

### 2.2.3 Datenformate

Im Rahmen dieser Arbeit wurden mehrere Datenformate verwendet, die sowohl zur Darstellung (Rasterdaten und Vektordaten), als auch zum Austausch der Geodaten (XML (siehe Abschnitt 2.4.4)) verwendet wurden.

#### Vektordaten

Vektordaten dienen zur Verspeicherung von Punkten, Linien und flächenhaften Strukturen (Polygonen). Die flächenhaften Strukturen können als geschlossene Linienzüge dargestellt werden, während Linien wiederum, als Verbindung zweier Endpunkte (Knoten), die Koordinaten besitzen, angesehen werden können. Somit lassen sich alle Geometrien höherer Strukturen aus den Koordinaten von Punkten ableiten. Neben der geometrischen Information in Form der Koordinaten (Lage (2D)- und Höheninformation (3D)) beinhalten Vektordaten im Allgemeinen eine Topologie. Ausnahmen bilden Vektordaten, die nicht sauber aufbereitet sind (vgl. Spaghettitopologie). Des weiteren können Vektordaten mit einer Reihe von Attributen (Symbolik und Semantik, Farben, Linienstärke, Linienart, Flächenfüllung etc.) versehen sein. Der Vorteil von Vektordaten gegenüber Rasterdaten ist ein vergleichsweise geringer Speicherplatzbedarf und daraus folgend, kurze Rechenzeiten bei entsprechenden Analyseoperationen. Die Grundtypen bei der

---

<sup>6</sup><http://www.openstreetmap.org/>

Beschreibung von Vektorgeometrien sind:

- Punkt.
- Linie.
- Fläche.

Diese Grundtypen werden oft durch verschiedene Spezialtypen erweitert, so werden zum Beispiel in der Simple Feature Access (SFA)-Spezifikation des OGC, die in Abschnitt 2.4.2 beschrieben wird, auch aus mehreren Grundtypen zusammengesetzte Typen, wie Multigeometrien, sowie Geometriekollektion definiert.

## **Rasterdaten**

Im Gegensatz zu Vektormodellen gibt es bei Rastermodellen nur einen Entitätstyp, die sogenannte Rasterzelle oder Rastermasche. Die Rasterelemente sind in der Regel quadratisch geformt und von identischer Größe. Jedem Element wird ein bestimmtes Attribut zugewiesen, das einen bestimmten Sachverhalt darstellt (z.B. Farbwert, Höhenwert, Temperaturwert). Die Verspeicherung der Zellen, die in Anlehnung an die graphische Datenverarbeitung und Bildverarbeitung auch Pixel genannt werden, erfolgt über Koordinaten, deren Anordnung sich in Reihen und Spalten gliedern lässt. Bei der Wahl des richtigen Bildformates sollte immer die benötigte Auflösung, und damit der Bedarf an Speicherkapazität in Betracht gezogen werden. Anschließend werden einige gängige Rasterformate vorgestellt (Macrec (Zuletzt besucht: 23.Juni 2011)).

### **Graphics Interchange Format (GIF)**

Das Graphics Interchange Format ist mit JPEG derzeit noch das am häufigsten vorkommende Bildformat für Webseiten. Ein großer Vorteil von GIF-Bildern ist die Möglichkeit der verlustfreien Komprimierung, wodurch das Bild bei der Verkleinerung der Dateigröße keinen Qualitätsverlust erleidet. Weiters können GIFs animiert werden und auch die Transparenz des Hintergrundes wird vom Format unterstützt. Nachteil des Formates ist die geringe Farbtiefe, da nur 256 Farben ( $2^8$  bzw. 8 Bit) unterstützt werden, wodurch sich das Format speziell für Fotos nur sehr schlecht eignet.

### **Joint Photographic Experts Group (JPEG)**

Das JPEG (auch abgekürzt JPG)-Format wurde von der Joint Photographic Experts Group entwickelt, worauf auch die Bezeichnung „JPEG“ zurückgeht. Dieses Format eignet sich im Gegensatz zu GIF sehr gut für Fotos und ist das im Internet am weitesten verbreitete Bildformat. JPEG wird auch von den meisten Digitalkameras eingesetzt. Das JPEG-Format, stellt bis zu 16,7 Millionen Farben dar, was einer Farbtiefe von 24 Bit ( $2^{24}$ ) entspricht. Hierbei ist das JPEG-Format dem GIF klar überlegen, jedoch können JPEGs weder animiert, noch transparent dargestellt werden. Außerdem gehen bei der Komprimierung von JPEGs Information verloren.

## Portable Network Graphics (PNG)

PNG ist das modernste der drei Webformate (GIF, JPEG, PNG). Es bietet die Vorteile von GIF (Animation, Transparenz), lässt sich jedoch neben 8 Bit auch mit einer Farbtiefe von 24 Bit abspeichern. Außerdem ist auch die verlustfreie Komprimierung von PNG-Bildern möglich. Ein Nachteil des PNG-Formats ist die fehlende Unterstützung von einigen älteren Webbrowsern (bis Internet Explorer 6), wodurch es für Webseiten nur bedingt geeignet ist.

### 2.2.4 Räumliche Interpolation von Geodaten

Ein Problem bei vielen praktischen Arbeiten mit Geodaten besteht darin, dass die Daten für das betrachtete Untersuchungsgebiet nicht lückenlos zur Verfügung stehen. In der Anwendung sollten Werte jedoch in möglichst glatten Werteübergängen für alle Gebiete vorhanden sein. Aus Zeit- und Kostengründen können Werte jedoch nur an einigen wenigen Orten erfasst werden.

Diese Problematik lässt sich mit Hilfe einer räumlichen Interpolation in den Griff bekommen. Dafür legt man bei der Untersuchung von Werteoberflächen im Raum das Feld-Konzept zugrunde, wobei es bei der räumlichen Interpolation darum geht, ein komplettes Feld (im 2D-Fall die komplette Funktion  $f(x,y)$ ) aus einer endlichen Anzahl von Stützstellen zu ermitteln. Dafür muss man versuchen, mit Hilfe eines numerischen oder statistischen Modells, die stetige Veränderung der Werte im Raum nachzubilden.

In dieser Arbeit ist vor allem die zweidimensionale räumliche Interpolation von Bedeutung. Solche räumlichen Verteilungen sind an allen Punkten des Raumes (2D- oder 3D Oberfläche) definiert, wobei jedem Punkt ein bestimmter Attributwert zugewiesen wird. Damit werden flächenhaft verteilte diskrete Punkte in eine glatte Werteoberfläche abgebildet.

Die reale räumliche Verteilung der Lufttemperatur wird oftmals in einer stetigen Werteoberfläche ohne Sprungstellen dargestellt. Wenn man jedoch die räumliche Verteilung der Lufttemperaturen mittels eines Messnetzes von Wetterstationen erfassen will, so erhält man Werte in der Praxis nur für eine endliche, meist auch sehr geringe Anzahl an Messpunkten.

Eine Methode, die sich zur räumlichen Interpolation verwenden lässt, ist das Prinzip des gewichteten Mittelwertes. Dafür wird das Untersuchungsgebiet in ein regelmäßiges, feinmaschiges Gitternetz unterteilt, und an jedem Gitterpunkt wird lokal eine räumliche Interpolation mit Hilfe der benachbarten Stützstellen durchgeführt. Dadurch entsteht eine approximierte Werteoberfläche, die den Bedingungen in der Realität oft recht gut entspricht (Räumliche-Interpolation (Zuletzt besucht: 20.August 2011)).

## Inverse Distance Weighting (IDW)

Das IDW-Verfahren beruht auf dem Prinzip des gewichteten Mittelwertes. Es wird realisiert, indem man über das zu untersuchende Gebiet ein regelmäßiges feinmaschiges Gitternetz legt. Anschließend wird in jedem Gitterpunkt lokal eine räumliche Interpolation mithilfe der benachbarten Stützstellen durchgeführt. Das berechnete Gitternetz liefert eine approximierte Werteoberfläche,

die sehr gute Näherungen zu den realen Verhältnissen liefert. Die Auswahl der Nachbarpunkte, sowie die Gewichtung des Mittelwertes kann nach verschiedenen Kriterien festgelegt werden. Meistens erfolgt die Gewichtung anhand der Distanz zwischen Rasterzelle und Nachbarpunkt.

### Nearest Neighbor

Oftmals ist es ausreichend, einem Gitterpunkt einfach den Wert der nächstgelegenen Stützstelle zuzuweisen. Diese Interpolationsmethode führt zu einem „pixeligen“ Endergebnis, jedoch lässt sich der Algorithmus einfach implementieren und ist desweiteren auch sehr schnell. In Abbildung 2.2 werden die zwei vorgestellten Interpolationsverfahren gegenübergestellt, dabei sind die Unterschiede zwischen den Verfahren zu erkennen. Die linke Grafik zeigt die Nearest Neighbor Interpolation von drei 2D-Punkten mit Z-Wert (1 1 5, 2 7 10, 6 3 8), dabei erfolgt die Zuweisung des Farbwertes nach der nächstgelegenen Stützstelle. In der rechten Grafik werden die selben Punkte mit dem IDW-Verfahren interpoliert, was zu einem regelmäßig abgestuften Farbverlauf führt.

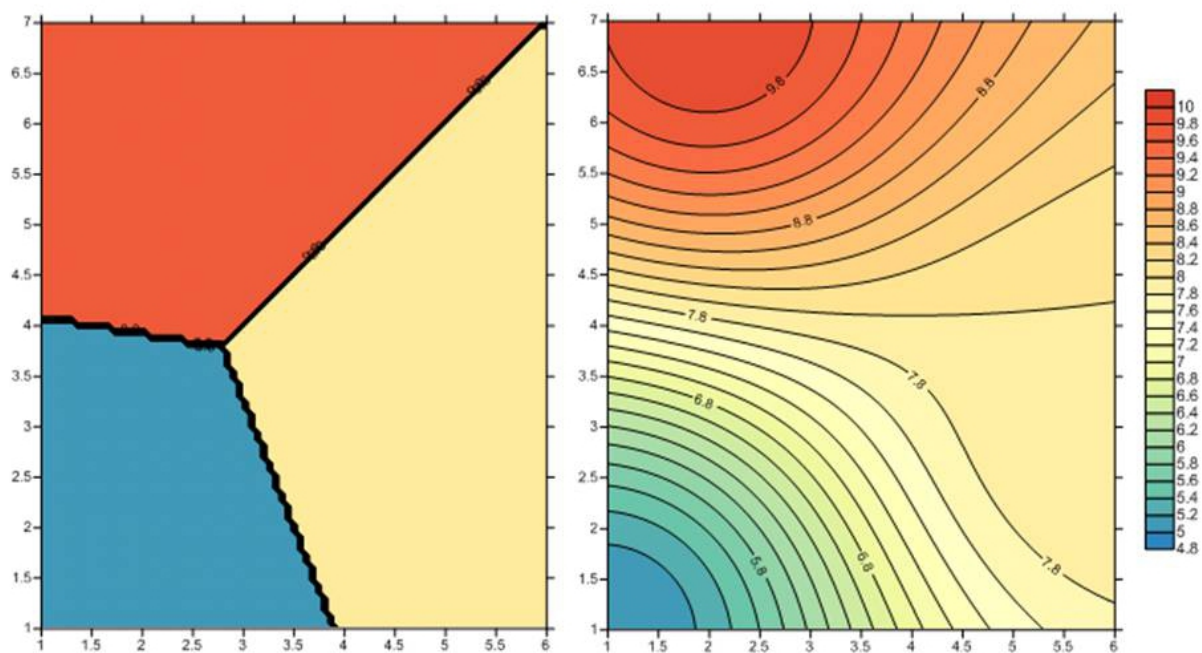


Abbildung 2.2 – Gegenüberstellung von Nearest Neighbor und Inverse Distance Weighting

## 2.3 Positionsbestimmung im Zusammenhang mit LBS

Die Information zur Positionsbestimmung im Zusammenhang mit LBS bezieht sich zu einem großen Teil auf Blankenbach (2007).

Standortbezogene Dienste sind nur dann sinnvoll, wenn die Position des Anwenders bekannt ist. Daher ist die Positionierung ein zentrales Element bei LBS. Die Position eines Anwenders

bzw. eines mobilen Endgerätes kann auf mehrere Arten ermittelt werden. Unterschieden wird dabei zwischen netzwerkgestützten Technologien (Zellulare Netzwerke, WLAN, Bluetooth) und gerätebasierenden Technologien (GPS). Ein weiteres Verfahren, auf das in dieser Arbeit noch näher eingegangen wird, ist die Positionierung mittels IP-Adresse bzw. zusammenfassend die sogenannte Geolocation API. Positionierungsmethoden der Indoorpositionierung (Ultraschall, Infrarot, Funk etc.) und andere weiterführende Positionierungsmethoden werden in dieser Arbeit nicht behandelt, da sie sich im Zusammenhang mit einem LBS zum Thema Wetter nicht unbedingt als sinnvoll erweisen. Im Wesentlichen richtet sich die Wahl der Positionierungsmethode immer nach der für die entsprechende Anwendung geforderten Positionsgenauigkeit und dem damit verbundenen Kostenaufwand.

### 2.3.1 Netzwerkgestützte Technologien

Nachdem LBS im ursprünglichen Sinn für mobile Clients entwickelt wurden, welche ohnehin ein Kommunikationsnetzwerk wie Global System for Mobile Communications (GSM), Universal Mobile Telecommunications System (UMTS) oder WLAN benötigen um diverse Informationen zu übertragen, erscheinen netzwerkgestützte Technologien als geeignete Positionierungsmethode für standortbezogene Dienste.

#### Zellulare Netzwerke

Mobiltelefone und andere mobile Devices kommunizieren in erster Linie über ein terrestrisches Mobilfunknetz, das GSM-Netz. Das GSM-Netzwerk hat eine hierarchische Systemarchitektur und besteht grundsätzlich aus drei Subsystemen (vgl. Blankenbach (2007)):

- Funk subsystem (Radio Subsystem (RSS)).
- Mobilvermittlungssystem (Network and Switching Subsystem (NSS)).
- Betriebs- und Wartungssystem (Operation Subsystem (OSS)).

Im Funk subsystem sind alle funkspezifischen Komponenten des Netzes enthalten. Vor allem die mobilen Endgeräte, die sogenannten Mobilstationen (Mobile Subscriber (MS)). Die Kommunikation zwischen den MS und dem GSM-Netzwerk erfolgt über die Basisstationen, die im GSM-Netz als Sende-/Empfangsstationen (Base Transceiver Station (BTS)) bezeichnet werden. Die BTS enthält alle funktechnischen Einrichtungen, wie Antennen, Signalverarbeitung und Verstärker für die Übertragung. Jede BTS ist für eine Zelle verantwortlich, wobei innerhalb eines GSM-Netzes eine vollständige Flächenabdeckung durch die Mobilfunkzellen angestrebt wird. Die Reichweite bei GSM wird durch die Ausdehnung der Zelle definiert, die von verschiedenen Faktoren (Bebauung, Hindernisse etc.) beeinflusst wird. Dabei wird zwischen folgenden Zellentypen unterschieden:

- Mikrozellen mit einem Radius von 100 - 1000 Metern.



- Makrozellen mit einem Radius bis zu 20 Kilometern.

Der urbane Raum unterteilt sich aufgrund der dichten Bebauung und einer damit verbundenen geringen Reichweite bei GSM in mehrere Mikrozellen. Im ländlichen Raum ist es meist ausreichend ein weniger dichtes Mobilfunknetz aufzubauen. Durch die größere Reichweite der Zellen wird daher von Makrozellen gesprochen. Neben der Reichweite spielt auch die erwartete Anzahl an MS eine entscheidende Rolle beim Aufbau eines Mobilfunknetzes, da jede BTS nur eine beschränkte Anzahl an Nutzern bedienen kann.

Das Mobilvermittlungssystem stellt das Kernstück von GSM dar, da es für die Verbindung zwischen dem Mobilfunknetz und herkömmlichen, öffentlichen Netzen, wie dem Telefonetz (auch Festnetz) verantwortlich ist.

Das Betriebs- und Wartungssystem enthält alle Funktionen, die für den zuverlässigen Betrieb und die Wartung des gesamten Systems notwendig sind, unter anderem die Authentifizierungszentrale zum Schutz der Identität der Teilnehmer und das Geräteidentifikationsregister, das der Netzbetreiber benutzt um die einzelnen Mobilfunkteilnehmer zu erkennen.

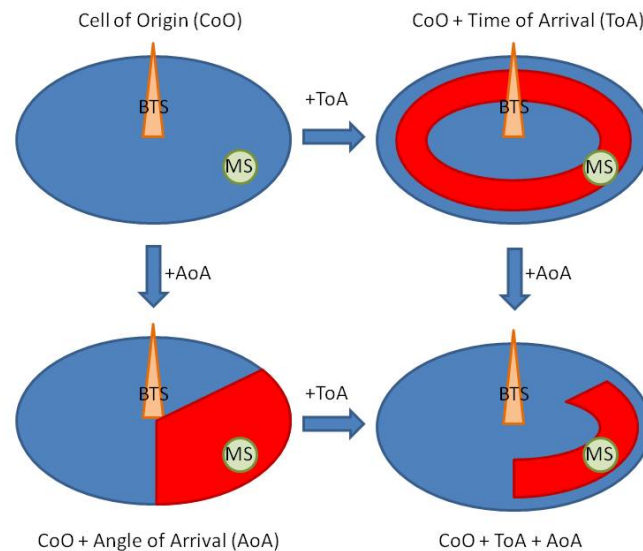
Der Aufbau eines Mobilfunknetzes ist hauptsächlich auf die Kommunikation zwischen den mobilen Endgeräten optimiert. Es wird versucht, die Zellen möglichst wenig miteinander zu überlappen, um mögliche Interferenzen zu minimieren. Daher ist die Struktur des Netzes meist ungünstig für die Positionierung, da meist keine Redundanzen in den Beobachtungen erreicht werden können. Diese Probleme lassen sich durch verschiedene Positionierungsmethoden minimieren.

Grundsätzlich wird bei der Positionierung in zellulären Netzwerken zwischen zellenbasierten Techniken, und Methoden der terrestrischen Radionavigation unterschieden.

Zellenbasierte Verfahren machen es sich zu Nutze, dass jede Mobilfunkzelle mit einer eindeutigen Zell-Kennung, der sogenannten Cell Global Identity (CGI) oder kurz Cell-ID, versehen ist. Somit ist es dem Netzbetreiber jederzeit bekannt, in welcher Zelle sich ein Nutzer aktuell befindet. Die tatsächliche Benutzerposition wird mit dem Zellmittelpunkt angegeben. Daher ist die Positionsgenauigkeit natürlich von der Größe der Zelle abhängig, in Mikrozellen lassen sich sehr gute Ergebnisse (unter 100 m) erzielen, während die Positionsgenauigkeit in den Makrozellen deutlich schlechter (mehrere Kilometer) wird. Das Verfahren der Cell-ID, auch Cell of Origin (CoO)-Verfahren genannt, lässt sich durch verschiedene Methoden verfeinern. Eine Möglichkeit ist, die Mobilfunkzelle in mehrere Sektoren ( $120^\circ$  oder  $180^\circ$ ) zu unterteilen. Das wird erreicht, indem man in der BTS mehrere gerichtete Antennen verwendet. Die Sektoren lassen sich in einzelne Segmente aufteilen. In welchem Segment sich der MS befindet, kann man mittels Timing Advance (TA), einer vereinfachten Laufzeitmessung bestimmen.

Neben den zellenbasierten Methoden werden zur Positionsbestimmung auch Verfahren der terrestrischen Radionavigation angewendet. Im Wesentlichen basieren diese auf Richtungs- und Laufzeitmessungen.

Beim Verfahren Angle of Arrival (AoA) wird ermittelt, aus welcher Richtung ein Signal beim MS eintrifft. Dazu werden bei den BTS Antennen mit Richtungscharakteristika (z.B. Anten-



**Abbildung 2.3** – Verfeinerung der zellbasierten Ortung im Mobilfunk

nenarrays) eingesetzt, die auch zur Verfeinerung des Cell-ID Verfahrens (Abbildung 2.3) eingesetzt werden können. Die Position des Empfängers wird anschließend über Richtungsschnitte bestimmt. Dabei sind Messungen zu mindestens zwei BTS notwendig. Dies ist jedoch aufgrund der geringen Überlappung der Mobilfunkzellen in den meisten Fällen nicht gewährleistet. Außerdem gilt beim AoA wie beim TA, dass die Genauigkeit der Lokalisierung durch Reflexion und Mehrwegeeffekte stark beeinflusst wird (vgl. Blankenbach (2007)).

Verfahren der Laufzeitmessung, Time of Arrival (ToA) beruhen auf dem Prinzip der Messung der Signallaufzeiten zwischen dem Endgerät und der Basisstation. Für eine 3D-Positionslösung ist die Messung zu vier Basisstationen notwendig. Anschließend kann über einen mehrfachen Bogenchnitt die Position der Mobilstation ermittelt werden. Die Messung der Signallaufzeit erfolgt in einer sogenannten Location Measurement Unit (LMU). Hierfür wird eine genaue Synchronisation der Uhren in den Base Transceiver Stationen vorausgesetzt. Als präziser Zeitgeber kann dafür beispielsweise GPS eingesetzt werden. Eine Erweiterung des ToA-Verfahrens ist die Bildung von Laufzeitdifferenzen der Signale in zwei Basisstationen. Dadurch lässt sich die Streckendifferenz zwischen Sender und Empfänger ermitteln und aus den Messungen zu mindestens drei BTS kann ein Hyperbelschnittverfahren angewandt werden. Dieses Verfahren wird auch als Time Difference of Arrival (TDoA) bezeichnet. Die Berechnung der Position erfolgt wie beim ToA im Mobile Location Center (MLC), um den die Infrastruktur des Mobilfunknetzes erweitert werden muss. Ein weiteres Verfahren der Differenzbildung ist das Enhanced Observed Time Difference (E-OTD). Hierbei erfolgt die Differenzbildung nicht im Netzwerk, sondern im Mobiltelefon, das von der Basisstation gesendete Zeitsignale auswertet und daraus die Laufzeitunterschiede ermittelt. Danach wird die Position entweder im Endgerät berechnet oder die Laufzeitunterschiede werden an das MLC übertragen, wo anschließend die Position berechnet wird. Nachteil dieses Verfahrens sind

erhebliche Kosten in der Infrastruktur, da einerseits eine Verdichtung des Netzes erfolgen, und andererseits einige Zusatzkomponenten bereitgestellt werden müssen (Blankenbach (2007)).

Zusammenfassend kann gesagt werden, dass TDoA bzw. E-OTD die am besten geeigneten Verfahren zur genauen Positionsbestimmung in zellularen Netzwerken sind, jedoch der finanzielle Aufwand in keiner Relation zu etwaigen Alternativen wie dem Einbau von GPS-Empfängern in mobile Endgeräte steht.

### **Wireless Local Area Network (WLAN)**

WLANs sind meist lokale Funknetze, die der drahtlosen Kommunikation zwischen mobilen Endgeräten (z.B. PDA oder Notebook) mit einem stationären Computernetzwerk dienen. Mittlerweile sind die meisten öffentlichen Gebäude und Bildungseinrichtungen mit Zugangspunkten, sogenannten Hot Spots (auch Access Point (AP)) zu einem WLAN ausgestattet um den Nutzern eine Verbindung zum Internet zu ermöglichen. Meistens wird ein WLAN über die Implementierung des Industriestandards für drahtlose Netzwerkkommunikation IEEE 802.11 realisiert. Dieser Standard wurde 1997 vom Institute of Electrical and Electronics Engineers (IEEE) hervorgebracht.

Zum Großteil benutzen Systeme dieses Standards das Industrial, Scientific and Medical (ISM)-Frequenzband für Hochfrequenz-Sendegeräte bei 2,4 GHz, welches in vielen Staaten gebührenfrei genutzt werden kann und keiner staatlichen Regulierung unterliegt. Zur Datenübertragung wurden grundsätzlich drei Verfahren vorgeschlagen:

- Frequenzsprungverfahren (Frequency Hopping Spread Spectrum (FHSS)).
- Direct-Sequence-Verfahren (Direct Sequence Spread Spectrum (DSSS)).
- Infrarot-Übertragung.

Der Aufbau von Netzen in der IEEE 802.11-Familie kann sowohl als Ad-hoc-Netzwerk zwischen zwei oder mehreren Geräten, wie auch als Infrastrukturnetzwerk erfolgen. Im Gegensatz zu Ad-hoc-Netzwerken, wo Geräte miteinander kommunizieren können, sobald sie sich im gegenseitigen Übertragungsbereich befinden, muss einem Infrastrukturnetzwerk ein gewisser Unterbau zu Grunde liegen. Bei diesen findet die Kommunikation eines mobilen Endgerätes mit einem Access Point statt. Die direkte Kommunikation zwischen einzelnen Teilnehmern ist dabei nicht vorgesehen. Die mobilen Endgeräte, die über eine WLAN-Funktion verfügen bzw. kompatibel zu IEEE 802.11 sind, werden innerhalb des Netzwerkes auch als Stations bezeichnet. Alle Stationen im Übertragungsbereich und der zuständige Access-Point werden Basic Service Set (BSS) genannt.

Der ursprüngliche Standard (802.11) nutzt wie bereits erwähnt das ISM-Frequenzband bei 2,4 GHz. Dieses wird zur Datenübertragung in 79 Kanäle mit einer Bandbreite von jeweils 1 MHz aufgeteilt. Beim FHSS-Verfahren werden nun durch die Vergabe einer bestimmten Sprungsequenz (Hopping-Sequence), die Kanäle nach einem Zufallsmuster gewechselt. Dadurch werden

Störungen kompensiert und es können mehrere drahtlose Netze am selben Ort errichtet werden. Die Trennung der Netze erfolgt durch unterschiedliche Sprungsequenzen. Die Vorteile der FHSS-Methode sind ein relativ geringer Kostenaufwand, sowie die energiesparende Umsetzung. Nachteil des Verfahrens, ist der enorme Verwaltungsaufwand, der durch die Frequenzsprünge entsteht. Dadurch wird die Nutzdatenrate reduziert, und der Wechsel zwischen mehreren Access Point (Roaming) verkompliziert sich. Außerdem ist die Reichweite der Technik begrenzt.

Alternativ zum FHSS wird das DSSS-Verfahren als Modulationsverfahren verwendet, wodurch größere Distanzen im Netz überbrückt, und höhere Datenübertragungsraten erzielt werden können. Hierbei werden die 79 Kanäle zu mehreren breitbandigen Kanälen zusammengefasst. Alle Teilnehmer in einem AP, die auf den selben Kanal eingestellt sind, müssen sich die Übertragungsrate teilen. Zur Übertragung des Signals wird ein Bandspreizverfahren nach Code Division Multiple Access (CDMA) mit einem Störcode (Pseudo Noise Code (PNC)) eingesetzt. Dadurch ist die Datenübertragung unempfindlicher gegen Störungen, allerdings wird die Implementierung komplexer und teurer. Das dritte Verfahren, das ursprünglich zur Datenübertragung bei WLAN-Netzen vorgesehen wurde ist die Infrarot-Übertragung, der in der Praxis jedoch keine Bedeutung zugemessen wird.

Umgangssprachlich werden Produkte, die auf Basis des IEEE 802.11 Standards entwickelt werden auch als Wireless-Fidelity (Wi-Fi) bezeichnet. Zertifiziert werden diese Produkte von der 1999 gegründeten Wi-Fi-Alliance, der derzeit ca. 200 Unternehmen (u.a. Apple, HTC, IBM, Microsoft, Nokia, Sony) angehören. Die Wi-Fi-Alliance hat es sich zum Ziel gemacht, den Betrieb von verschiedenen Wireless-Geräten im Sinne einer großen Interoperabilität zu gewährleisten.

Die Positionsbestimmung in einem WLAN-Netzwerk kann im Grunde durch zwei Methoden erfolgen:

- Signalstärkemessung.
- Szenenanalyse.

Bei der Signalstärkemessung wird der Signalpegel (Received Signal Strength (RSS)) gemessen, um daraus Rückschlüsse auf die Entfernung zum Access Point (AP) zu ziehen. Dafür müssen im Idealfall in einer Testphase für jeden AP die individuellen Parameter der Signalstärkefunktion ermittelt werden, da verschiedene Einflüsse (Dämpfung durch Hindernisse, Multipath) die Signalstärken beeinflussen. Anschließend kann die Position des mobilen Endgeräts aufgrund der ermittelten Entfernungen zu drei APs ermittelt werden. Eine andere Möglichkeit ist die Szenenanalyse. Hierbei werden die gemessenen Signalstärken zu mehreren Sendern betrachtet. Zu jedem gemessenen Ort wird in einer Datenbank ein Signalmuster als Referenzwert gespeichert. Aus dieser Datenbank kann anschließend für bestimmte Orte eines Objekts, zum Beispiel einen Punktraster, ein Signalstärkeprofil erstellt werden. Während der Messphase kann nun das gespeicherte Profil mit dem Messwert verglichen werden. In Form einer Korrelationsberechnung erfolgt dann die Positionsermittlung des mobilen Endgerätes (Blankenbach (2007)).

### Internet Protocol (IP)-Positionierung

Ein weiteres Verfahren, das auch zu den netzwerkbasierenden Technologien zu zählen ist, ist die Positionierung über die IP-Adresse. Diese Adresse wird jedem Computer zugewiesen, der an ein Computernetz (meistens Internet, aber auch Intranet) angebunden ist. Sie macht das Gerät damit für diverse Dienste erreichbar. Die Vergabe der IP-Adressen wird von der Internet Assigned Numbers Authority (IANA) geregelt. Während in den Anfangsjahren des Internet, die Vergabe der Adressen direkt von der IANA vorgenommen wurde, vergibt diese heute Blöcke an verschiedene regionale Vergabestellen RIR. Für Österreich, Deutschland, Schweiz und Liechtenstein ist die Réseaux IP Européens Network Coordination Centre (RIPE NCC) verantwortlich, die wiederum die ihr von der IANA zugeteilten Netze an lokale Vergabestellen Local Internet Registry (LIR) weitergibt. Die Aufgabe der LIR übernehmen in der Regel sogenannte Internet Service Provider (ISP). Welchem Endkunden welche IP-Adresse zugewiesen wurde, lässt sich in den Whois<sup>7</sup>-Datenbanken des RIRs ermitteln. Nach Abfrage der IP-Adresse 129.27.89.66 (Webserver des Institutes für Geoinformation der TU Graz) liefert die Datenbank die in Tabelle 2.1 dargestellten Daten.

Anhand dieser Informationen, vor allem aus dem Punkt *address*, lässt sich der IP-Adresse eine entsprechende Lokation bzw. in weiterer Folge eine Position beispielsweise in Form von World Geodetic System 1984 (WGS84) Koordinaten zuweisen. Nachteil dieser Methode ist, dass sich die Position meist nur auf eine zentrale Stelle (z.B. Internet Service Provider, Rechenzentrum eines Unternehmens) bezieht (Wikipedia-IP-Positionierung (Zuletzt besucht: 15. Mai 2011)).

### 2.3.2 Gerätebasierende Technologien

Neben den netzwerkbasierenden Technologien zur Positionierung von Nutzern in einem Location Based Service, können auch sogenannte gerätebasierende Technologien verwendet werden. Hierbei handelt es sich um Technologien, wo die Position netzunabhängig bestimmt werden kann. Hierzu müssen die Endgeräte jedoch mit zusätzlicher Hardware, z.B. GPS-Empfänger ausgestattet werden.

### Global Navigation Satellite System (GNSS)

Die Positionierung mittels GNSS ist mittlerweile die bekannteste Form der Positionierung im Rahmen von Location Based Services. Vor allem das US-Amerikanische Global Positioning System (GPS) ist heutzutage weit verbreitet. Das russische Globalnaja Nawigazionnaja Sputnikowaja Sistema (GLONASS) und das noch im Aufbau befindliche europäische GALILEO spielen in der breiten Öffentlichkeit (noch) eine untergeordnete Rolle.

Das Navigation Satellite Timing and Ranging (NAVSTAR)-GPS wurde vom Amerikanischen Verteidigungsministerium (Department of Defense (DoD)) in den 1980er Jahren für militärische Zwecke entwickelt. Erstmals zum Einsatz kam GPS Anfang der 1990er Jahre. Das Global Positio-

---

<sup>7</sup><http://www.db.ripe.net/whois>

**Tabelle 2.1** – Daten der Whois-Datenbank des RIRs für die IP-Adresse des Webserver des Institutes für Geoinformation der TU Graz

```
% Information related to '129.27.0.0 - 129.27.255.255'
inetnum:          129.27.0.0 - 129.27.255.255
netname:          TUGNET
descr:            Technische Universitaet Graz
descr:            Zentraler Informatikdienst
descr:            Steyrergasse 30
descr:            Graz, A-8010
country:          AT
admin-c:          DC329-RIPE
tech-c:           DC329-RIPE
mnt-by:           AS1113-MNT
status:           ASSIGNED PI
source:           RIPE # Filtered
person:           Dept. of Communications
address:          TU Graz/Zentraler Informatikdienst
address:          Steyrergasse 30
address:          A-8010 Graz
address:          AUSTRIA
phone:            +43 316 873 6390
fax-no:           +43 316 873 7699
e-mail:           tugnet@tugraz.at
nic-hdl:          DC329-RIPE
mnt-by:           AS1113-MNT
source:           RIPE # Filtered
% Information related to '129.27.0.0/16 AS1853'
route:            129.27.0.0/16
descr:            TUGNET
origin:           AS1853
mnt-by:           AS1853-MNT
source:           RIPE # Filtered
% Information related to '129.27.0.0/16 AS1113'
route:            129.27.0.0/16
descr:            TUGNET
origin:           AS1113
mnt-by:           AS1113-MNT
source:           RIPE # Filtered
```

ning System unterteilt sich in drei Segmente (vgl. Hofmann-Wellenhof u. a. (2007), Blankenbach (2007)):

- Das *Raumsegment* besteht aus mindestens 24 Satelliten, die sich auf sechs Bahnen mit einer Umlaufzeit von 12 Stunden in einer Höhe von 20200 Kilometern um die Erde bewegen. Die Neigung der Satellitenbahnen gegen die Äquatorebene (Inklination) beträgt  $55^\circ$ , wodurch für jeden Punkt der Erde mindestens vier Satelliten (Mindestanzahl für eine korrekte 3D-Positionsbestimmung) „sichtbar“ sein sollen.
- Das *Kontrollsegment* besteht aus fünf Monitorstationen und einer Leitstation, die ständig die Signale der Satelliten erfassen und auswerten. Dabei wird die exakte Umlaufbahn der einzelnen Satelliten überwacht und es werden die Navigationsnachrichten an die Satelliten übermittelt.
- Das *Nutzersegment* umfasst die GPS-Empfänger, die der Nutzer zur Positionsbestimmung verwendet.

Die Positionsbestimmung mittels GPS beruht auf dem Prinzip der Trilateration (Streckenmessung). Dabei werden die Strecken zwischen dem Empfänger und von mindestens vier Satelliten gemessen um die Empfängerposition eindeutig zu bestimmen. Die vier Satelliten werden benötigt um die vier Unbekannten der Empfängerposition (3 Koordinaten(X,Y,Z) und Uhrenfehler ( $\Delta t$ )) zu lösen. Zu diesem Zweck muss die Position der einzelnen Satelliten zum Zeitpunkt der Messung bekannt sein. Die GPS-Satelliten senden permanent Signale auf zwei Trägerfrequenzen(L1 und L2):

- L1: 1575.42 MHz: C/A-Code + Navigationsdaten (D-Code) sowie P-Code + D-Code.
- L2: 1227.60 MHz: P-Code + D-Code.

Diese Signale beinhalten die Navigationsdaten der Satelliten (Bahndaten, Uhrenkorrektur). Dabei werden die Signale über ein CDMA-Verfahren mit einem pseudozufälligen Pseudo Random Noise (PRN) Code moduliert, wobei dieser Code für jeden Satellit eindeutig ist. Um die Distanz zwischen Empfänger und Satellit zu ermitteln werden grundsätzlich zwei Verfahren eingesetzt: Die Codephasenmessung und die Trägerphasenmessung. Bei der Codephasenmessung wird mittels Ermittlung der Signallaufzeit die Distanz zwischen Satellit und Empfänger ermittelt. Bei der Trägerphasenmessung wird diese Raumstrecke über die Auswertung der Trägerphase und Lösung der Mehrdeutigkeiten (Ambiguitäten) ermittelt.

## 2.4 Standardisierungsgremien und Interoperabilität

### 2.4.1 Open Geospatial Consortium (OGC)

Das OGC ist eine internationale Vereinigung von derzeit 422 Mitgliedern (Stand: April 2011). Es setzt sich aus wissenschaftlichen Einrichtungen, Regierungsorganisationen und privatwirtschaftlichen Unternehmen zusammen, die das Ziel verfolgen, kostenlose und frei zur Verfügung

stehende Standards im Bereich der raumbezogenen Informationsverarbeitung festzulegen. Diese Standards sollten Entwicklern dabei helfen, komplexe räumliche Information und Anwendungen interoperabel und allgemein zugänglich zu machen. Im Rahmen dieser Arbeit wurden einige vom OGC definierte Standards verwendet, die in weiterer Folge näher beleuchtet werden (OGC-About (Zuletzt besucht: 13.April 2011)).

## 2.4.2 Ausgewählte Standards des OGC

### Web Map Service (WMS)

Ein OGC konformer WMS dient dazu, aus geographischen Informationen dynamisch eine Karte zu erstellen. In der WMS-Spezifikation wird eine Karte als Darstellung von geographischer Information in Form eines digitalen Bildes, passend um auf einem Computerbildschirm dargestellt zu werden, bezeichnet. In einem WMS erzeugte Karten werden allgemein in Rasterbildformaten wie PNG, GIF oder JPEG generiert. Fallweise jedoch auch als vektorbasiertes Grafikformat wie Scalable Vector Graphics (SVG) oder Web Computer Graphics Metafile (CGM).

Im internationalen Standard wurden drei Operationen definiert:

- GetCapabilities.
- GetMap.
- GetFeatureInfo (optional).

Die GetCapabilities-Anfrage liefert Metadaten über den verwendeten Service. In Tabelle 2.2 werden die möglichen Parameter dieser Anfrage aufgelistet.

**Tabelle 2.2** – Parameter der GetCapabilities-Anfrage

Anfrageparameter	Beschreibung
VERSION=version	Version des Service
SERVICE=WMS	Art des Services (verpflichtend)
REQUEST=GetCapabilities	Name der Anfrage (verpflichtend)
FORMAT=MIME_type	Ausgabeformat der Metadaten
UPDATESEQUENCE=string	Sequenznummer oder String zur Speicherkontrolle

Die zweite standardmäßig in einem WMS definierte Operation GetMap liefert eine Karte, deren geographische Parameter und Größe eindeutig definiert sind. Optional kann auch noch die dritte Operation GetFeatureInfo eingesetzt werden, die Information über in der Karte dargestellte Objekte zurückgibt.

Die einzelnen Operationen können in jedem Standardwebbrowser mithilfe eines Hypertext Transfer Protocol (HTTP)-Requests in Form des Uniform Resource Locator (URL) ausgelöst werden. Der Inhalt des URL hängt von der jeweiligen Operation ab, die der User ausführen will. In Tabelle 2.3 sind die Parameter der GetMap-Anfrage dargestellt (OGC-Spezifikation-WMS (Zuletzt besucht: 27.April 2011)).



**Tabelle 2.3** – Parameter der GetMap-Anfrage

Anfrage Parameter	Beschreibung
VERSION=version	Art der Anfrage
REQUEST=GetCapabilities	Name der Anfrage (verpflichtend)
LAYERS=layer_list	Angabe der Layer, getrennt durch Beistrich
STYLES=style_list	Styleinformationen zum jeweiligen Layer, getrennt durch Beistrich
CRS=namespace:identifizier	Angaben zum Koordinatensystem
BBOX=minx,miny,maxx,maxy	Koordinaten der Eckpunkte des Kartenausschnitts
WIDTH=output_width	Breite des Kartenausschnitts in Pixel
HEIGHT=output_height	Höhe des Kartenausschnitts in Pixel
FORMAT=MIME_type	Ausgabeformat der Metadaten
TRANSPARENT=TRUE FALSE	Hintergrundtransparenz
BGCOLOR=color_value	Hintergrundfarbe (Hexadezimal)
EXCEPTIONS=exception_format	Format, in dem Fehlermeldungen mitgeteilt werden (default=XML)
TIME=time	Zeit zu einem bestimmten Layer
ELEVATION=elevation	Höhe zu einem bestimmten Layer
Other sample dimension(s)	Einheit zusätzlicher auftretender Dimensionen (Neben Raum und Zeit)

### Web Feature Service (WFS)

Ein WFS spezifiziert einen Standard, der den internetgestützten Zugriff auf geografische Objekte erlaubt. Der WFS beschränkt sich hierbei auf Vektordaten, wie sie in Datenbanken abgelegt werden können. Man unterscheidet zwischen Services die den Zustand der Applikation und deren Objektklassen erkunden (Discovery Operations) und Services, die man einsetzt um Abfragen auf die der Applikation zugrunde liegenden Datenbanken zu tätigen (Query Operations). Dabei werden 11 verschiedene Operationen unterstützt (OGC-Spezifikation-WFS (Zuletzt besucht: 27.April 2011)).

### Simple Feature Access (SFA)

Die dritte OGC-Spezifikation, die in dieser Masterarbeit immer wieder vorkommt, ist die OpenGIS SFA-Spezifikation, auch ISO 19125. In dieser Spezifikation wird eine plattformunabhängige Umgebung für einfache Geometrieobjekte definiert. Die Klasse Geometrie unterteilt sich in die Unterklassen Punkt, Kurve, Oberfläche und Geometriesammlung (Point, Curve, Surface und GeometryCollection). Jedes geometrische Objekt ist mit einem räumlichen Koordinatensystem verbunden, indem der Koordinatenraum des Objektes definiert wird. Das erweiterte Geometrie-Modell hat spezielle 0-, 1- und 2-dimensionale Klassen (MultiPoint, MultiLineString und MultiPolygon), die eingesetzt werden, um Geometrien, die aus mehreren Punkten, Linien oder Polygonen bestehen zu modellieren. Neben den verschiedenen Geometrietyten wurden in der SFA-Spezifikation auch verschiedene Test- und Analysemethoden definiert. Die Basismethoden, die z.B. mittels einer SQL-Anweisung in einer Objektrelationalen Datenbank (ORDB) abgefragt werden können finden sich in Tabelle 2.4.

Neben den Basismethoden, die hauptsächlich Informationen zu einzelnen Geometrieobjekten

**Tabelle 2.4** – Basismethoden an Geometrieobjekten

Methoden	Beschreibung
Dimension (): Integer	Dimension des Geometrieobjekts
GeometryType (): String	Name des Geometrietyps, dessen Instanz die entsprechende Geometrie ist
SRID (): Integer	Ausgabe der Spatial Reference System Identifier der Geometrie
Envelope (): Geometry	Minimales Umgebungsrechteck der Geometrie (Boundingbox) als Geometrie
AsText (): String	Geometrie wird im Well Known Text (WKT)-Format ausgegeben
AsBinary (): Binary	Geometrie wird im Well Known Binary (WKB)-Format ausgegeben
IsEmpty (): Integer	Liefert 1(TRUE), falls die Geometrie leer ist
IsSimple (): Integer	Liefert 1(TRUE), wenn keine Anomalitäten (z.B. selbstschneidende Geometrie) auftreten
Is3D (): Integer	Liefert 1(TRUE), falls das Objekt z-Koordinatenwerte enthält
IsMeasured (): Integer	Liefert 1(TRUE), wenn das Objekt Koordinatenwerte in der Einheit Meter hat
Boundary (): Geometry	Liefert den Rand des Geometrieobjektes (ohne Aussparungen)

liefern, lassen sich auch Methoden anwenden, um die Beziehungen zwischen zwei oder mehreren Geometrien abzufragen (siehe Tabelle 2.5).

**Tabelle 2.5** – Testmethoden für räumliche Beziehungen von zwei Geometrieobjekten (A und B)

Methoden	Beschreibung
Equals (B : A)	TRUE, wenn die Geometrie A der Geometrie B gleicht
Disjoint (B : A)	TRUE, falls Geometrie A von Geometrie B nicht geschnitten wird
Intersects (B : A)	TRUE, falls Geometrie A von Geometrie B geschnitten wird
Touches (B : A)	TRUE, wenn Geometrie A Geometrie B berührt
Crosses (B : A)	TRUE, wenn Geometrie A Geometrie B schneidet
Within (B : A)	TRUE, wenn Geometrie A vollständig in Geometrie B liegt
Contains (B : A)	TRUE, wenn Geometrie A vollständig in Geometrie B liegt
Overlaps (B : A)	TRUE, wenn Geometrie A sich mit Geometrie B überlappt

Neben den räumlichen Beziehungen von Geometrieobjekten wurden in der SFA-Spezifikation auch Methoden definiert um räumliche Analysen von Geometrieobjekten durchzuführen (vgl. Tabelle 2.6) (OGC-Spezifikation-SFS (Zuletzt besucht: 27.April 2011)).

### Keyhole Markup Language (KML)

Eine weitere OGC-Spezifikation ist KML. Die Keyhole Markup Language ist eine Sprache zur Kodierung und zum Transport von geografischen Daten, die in einem Earthbrowser dargestellt werden sollen. Entwickelt wurde die Sprache von der Firma Keyhole, die im Jahr 2004 von Google übernommen wurde. KML basiert auf dem XML-Standard (siehe Abschnitt 2.4.4), das bedeutet, dass eine tag-basierte Struktur mit verschachtelten Elementen und Attributen zum Austausch der Informationen verwendet wird.

KML wurde dem Open Geospatial Consortium von Google zur Standardisierung vorgeschlagen. Ziel von Google war es einen Standard zu schaffen, der folgende vier Ziele erreichen sollte:

- Es sollte eine international gültige Sprache zur Beschreibung und Visualisierung von web-basierten Karten (online oder mobile) und von Earth Browsern (Google Earth, Bing Maps,

**Tabelle 2.6** – Methoden, die räumliche Analysen von Geometrieobjekten unterstützen

Methoden	Beschreibung
Distance (B : A)	Kürzeste Distanz zwischen Geometrie A und Geometrie B
Buffer (Distanz) : Geometrie	Erzeugt eine um die Bufferdistanz vergrößerte Geometrie zum Ausgangsobjekt
ConvexHull () : Geometrie	Erzeugt eine konvexe Hülle über die Eingangsgeometrie
Intersection (B : A)	TRUE, wenn Geometrie A Geometrie B berührt
Union (B : A)	Vereinigung von Geometrie A und Geometrie B
Difference (B : A)	Differenzbildung von Geometrie A und Geometrie B
SymDifference (B : A)	Symmetrische Differenz von Geometrie A und Geometrie B

National Aeronautics and Space Administration (NASA) World Wind) geschaffen werden.

- KML sollte mit internationalen Standards und Praxiserfahrungen abgestimmt sein. Außerdem sollte die Erlernbarkeit der Umsetzung von Applikationen, und die Interoperabilität mit Earth Browsern gefördert werden.
- Das OGC und Google sollten bei der Umsetzung von KML als Standard eng zusammenarbeiten. Vor allem die OGC-Community sollte stark in den Umsetzungsprozess miteingebunden und von Fortschritt und Themen informiert sein.
- Die OGC Standardisierung sollte das „Gesamtsystem“ KML sicherstellen. Vor allem die Abwärtskompatibilität erscheint hier sehr wichtig.

KML hat sich mittlerweile in einer breiten und variantenreichen Community durchgesetzt und wird einerseits von gewöhnlichen Usern benutzt um KML Placemarks ihrer Häuser oder geplante Fahrradrouen zu mappen. Wissenschaftler verwenden KML um detaillierte Kartierungen von Ressourcen, Modellen oder Trends wie Vulkaneruptionen oder Erdbebenaktivität zu visualisieren. Darüberhinaus sind noch eine Menge an anderen Anwendungsbereichen von KML vorhanden. Beispielhaft können international anerkannte Organisationen wie National Geographic oder die United Nations Educational, Scientific and Cultural Organization (UNESCO) genannt werden, die ebenfalls KML benutzen um vielerlei ihrer Datenbestände darzustellen.

KML ist eine XML Sprache, die sich auf die Geografische Visualisierung und Beschreibung von Karten und Bildern konzentriert. Neben der grafischen Präsentation der Daten wird das Augenmerk vor allem auch auf das Kontrollieren der Usernavigation gelegt, so kann auch berücksichtigt werden aus welchem Blickwinkel der User die Daten präsentiert bekommt.

Aus dieser Perspektive ist KML ein Gegensatz zu den bereits beschriebenen Standards WMS und WFS. Derzeit werden auch von KML einige Geometrielemente unterstützt, zum Beispiel *Point*, *Linestring*, *Linear Ring* und *Polygon*. Für die Zukunft ist angedacht, KML und Geography Markup Language (GML) noch besser zu harmonisieren, das betrifft vor allem den Umgang mit der Geometrieinformation. OGC-Spezifikationen, wie Context und der Styled Layer Descriptor (SLD), sollten hier berücksichtigt werden (OGC-Spezifikation-KML (Zuletzt besucht: 27.April 2011)).

### 2.4.3 World Wide Web Consortium (W3C)

Das W3C ist eine Internationale Community, die sich ähnlich dem OGC aus Mitgliedsorganisationen (Regierungsorganisationen, Unternehmen, Universitäten), aber auch einer Gruppe fixer Mitarbeiter rund um die ganze Welt zusammensetzt. Ziel der Organisation ist die Schaffung von Richtlinien um das Langzeitwachstum des Internets zu sichern. Dabei spielen der freie Zugang zum Internet, als Kommunikationsmedium, zum Wissensaustausch und zu kommerziellen Zwecken eine große Rolle, unabhängig davon welche Hardware, Software, Netzwerkinfrastruktur bzw. Nationalität, Kultur oder Sprache der Nutzer sein eigen nennt. Des weiteren hat sich die Anzahl an verschiedenen Endgeräten, die das Internet als Medium benutzen in den letzten Jahren deutlich erhöht. Handys, Smartphones, PDAs und diverse andere Systeme bieten den Zugriff zum World Wide Web (WWW). Unabhängig vom Gerät sollte die Information in gleicher Form zur Verfügung stehen. Außerdem hat sich das Internet in den letzten Jahren von einem „read-only“ Tool zu einer Welt des regen Austausches gewandelt. Soziale Netzwerke und andere personalisierte Dienste haben sehr viele Menschen mehr zum Internet gebracht. Die Standards des World Wide Web Consortium, z.B. Hypertext Markup Language (HTML), Cascading Style Sheets (CSS) oder PHP: Hypertext Preprocessor (PHP) haben diese Entwicklung mehr und mehr unterstützt. In Zukunft wird ein reibungsloser Datenaustausch eine immer größere Rolle spielen, um die riesigen verlinkten Datenmengen des Internets vorteilhaft zu nutzen (W3C-Mission (Zuletzt besucht: 09.Mai 2011)).

### 2.4.4 Webentwicklung und ausgewählte Standards des W3C

In folgendem Abschnitt werden einige ausgewählte Standards des W3C wie z.B. HTML, CSS und XML näher betrachtet. Außerdem wird auf weitere Standards bzw. Sprachen (Javascript, Asynchronous JavaScript and XML (AJAX) sowie PHP), die in der Webentwicklung gebräuchlich sind eingegangen.

#### **Hypertext Markup Language (HTML) und Cascading Style Sheets (CSS)**

HTML ist zusammen mit CSS die Kerntechnologie um Webseiten zu implementieren. Mit der Auszeichnungssprache HTML wird die Struktur einer Webseite definiert. Hierbei können Online-Dokumente mit Headings, Text, Tabellen, Listen, Photos, Videos und anderen Elementen erstellt werden. Die moderne Variante von HTML ist Extensible Hypertext Markup Language (XHTML). Hierbei können alle Element aus HTML verwendet werden, jedoch weicht die Syntax leicht von der ursprünglichen Syntax ab. Mit der Beschreibungssprache CSS wird die visuelle Darstellung einer Webseite gestaltet. Farben, Schriften und Positionierung der Elemente können festgelegt werden. Außerdem kann man die Seiten auf das jeweilige Ausgabegerät anpassen. Um die Wartung von Webseiten einfacher zu machen sollten die technische Umsetzung einer Webseite (HTML, PHP, Javascript etc.) und deren graphische Gestaltung (CSS) strikt voneinander getrennt werden (W3C-HTML-CSS (Zuletzt besucht: 09.Mai 2011)).

### **Extensible Markup Language (XML)**

In den letzten Jahren hat sich neben HTML mit seinen Cascading Style Sheets ein weiterer W3C-Standard immer mehr in den Vordergrund gedrängt, die Extensible Markup Language.

XML ist ein einfaches textbasiertes Format zur strukturierten Darstellung von Information, beispielsweise Dokumente, Daten, Bücher, Transaktionen und vieles mehr. Entwickelt wurde XML aus dem älteren Standard Generalized Markup Language (SGML)-Format.

Einer der größten Vorteile der Sprache ist die Interpretierbarkeit sowohl von Mensch als auch von Maschine, wodurch sich XML sehr gut als Datenaustauschformat zwischen Computern, aber auch zwischen Mensch und Computern eignet. Die Syntax von XML ist ähnlich der HTML-Syntax, jedoch sind die Regeln der Extensible Markup Language wesentlich strikter.

Mittlerweile existiert bereits eine große Anzahl an Sprachen, die sich der Grundlagen von XML bedienen. In dieser Arbeit spielen neben dem ursprünglichen XML vor allem XHTML (als XML-konformes HTML) sowie die Keyhole Markup Language, als Sprache zur Darstellung von geografischen Elementen in Earth Browsern eine größere Rolle. Jedoch bedienen sich einige weitere Sprachen im Geoinformationsbereich der XML-Syntax. Unter ihnen sind GML, das GPX, City GML sowie die OSM oder die 3D-Modellierungssprache Extensible 3D (X3D) und das 3D-Datenaustauschformat Collada. (W3C-XML (Zuletzt besucht: 09.Mai 2011))

### **Javascript und Asynchronous JavaScript and XML (AJAX)**

Der dritte wichtige Webstandard in dieser Arbeit ist die Skriptsprache JavaScript, und in weiterer Folge auch ihre Erweiterung AJAX.

JavaScript ist eine Skriptsprache, die im Unterschied zu PHP hauptsächlich clientseitig eingesetzt wird. Hierbei wird das Skript direkt im Browser ausgeführt.

AJAX ist keine neue Programmiersprache, sondern bedient sich der bereits bestehenden Standards Javascript und XML. Es wurde entwickelt, um Daten mit einem Server auszutauschen, und Teile einer Webseite zu erneuern, ohne die gesamte Webseite neu laden zu müssen. Dadurch können Ladezeiten von Webseiten deutlich verringert werden, wodurch sich die Benutzerfreundlichkeit von Webseiten enorm verbessert.

### **PHP: Hypertext Preprocessor (PHP)**

PHP ist zwar kein W3C-Standard, jedoch ein sehr wichtiges Entwicklungstool für beinahe jede WebGeoinformationssystem (GIS)-Applikation.

PHP ist eine weit verbreitete und für den allgemeinen Gebrauch bestimmte Open-Source-Skriptsprache, welche speziell für die Webprogrammierung geeignet ist und in HTML eingebettet werden kann. PHP ist auf serverseitige Skripte fokussiert, das heißt ein auszuführendes Skript wird nicht vom Webbrowser interpretiert, sondern von einem Interpreter auf dem Webserver (PHP-Manual (Zuletzt besucht: 09.Mai 2011)).

## 2.5 Objektrelationale Datenbanken (ORDB) und die Structured Query Language (SQL)

Eine Objektrelationale Datenbank (ORDB) stellt das Bindeglied zwischen relationalen Datenbanken und objektorientierten Datenbanken dar. Während sich relationale Datenbanken besonders für große Mengen an einfach strukturierten Daten eignen, so wurden objektorientierte Datenbanken für die Beschreibung von komplexen Informationen entwickelt. Eine ORDB bedient sich nun der Vorteile von beiden Systemen. Sowohl die einfache tabellarische Aufbereitung der Daten in relationalen Datenbanken, wie auch die oftmals vorteilhafte Möglichkeit Objekte mitsamt ihrer Vielzahl an Attributen als ein Ganzes ansprechen zu können, wie es in objektorientierten Datenbanken möglich ist. Da diese Vorteile natürlich auch für GI-Anwendungen von Bedeutung sind werden viele bestehende Datenbanksysteme durch entsprechende Erweiterungen aufgerüstet, die den Anforderung der Geoinformation entsprechen. So wurde auch das in dieser Arbeit verwendete Datenbankmanagementsystem PostgreSQL (siehe Abschnitt 3.3.1) mithilfe der Erweiterung PostGIS (siehe Abschnitt 3.3.2) zu einem geeigneten Werkzeug für GI-Anwendungen aufgerüstet.

Die Kommunikation zwischen einem Datenbanksystem und seinen Benutzern erfolgt mithilfe einer Datenbanksprache wie z.B. SQL. Die Structured Query Language ist eine Datenbanksprache zur Abfrage, Definition und Bearbeitung von Daten in relationale Datenbanken. Die Sprache wurde ab 1972 von einer Gruppe von IBM-Mitarbeitern entwickelt und seither ständig erweitert sowie an neue Entwicklungen angepasst. Ein großer Vorteil der Sprache ist die Systemunabhängigkeit, wodurch SQL in mehreren Datenbanklösungen zum Einsatz kommt.

## Kapitel 3

# Konzeptionelle Grundlagen

Nachdem in Kapitel 2 vor allem auf die Theorie, die sich hinter einer WebGIS-Anwendung verbirgt eingegangen wurde, werden in Kapitel 3 die Werkzeuge näher beschrieben, die notwendig sind um solche Applikation erstellen zu können. Neben der Erschließung von möglichen Datenquellen für den „Weather Twitter“ und deren Untersuchung, ist auch der Aufbau einer Datenbankinfrastruktur notwendig. Hierfür wurde das Datenbanksystem PostgreSQL mit seiner Erweiterung PostGIS verwendet. Für die Positionsermittlung im Rahmen der Applikation wurde die W3C Geolocation API untersucht. Weiters werden die verwendeten Kartendarstellungstools wie MapServer und OpenLayers näher betrachtet.

### 3.1 Wetterdienste und Datenbeschaffung

Zur Realisierung einer WebGIS-Applikation werden Daten benötigt, die in den Webdienst eingebunden werden können. Im Falle der Wetterapplikation „Weather Twitter“ werden Wetterdaten benötigt um den Usern eine gewisse Grundinformation zur Verfügung stellen zu können. Dafür sind Temperaturdaten und eine Beschreibung der aktuellen Wettersituation notwendig. Die Beschaffung dieser Daten kann über diverse Dienste in verschiedenen Datenformaten getätigt werden, wobei in dieser Arbeit nur kostenlose Dienste berücksichtigt werden. Außerdem sollten die Daten in einem möglichst allgemeinen Format zur Verfügung stehen, das einfach weiterzuverarbeiten ist, vorzugsweise XML. Desweiteren ist ein Ziel dieser Arbeit die Umsetzung eines Dienstes, der eine möglichst regionale Wetterinformation für das Bundesland Steiermark anbietet. Wo kann man sich solche Daten besorgen? – Der erste Weg führt zu den allgemein bekannten Anbietern von Wetterinformation. In Österreich sind das unter anderem der Österreichische Rundfunk<sup>1</sup>, die Wetterplattform [wetter.at](http://wetter.at)<sup>2</sup> oder die Zentralanstalt für Meteorologie und Geodynamik<sup>3</sup>. Der zweite Gedanke ist die Suche nach diversen Wetteranbietern im Internet. Beispielsweise liefert die bekannte Internetsuchmaschine Google bei der Eingabe des Begriffes „Wetter“ das aktuelle Wetter in der näheren Umgebung. Woher weiß Google wo ich bin? Woher

---

<sup>1</sup><http://wetter.orf.at>

<sup>2</sup><http://wetter.at>

<sup>3</sup><http://zamg.ac.at>

weiß Google mein Wetter? – Weitere Fragen, die in den folgenden Abschnitten dieses Kapitels nach und nach beantwortet werden.

### 3.1.1 Wetterdienste in Österreich

In diesem Abschnitt werden drei wichtige österreichische Wetterinformationsdienste betrachtet. Dabei ist vor allem der Bezug von Wetterdaten in einem geeigneten Datenformat von Interesse. In Österreich hat sich auf den ersten Blick kein geeigneter Anbieter von kostenlosen Wetterdaten gefunden, jedoch konnte dieses Problem über Umwege gelöst werden.

#### Zentralanstalt für Meteorologie und Geodynamik (ZAMG)

Die Zentralanstalt für Meteorologie und Geodynamik ist der 1851 gegründete staatliche meteorologische und geophysikalische Dienst Österreichs. Der Dienst bietet eine große Produktpalette, von der klassischen Wettersvorhersage, über Klimainformation, Geophysikalische Dienste und Umweltberatung. Das Interesse für den „Weather Twitter“ beschränkt sich dabei hauptsächlich auf aktuelle Lufttemperaturen und die Wettersituation, sowie eventuell Wetterprognosen. Die ZAMG betreibt österreichweit ein umfassendes Stationsnetz. Dieses Netz umfasst folgende Stationen:

- *246 teilautomatische Wetterstationen (TAWES)*: Sie liefern die wichtigsten Wetterparameter (Lufttemperatur, Luftfeuchte, Luftdruck, Niederschlag, Sonnenscheindauer, Globalstrahlung etc.). An 66 Standorten werden auch Zusatzbeobachtungen für den internationalen Vereinigung der Wetterdienste World Meteorological Organization (WMO) aufgezeichnet.
- *6 Minitaklis-Messstellen*: Diese Stationen sind das Vorgängermodell der jetzt im Einsatz befindlichen Stationen.
- *90 Stationen mit phänologischen Beobachtungen*: Hier werden seit 1951 kontinuierlich Pflanzen- und tierphänologische Beobachtung erfasst.
- *Mobile Messstationen*: Diese werden meist im Rahmen von Projekten in Österreich betrieben.

Außerdem werden die Messdaten verschiedener Fremdstationen zu Vergleichszwecken herangezogen:

- *Austro Control Group (ACG)-Stationen*: Stationen der Austro Control, die den geregelten Ablauf des Flugverkehrs im österreichischen Luftraum regelt, und diese Stationen betreibt, um eine punktgenaue Analyse und Prognose der Wetterverhältnisse für den Luftverkehr an den Flughäfen zu erstellen.



- Ö3-Stationen: Seit 2006 von Gemeinden betriebene Stationen, die vom Radiosender Ö3 und der ZAMG vermarktet werden. Gewartet werden die derzeit 8 Stationen in Österreich von der ZAMG.

Aus diesem Stationsnetz bezieht die ZAMG seine Wetterdaten und erstellt daraus Prognosen und Analysen. Die ZAMG bietet verschiedene meteorologische Parameter wie z.B. Temperatur, Wind, Niederschlag in digitaler Form an. Dabei werden die Daten in unterschiedlicher Weise angeboten:

- Aktuelle Messwerte in 10-minütiger Messfrequenz.
- Analyse- und Vorhersagewerte als digitale Werte (15-minütig bis 6-stündig).
- Graphisch aufbereitete Daten.

Der Zugriff auf die Daten erfolgt entweder direkt über einen eigens eingerichteten Server oder die Daten werden in einem gewünschten Format z.B. American Standard Code for Information Interchange (ASCII) bzw. Comma-Separated Values (CSV) zugeschickt. Die Bezahlung der Daten erfolgt je nach Umfang und Art der Daten (ZAMG (Zuletzt besucht: 03.Juli 2011)).

### Österreichischer Rundfunk (ORF)

Der Wetterdienst des ORF bietet dem Besucher detaillierte Wetterinformation für Österreich. Jedoch kann auch Information zu Österreichs Nachbarländern, Europa und der ganzen Welt abgerufen werden. Der ORF bezieht seine Daten von der ZAMG und stellt neben den allgemeinen Informationen wie Temperatur, Luftdruck, Wind und Wetterlage auch Wassertemperaturen, Ozonwerte, Luftwerte und Polleninformation zur Verfügung. Desweiteren können auch Spezialkarten (Satellitenbilder, Bergwetter, Blitzaktivität) abgerufen werden. Neben den aktuellen Messwerten werden auch Wettersvorhersagen bis zu einer Woche geliefert. Für die Steiermark wird das Wetter in 42 Orten aufbereitet, wobei die Daten stündlich aktualisiert werden.

#### wetter.at

Der Dienst [wetter.at](http://wetter.at) ist ein privater Anbieter von Wetterinformation, der zur Mediengruppe Österreich gehört. Auch dieser Dienst bietet weltweite Wetterinformation an. Ein beliebtes Service ist die Ortseingabe und die damit verbundene Wetterabfrage. Das Messnetz von [wetter.at](http://wetter.at) umfasst fixe Wetterstationen verteilt in ganz Österreich. Neben dem bestehenden Messnetz bietet [wetter.at](http://wetter.at) eine sogenannte Messnetzpartnerschaft an. Hierbei kann eine Wetterstation erworben werden, die gleichzeitig Daten in das [wetter.at](http://wetter.at) Messnetz einbindet. Dieses Angebot wird hauptsächlich von Firmen und Tourismusbetrieben genutzt.

### 3.1.2 Wetterdienste international

International wurden im Gegensatz zu Österreich einige Anbieter gefunden, die Wetterinformation in diversen Formaten (XML-Format, CSV-Format oder als Really Simple Syndication (RSS)-

Feed) kostenlos bereit stellen. Davon wurden drei Dienste untersucht, wobei vor allem die regionale Verfügbarkeit der Wetterinformation von Bedeutung ist.

### World Weather Online

World Weather Online ist ein Dienst, der globale Wetterdaten und Wettervorschau Daten in verschiedenen Formaten zur Verfügung stellt. Der Dienst verfügt über Mitarbeiter in den USA, Großbritannien und Indien, sowie über Wetterzentren in Dänemark und Deutschland. Die Wetterdatenbank deckt über zwei Millionen Orte weltweit ab. Die Weather API von World Weather Online, die nach erfolgter Anmeldung und mit einem vom Dienst zur Verfügung gestellten Benützungscodes genutzt werden kann, ermöglicht Web-Entwicklern den Zugriff auf Wetterdaten für die jeweils nächsten fünf Tage ab dem Zugriff im XML-, CSV- oder JavaScript Object Notation-Format (JSON)-Format. Dabei kann der Entwickler Daten für einen beliebigen Standort in Form von Koordinaten oder Postleitzahl bzw. Zipcode in den USA abrufen. Nachdem sich das XML-Format sehr gut zur Weiterverarbeitung eignet wurden die Abfragen an den Dienst World Weather Online in dieser Arbeit im XML-Ausgabeformat angefordert. Eine entsprechende Anfrage an den Dienst kann über einen HTTP-Request gesendet werden. Dabei kann der Benutzer folgende vier Parameter übergeben:

- *Abfrage (q)*: Hier können entweder WGS84-Koordinaten (lat,lon), ein Ortsname, oder ein Postcode übergeben werden.
- *Format (format)*: Dieser Parameter dient zur Auswahl des Ausgabeformates, wobei der Dienst zwischen XML, CSV und JSON wählen lässt.
- *Anzahl der Tage (num\_of\_days)*: Anzahl der Tage, für die Wetterinformation benötigt wird (bis zu 5 Tage).
- *Benutzerschlüssel (key)*: Eindeutige Kennung des Benutzers, der nach Anforderung durch Anmeldung übermittelt wird.

Die Abfrage für Graz ([http://www.worldweatheronline.com/feed/weather.ashx?q=Graz&format=xml&num\\_of\\_days=1&key=949e2597c8081703101211](http://www.worldweatheronline.com/feed/weather.ashx?q=Graz&format=xml&num_of_days=1&key=949e2597c8081703101211)) liefert als Ergebnis das in Listing 3.1 exemplarisch dargestellte XML-Dokument. Über die entsprechenden XML-Tags kann auf die benötigte Information zugegriffen werden. World Weather Online unterteilt seine Antwort in drei Teile:

- *request*: Der Request enthält die Abfrageparameter (type und query).
- *current\_condition*: Enthält Details zur aktuellen Wettersituation (u.a. Temperatur, Wettericon, Wind, Druck, Feuchtigkeit).
- *weather*: Beinhaltet für jeden Tag die entsprechende Wetterprognoseinformation (u.a. Datum, Temperatur, Wettericon, Wind).

**Listing 3.1** – XML-Abfrage beim Dienst World Weather Online

```

<data>
- <request>
  <type>City</type>
  <query>Graz, Austria</query>
</request>
- <current_condition>
  <observation_time>07:24 AM</observation_time>
  <temp_C>15</temp_C>
  <temp_F>59</temp_F>
  <weatherCode>116</weatherCode>
- <weatherIconUrl>
  http://www.worldweatheronline.com/.../wsymbol_0002_sunny_intervals.png
</weatherIconUrl>
<weatherDesc>Partly Cloudy </weatherDesc>
<windspeedMiles>0</windspeedMiles>
<windspeedKmph>0</windspeedKmph>
<winddirDegree>38</winddirDegree>
<winddir16Point>NE</winddir16Point>
<precipMM>0.0</precipMM>
<humidity>77</humidity>
<visibility>10</visibility>
<pressure>1022</pressure>
<cloudcover>50</cloudcover>
</current_condition>
- <weather>
  <date>2011-05-18</date>
  <tempMaxC>21</tempMaxC>
  <tempMaxF>69</tempMaxF>
  <tempMinC>5</tempMinC>
  <tempMinF>40</tempMinF>
  <windspeedMiles>6</windspeedMiles>
  <windspeedKmph>10</windspeedKmph>
  <winddirection>SE</winddirection>
  <winddir16Point>SE</winddir16Point>
  <winddirDegree>145</winddirDegree>
  <weatherCode>113</weatherCode>
- <weatherIconUrl>
  http://www.worldweatheronline.com/.../wsymbol_0001_sunny.png
</weatherIconUrl>
<weatherDesc>Sunny</weatherDesc>
<precipMM>0.0</precipMM>
</weather>
</data>

```

Aus den vom Wetterdienst gesendeten Informationen lassen sich die benötigten Teile mittels XML-Parser auslesen und in der benötigten Form weiterverarbeiten.

Auf den ersten Blick scheint mit World Weather Online bereits ein guter Dienst gefunden zu sein, jedoch konnte der Dienst, wie in Abschnitt 3.2 zu sehen sein wird, die Anforderungen des Weather Twitter nicht zur Gänze erfüllen.

## Google

Einen weiteren Dienst zur Abfrage von Wetterinformation stellt Google bereit. Dafür stellt der Anbieter eine API zur Verfügung, von der Wetterdaten im XML-Format abgerufen werden können, wobei das Format bei Google nicht gewählt werden kann. Die Abfrage wird mittels zwei Parametern gesteuert:

- *Adressangabe (weather)*: Eingabe eines Ortes und des Landes.
- *Sprachcode (hl)*: Angabe eines Sprachcodes (en,de).

Google bedient sich dabei dem Dienst Wunderground<sup>4</sup>.

Die Abfrage für Graz (<http://www.google.com/ig/api?weather=Graz-Austria&hl=en>) liefert bei Google ein XML-Dokument, das ähnlich zu Worldweatheronline gegliedert ist (siehe Listing 3.2). Das Dokument umfasst drei Teile:

- *forecast\_information*: Dieser Tag enthält allgemeine Information zur Abfrage (Ort, Datum, Zeit).
- *current\_conditions*: Enthält ähnlich zu WWO Details zur aktuellen Wettersituation (u.a. Temperatur, Wettericon, Wind, Feuchtigkeit).
- *forecast\_conditions*: Beinhaltet für jeden Tag die entsprechende Wetterprognoseinformation (u.a. Wochentag, maximale und minimale Temperatur, Wettericon).

### Listing 3.2 – XML-Wetterabfrage beim Google-Wetterdienst

```
<xml_api_reply version="1">
- <weather module_id="0" tab_id="0" mobile_row="0" mobile_zipped="1" row="0" section="0">
  - <forecast_information>
    <city_data data="Graz, Styria"/>
    <postal_code data="Graz-Austria"/>
    <latitude_e6 data=""/>
    <longitude_e6 data=""/>
    <forecast_date data="2011-05-18"/>
    <current_date_time data="2011-05-18 08:00:00 +0000"/>
    <unit_system data="SI"/>
  </forecast_information>
  - <current_conditions>
    <condition data="Klar"/>
    <temp_f data="64"/>
    <temp_c data="18"/>
    <humidity data="Feuchtigkeit: 68 %"/>
    <icon data="/ig/images/weather/sunny.gif"/>
    <wind_condition data="Wind: N mit 2 km/h"/>
  </current_conditions>
  - <forecast_conditions>
    <day_of_week data="Mi."/>
    <low data="7"/>
    <high data="23"/>
  </forecast_conditions>
</weather>
</xml_api_reply>
```

<sup>4</sup><http://www.wunderground.com>

```
<icon data="/ig/images/weather/sunny.gif"/>
<condition data="Klar"/>
</forecast_conditions>
</weather>
</xml_api_reply>
```

## Wunderground

Nachdem Google seine Daten von der Wetterplattform Wunderground bezieht, wurde auch dieser Dienst näher betrachtet. Der Begriff Wunderground setzt sich aus den Begriffen „Weather“ und „Underground“ zusammen. Wunderground ist ein kommerzieller Wetterdienst aus den USA, der weltweite Wetterinformation anbietet. Der Dienst wurde 1995 an der Universität von Michigan gestartet.

Wunderground bezieht den Hauptanteil seiner Daten aus den USA. Dabei werden ca. 18000 Stationen genutzt, die sich in unterschiedlichen Stationsnetzen wie folgt aufgliedern:

- *Automated Surface Observation Stations (ASOS)*: Dieses Netzwerk besteht aus 2000 Stationen, die sich auf Flugplätzen verteilt in den ganzen USA befinden, und von der Nationalen Luftfahrtbehörde gewartet werden.
- *Private Weather Station (PWS)*: Wunderground betreibt und wartet in den USA ein über 8500 Stationen großes privates Stationsnetzwerk. Die dabei aquirierten Daten unterliegen laut Wunderground strengen Qualitätskontrollen und werden alle 2-3 Sekunden aktualisiert.
- *Meteorological Assimilation Data Ingest System (MADIS)*: Die National Oceanic and Atmospheric Administration (NOAA), eine US-Behörde, die vor allem die Vorgänge in der Atmosphäre und den Ozeanen beobachtet und die Bevölkerung vor auftretenden Wetterveränderungen (Stürme, Hurricans) warnt und Klimaveränderungen aufzeichnet, betreibt weitere 7500 Station in den USA, die Teil des MADIS-Netzwerkes sind.

Darüberhinaus bezieht Wunderground weltweit Wetterdaten aus über 8000 Wetterstationen. Dabei handelt es sich hauptsächlich um Flughafenstationen, jedoch auch Information von bestimmten Regierungsstellen. Diese Information wird je nach Anbieter alle ein bis sechs Stunden aktualisiert. Zusätzlich kann Wunderground, wie in den USA auch auf sein Private Weather Station-Netzwerk zurückgreifen. In Österreich umfasst das private Netzwerk derzeit 60 Stationen<sup>5</sup>, darunter sechs PWSs in der Steiermark. Zusammen mit den drei Flugplätzen (Aigen im Ennstal, Graz, Zeltweg) können in der Steiermark 9 Wetterstationen genutzt werden (Wunderground (Zuletzt besucht: 07.Juli 2011)).

Weather Underground bietet auch eine Schnittstelle an, von der Wetterinformation im XML-Format bezogen werden kann. Hierbei kann der Benutzer im Unterschied zu den zuvor vorgestellten Diensten nur mehr einen Parameter übergeben:

<sup>5</sup><http://www.wunderground.com/weatherstaion/ListStations.asp?selectedCountry=Austria>

- *Abfrageort (query)*: Auswahl des Ortes und des Landes, durch einen Beistrich getrennt.

Die Abfrage für Graz (<http://api.wunderground.com/auto/wui/geo/WXCurrentObXML/index.xml?query=Graz,Austria>) liefert ein Ergebnis, das äquivalent zu den Diensten World Weather Online und Google bestimmte Wetterinformationen beinhaltet, die je nach Notwendigkeit ausgelesen werden können.

### Norwegisches Meteorologisches Institut (yr.no)

yr.no ist die Webseite des Norwegischen Meteorologischen Institutes. Die Seite stellt detaillierte Wettervorschau Daten für Norwegen, aber auch für einige andere Länder, vor allem in Europa, bereit. Die verfügbaren Wettervorhersagedaten für steirische Orte werden im „Weather Twitter“ aus dem entsprechenden XML-File ausgelesen und verwendet. Die Abfrage für Graz (<http://www.yr.no/place/Austria/Styria/Graz/forecast.xml>) liefert die Wettervorhersage für die nächsten neun Tage. In Listing 3.3 ist beispielhaft die Wettervorhersage für einen Tag dargestellt.

**Listing 3.3** – XML-Wettervorhersage vom norwegischen Dienst yr.no

```
<weatherdata>
- <location>
  <name>Graz</name>
  <type>Regional capital</type>
  <country>Austria</country>
  <timezone id="Europe/Vienna" utcOffsetMinutes="120" />
  <location altitude="365" latitude="47.06667" longitude="15.45"
    geobase="geonames" geobaseid="2778067" />
</location>
- <credit>
  - <link text="Weather forecast from yr.no,
    delivered by the Norwegian Meteorological Institute and the NRK"
    url="http://www.yr.no/place/Austria/Styria/Graz/" />
</credit>
- <links>
  <link id="xmlSource" url="http://www.yr.no/place/.../Graz/forecast.xml" />
  <link id="overview" url="http://www.yr.no/place/Austria/Styria/Graz/" />
  <link id="hourByHour" url="http://www.yr.no/place/.../Graz/hour_by_hour.html" />
  <link id="longTermForecast" url="http://www.yr.no/place/.../Graz/long.html" />
</links>
- <meta>
  <lastupdate>2011-07-29T09:02:57</lastupdate>
  <nextupdate>2011-07-29T22:00:00</nextupdate>
</meta>
<sun rise="2011-07-29T05:33:02" set="2011-07-29T20:35:34" />
- <forecast>
  <tabular>
    <time from="2011-07-29T23:00:00" to="2011-07-30T05:00:00" period="0">
    <!-- Valid from 2011-07-29T23:00:00 to 2011-07-30T05:00:00 -->
    <symbol number="3" name="Partly cloudy" var="mf/03n.06" />
    <precipitation value="0.0" /><!-- Valid at 2011-07-29T23:00:00 -->
    <windDirection deg="324.3" code="NW" name="Northwest" />
    <windSpeed mps="2.8" name="Light breeze" />
    <temperature unit="celsius" value="16" />
    <pressure unit="hPa" value="1013.4" />
```

```

</time>
<time from="2011-07-30T05:00:00" to="2011-07-30T11:00:00" period="1">
<!-- Valid from 2011-07-30T05:00:00 to 2011-07-30T11:00:00 -->
  <symbol number="9" name="Rain" var="09"/>
  <precipitation value="0.4"/><!-- Valid at 2011-07-30T05:00:00 -->
  <windDirection deg="318.2" code="NW" name="Northwest"/>
  <windSpeed mps="2.5" name="Light breeze"/>
  <temperature unit="celsius" value="14"/>
  <pressure unit="hPa" value="1012.3"/>
</time>
<time from="2011-07-30T11:00:00" to="2011-07-30T17:00:00" period="2">
<!-- Valid from 2011-07-30T11:00:00 to 2011-07-30T17:00:00 -->
  <symbol number="9" name="Rain" var="09"/>
  <precipitation value="2.0"/><!-- Valid at 2011-07-30T11:00:00 -->
  <windDirection deg="247.6" code="WSW" name="West-southwest"/>
  <windSpeed mps="1.5" name="Light air"/>
  <temperature unit="celsius" value="19"/>
  <pressure unit="hPa" value="1012.1"/>
</time>
</tabular>
</forecast>
</weatherdata>

```

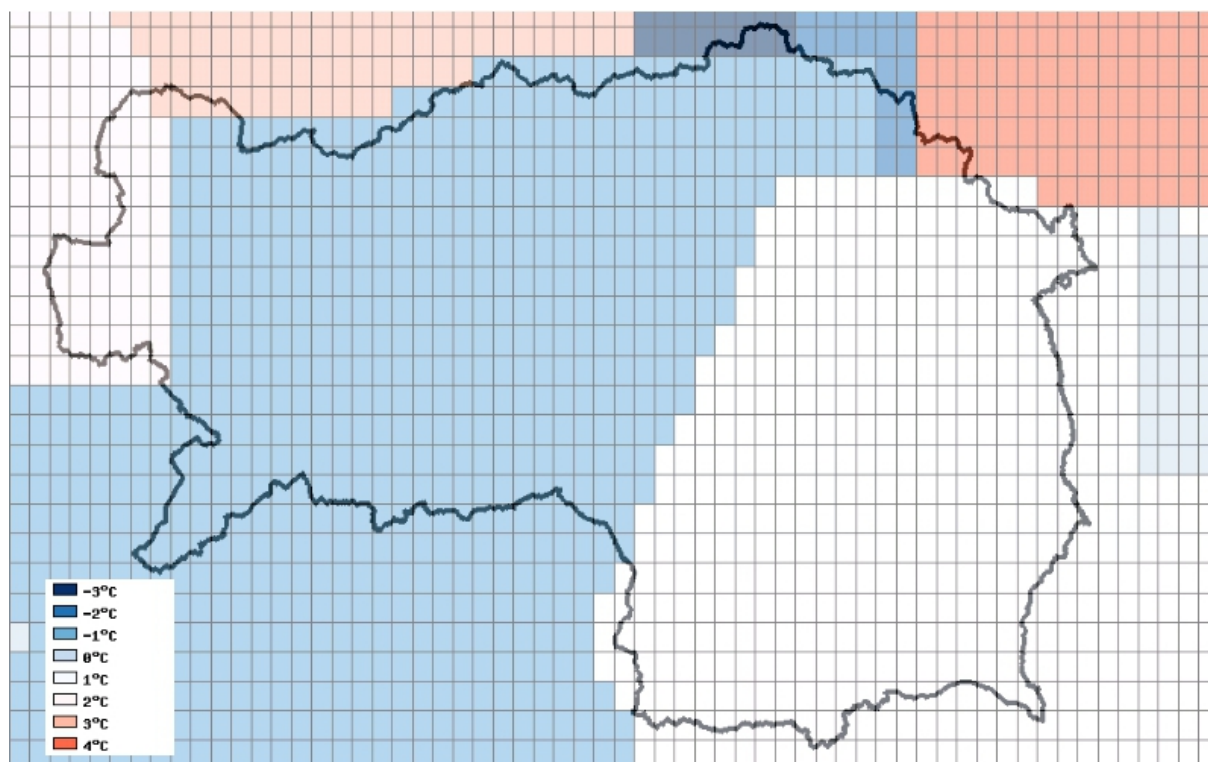
## 3.2 Analyse und Aufbereitung der verschiedenen Wetterdaten für die Steiermark

Um die von den Wetterdiensten erhaltenen Daten auf Verwendbarkeit überprüfen zu können, wurden die Daten untersucht. Zuerst wurden die Daten von World Weather Online als Raster über die gesamte Steiermark dargestellt (Abbildung 3.1).

Dabei ergab sich das Bild, dass in der Steiermark nur drei bis vier verschiedene Temperaturen auftraten. Anschließend wurden die Wetterdaten für die Bezirkshauptstädte der Steiermark von mehreren Anbietern untersucht. Nachdem der Dienst Wunderground neben den meteorologischen Informationen auch den Stationsnamen, von dem die Wetterdaten stammen, mitliefert, konnte man darauf rückschließen, dass der Dienst die Informationen für die Steiermark von vier Messstationen bezieht. Dabei handelt es sich um die Flughäfen Graz Thalerhof, Zeltweg, Aigen in der Steiermark, sowie den Flughafen Zell am See in Salzburg. Die Daten für die anderen Orte wurden mittels Nearest Neighbor Verfahren (siehe Abschnitt 2.2.4) ermittelt. Auch die anderen Dienste bedienen sich dieser Stationen, wobei die Ergebnisse aufgrund unterschiedlicher Interpolationsverfahren leicht voneinander abweichen, was in Abschnitt 3.2.1 zu sehen ist. Dieses Ergebnis ist aufgrund der Anforderung, eine möglichst regionale Wetterinformation zur Verfügung zu stellen, nicht zufriedenstellend, und es musste Ausschau nach einer Alternative gehalten werden.

### 3.2.1 Vergleich der Temperaturdaten der verschiedenen Anbieter

Um eine Auswahl der Wetterdaten treffen zu können, wurden die Temperaturen der verschiedenen Anbieter für die Bezirkshauptstädte der Steiermark über einen Zeitraum von 18 Stunden von



**Abbildung 3.1** – Rasterdarstellung der Temperaturen für die Steiermark von World Weather Online

20 Uhr bis 14 Uhr aufgezeichnet und verglichen. In Abbildung 3.2 werden die Temperaturen der vier Anbieter, World Weather Online (Rot), Wunderground (Grün), Google (Blau) und ORF (Orange) über einen Zeitraum von 18 Stunden dargestellt. Dabei fällt auf, dass sich die Temperaturwerte für die Städte Graz, Gleisdorf, Feldbach und Fürstenfeld bei allen Anbietern außer dem ORF nicht unterscheiden. Die maximalen Abweichungen zwischen den vier Anbietern betragen 4° Celsius, wobei sich diese geringen Differenzen durch die Wetterstation am Flughafen Graz-Thalerhof erklären lassen, deren Wetterdaten von allen Anbietern verarbeitet werden.

In anderen Orten (siehe Abbildung 3.3) unterscheiden sich die von den Diensten abgerufenen Temperaturen zum jeweiligen Zeitpunkt um bis zu 8° Celsius, was sich auf verschiedene Verteilungen der Messstationen und unterschiedliche Interpolationsverfahren der Anbieter zurückführen lässt.

Außerdem konnte beobachtet werden, dass die Temperaturdaten einiger Orte nicht von allen Diensten angeboten werden. Sehr interessant bei der XML-Abfrage der Temperaturdaten vom Dienst Wunderground ist, dass dieser die Wetterstation, von der die Daten stammen in der Antwort mitliefert. Dabei treten zur Abdeckung der Steiermark nur vier verschiedene Stationen auf (Graz, Zeltweg, Aigen im Ennstal und Zell am See), wobei z.B. die Temperaturdaten für Liezen von Aigen im Ennstal aber auch von Zeltweg stammen, wenn aus Aigen keine Werte bezogen werden können. Diese Beobachtung deckt sich nicht mit der Angabe der steirischen Messstationen in Abschnitt 3.1.2. Auffällig sind auch die unterschiedlichen Verläufe der Temperaturen von Google und Wunderground, obwohl Google seine Wetterdaten von Wunderground bezieht, diese



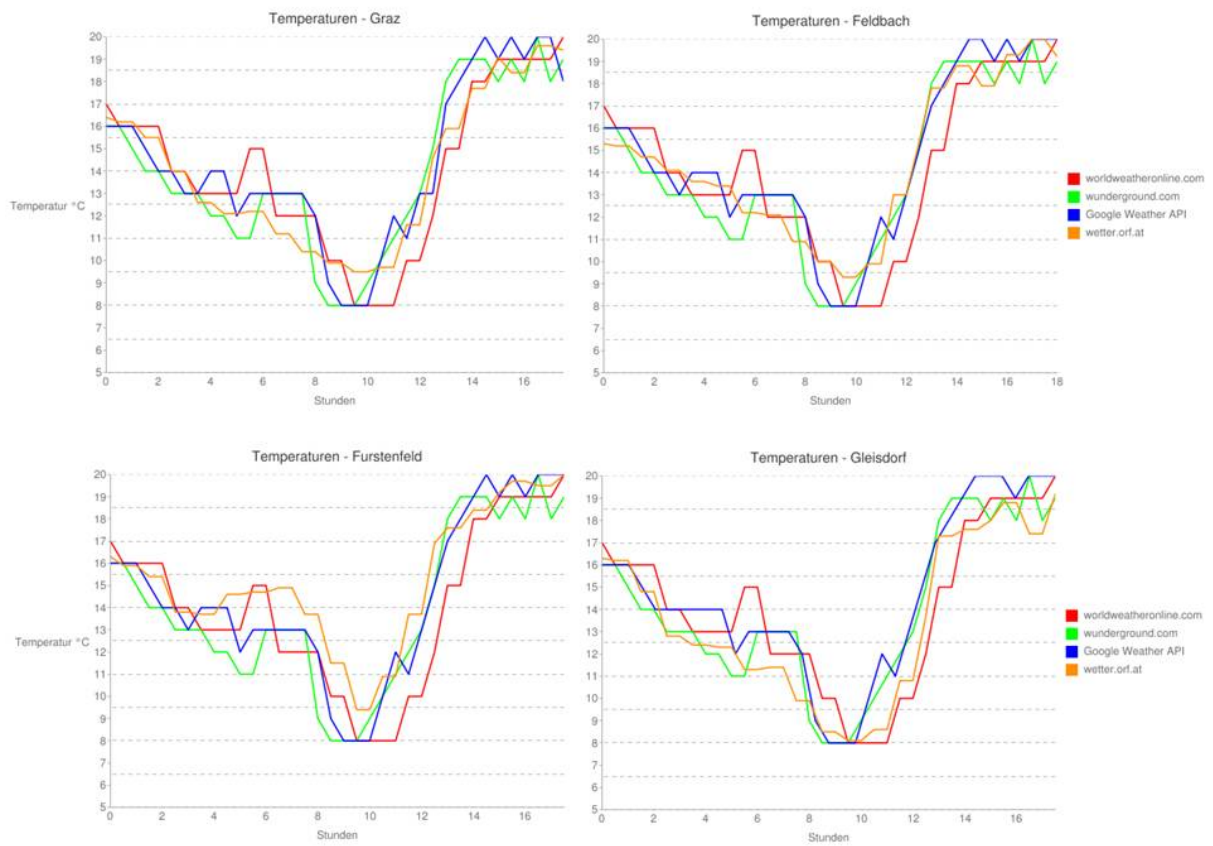


Abbildung 3.2 – Vergleich der Temperaturen von Graz und nahen Städten

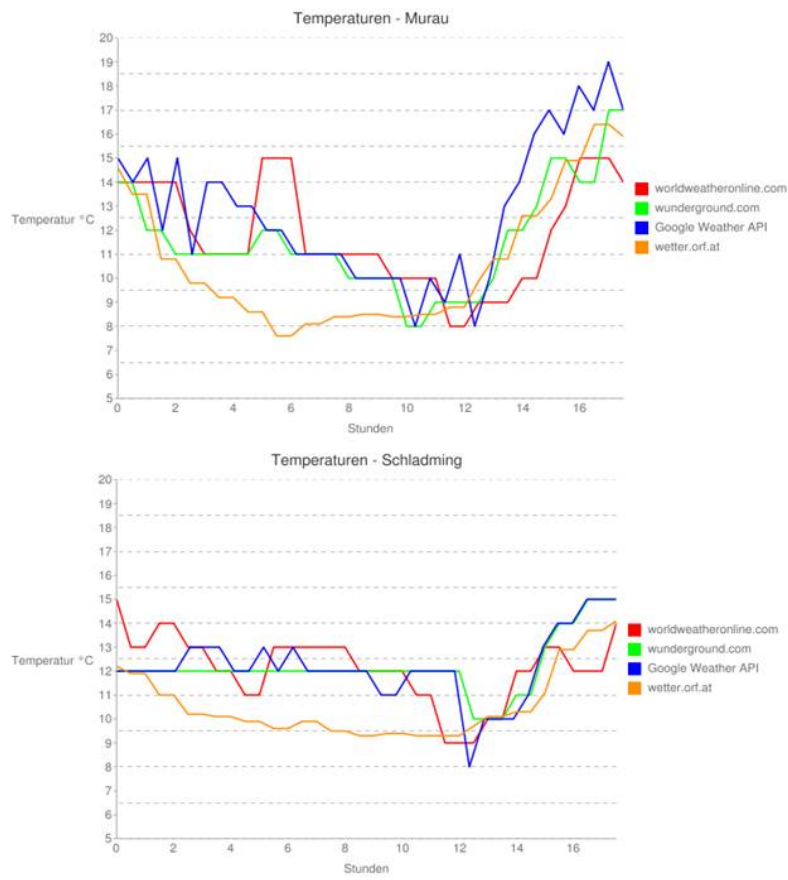


Abbildung 3.3 – Temperaturzeitreihen für Murau und Schladming

offensichtlich aber in anderer Form weiterverarbeitet und anbietet.

### 3.2.2 Wahl der Grundlagendaten

Aufgrund dieser Unsicherheitsfaktoren und der Tatsache, dass es sich beim „Weather Twitter“ nicht um eine kommerzielle Applikation handelt, wurden als Grundlagendaten für die Steiermark die Daten des ORF gewählt. Dabei werden die Temperaturdaten mittels PHP-Funktion aus dem HTML-Sourcecode der ORF-Webseite ausgelesen, wo sie anschließend in einer PostgreSQL-Datenbank mit PostGIS-Erweiterung abgespeichert werden. Problem dabei ist die rechtliche Grundlage, sowie die unzuverlässige Verfügbarkeit der Daten, falls der HTML-Sourcecode der Webseite `wetter.orf.at` in irgendeiner Form verändert werden sollte. Die Grundlage für die aktuelle Wetterlage in Form eines Wettericons stammt vom Dienst World Weather Online, wobei diese für die Orte Graz und Zeltweg abgerufen wird. Somit werden die Grundlagendaten für den „Weather Twitter“ von den ORF-Wetterseiten bzw. von World Weather Online bezogen. Je nach Anzahl der Nutzer haben die gelieferten Daten mehr oder weniger Gewicht, da die von den Nutzern eingetragenen Wettermeldungen natürlich auch in die Datenaufbereitung einfließen.

### 3.2.3 Wetterdaten aus den Benutzerwettermeldungen

Die Hauptdatenquelle des „Weather Twitter“ sind jedoch die Anwender der Applikation. Diese haben die Möglichkeit die aktuelle Lufttemperatur und Wetterlage an das System zu übermitteln und somit sowohl die räumliche, wie auch die zeitliche Auflösung des Datenbestandes entscheidend zu beeinflussen.

### 3.2.4 Aufbereitung der Daten

Von der Wetterseite des ORF können insgesamt Wetterdaten von 42 steirischen Orten abgerufen werden. Ziel ist jedoch eine flächendeckende Wetterinformation für die gesamte Steiermark zu realisieren. Das passiert in Form eines Rasters mit einer Rasterzellengröße von  $1000 \times 1000$  Meter. Diese Zellgröße ergab sich, da in dieser Auflösung ein DGM für die Steiermark vorliegt, und mit dieser Größe lokale Wetterphänomene noch gut aufgelöst werden können. Jeder dieser Rasterzellen werden mithilfe einer räumlichen Interpolation bestimmte Wetterdaten (Lufttemperatur, Wetterlage) zugeordnet.

#### Räumliche Interpolation der Temperaturdaten

Die räumliche Interpolation der Temperaturdaten ist notwendig, um Werte zu schätzen, wo keine Temperaturmessungen durchgeführt wurden. In dieser Arbeit wurde die Interpolation der Temperaturen mittels eines IDW-Verfahrens durchgeführt. Die Interpolation wurde anhand der Formel 3.1 umgesetzt, dabei entspricht  $t(x)$  dem interpolierten Temperaturwert  $t$  für einen Punkt  $x$ .

Inverse Distance Weighting (IDW)-Funktion

$$t(x) = \sum_{i=0}^N \frac{w_i(x)t_i}{\sum_{j=0}^N w_j(x)} \quad (3.1)$$

mit Gewichtsfunktion

$$w_i(x) = \frac{1}{d(x, x_i)^p} \quad (3.2)$$

In die Gewichtsfunktion (3.2) fließt die, mit dem Gewichtungsfaktor  $p$ , gewichtete Distanz  $d$  (in Metern), zwischen dem Rasterpunkt  $x$ , und dem nächsten Messwert  $x_i$  ein. Die Gewichtung der Temperaturdaten im Weather Twitter erfolgt je nach Alter der Wettermeldung in 10-Minuten-Schritten ( $t=1$  von 0 bis 10 Minuten alt ...  $t=6$  von 50 bis 60 Minuten alt) mit dem Faktor  $p=1/t$ , um nahe liegende und kürzlich eingetragene Temperaturwerte stärker zu berücksichtigen. Alle Wettereintragungen, die älter als eine Stunde sind, werden bei der Interpolation nicht mehr miteinbezogen.

Das Inverse Distance Weighting ist ein deterministisches Verfahren, womit keine Aussage zur statistischen Signifikanz des Ergebnisses möglich ist. Vorteil des Verfahrens ist die einfache Implementierung, wodurch der Rechenaufwand und damit die Rechenzeit gering ist. Um die Daten vergleichbarer zu machen, wurden die Temperaturen vor der Interpolation mithilfe eines Temperaturgradienten auf ein gemeinsames Niveau gebracht. Im Mittel über alle Wetterlagen nimmt die Temperatur in der Troposphäre pro 100m Höhenanstieg um  $0,65^\circ\text{C}$  ab (Klimedia (Zuletzt besucht: 07.Juli 2011)). Dieser Temperaturgradient wurde mit der Näherungsformel

$$t_0 = t_h + \frac{h}{100} * 0.65^\circ\text{C} \quad (3.3)$$

berücksichtigt. Eine lineare Temperaturabnahme ergibt im Mittel eine gute Näherung, wobei dabei auch Abweichungen z.B. durch Inversionswetterlagen entstehen können. Diese besonderen Wetterphänomene wurden bei der Normalisierung allerdings nicht berücksichtigt. Nach der Interpolation wird das Wetterraster wieder auf sein ursprüngliches Höhenniveau zurückgerechnet.

### Interpolation der Wetterlage

Die Interpolation der Wetterlage erfolgt sowohl über die von World Weather Online bezogenen Wetterlagen für Graz und Zeltweg, als auch aus den Wettermeldungen der letzten Stunde. Dabei wird der Rasterzelle mittels Nearest Neighbor Verfahren die entsprechende Wetterlage zugewiesen.

### Das Wetterraster

Das infolge der Interpolationen der letzten beiden Abschnitte entstandene Wetterraster stellt neben den einzelnen Wettermeldungen die Grundlage des „Weather Twitter“ dar. Es dient zur

optischen Darstellung in der OpenLayers-Anwendung bzw. im KML-File. Außerdem können aus dem Raster für jeden Punkt der Steiermark Wetterdaten abgefragt werden. Diese Daten werden z.B. zur Überprüfung der von den Anwendern eingegebenen Wetterdaten benutzt. So darf die Temperatureingabe des Users maximal um  $\pm 5^{\circ}\text{C}$  von der Rastertemperatur an der entsprechenden Position abweichen. Falls der Benutzer keine Temperatur eingeben kann, ist es möglich die Rastertemperatur zu bestätigen.

### 3.3 Datenverwaltung für den „Weather Twitter“

Die ausgewählten und aufbereiteten Daten müssen zur weiteren Verwendung entsprechend verwaltet werden. Die Speicherung der Daten erfolgt im Datenbanksystem PostgreSQL. Die räumlichen Analysen werden mit diversen Operatoren der räumlichen Spracherweiterung PostGIS durchgeführt, die jeweils über PHP-Skripts implementiert wurden.

#### 3.3.1 PostgreSQL

PostgreSQL ist ein freies Objektrelationales Datenbank Managementsystem (ORDBMS). Seine ursprüngliche Entwicklung begann im Jahre 1986 als Universitätsprojekt POSTGRES an der University of California in Berkeley. Zu Beginn der 1990er Jahre wurde das System immer bekannter und von verschiedenen Institutionen verwendet. Nachdem dem System POSTGRES im Jahr 1994 ein SQL Interpreter hinzugefügt wurde, und Datenbanken im Zuge der Entwicklung des World Wide Web eine immer größere Bedeutung zukam, wurde es unter dem Namen Postgres95 im Internet verbreitet. Ein Jahr später 1996 erhielt es seinen heutigen Namen PostgreSQL, unter welchem die Entwicklung des Datenbankmanagementsystems bis heute ungebrochen weiterging. Heute ist PostgreSQL das meist entwickelte Open Source Datenbanksystem, das zur Verfügung steht. Das Datenbanksystem beinhaltet einen großen Teil der SQL-Standards und bietet viele wichtige Funktionen. Neben komplexen Abfragen, Unterabfragen und Mengenoperationen, sind die Referenzielle Integrität, Trigger sowie Schnittstellen zu vielen Programmiersprachen wie z.B. PHP erwähnenswert. Außerdem lässt sich PostgreSQL durch eine Reihe von verschiedenen Zusatzmodulen sehr leicht erweitern. Die in diesem Zusammenhang wichtigste Erweiterung ist PostGIS, welche es erlaubt geografische Objekte und Datenstrukturen zu verwalten und somit als Datenbank für GIS sehr gut geeignet ist.

#### 3.3.2 PostGIS

PostGIS ist eine räumliche Erweiterung für die objektrelationale Datenbank PostgreSQL. In Kombination mit PostgreSQL bildet PostGIS eine Geodatenbank die bevorzugt in Geoinformationssysteme eingebunden wird. Das Projekt wurde nach den Richtlinien des Open Geospatial Consortium (OGC) entwickelt deren Simple Feature Access (SFA)-Spezifikation (siehe Abschnitt 2.4.2) umgesetzt wurde. Die Einbindung von PostGIS-Daten wird sowohl von der Javascript Bibliothek OpenLayers als auch vom University of Minnesota (UMN)-MapServer unterstützt.

PostGIS wird seit 2000 von der Refractions Research Inc, im Rahmen eines Forschungsprojektes als räumliches Datenbanksystem entwickelt.

### Geometrietypen

Die GIS-Objekte, die von PostGIS unterstützt werden sind eine Untermenge der „Simple Features for SQL“-Spezifikation, die vom OGC definiert wurden. Seit Version 0.9 unterstützt PostGIS alle diese Objekte. Die OGC-Spezifikation beinhaltet zwei Grundtypen zur Definition von Geometrieobjekten: das Well Known Text (WKT), sowie das Well Known Binary (WKB).

Im WKT-Format werden vom OGC folgenden Geometrietypen unterschieden und von PostGIS unterstützt:

- POINT(0 0).
- LINESTRING(0 0,1 1,1 2).
- POLYGON((0 0,4 0,4 0,0 0),(1 1, 2 1, 2 2, 1 2,1 1)).
- MULTIPOINT(0 0,1 2).
- MULTILINESTRING((0 0,1 1,1 2),(2 3,3 2,5 4)).
- MULTIPOLYGON(((0 0,4 0,4 0,0 0),(1 1,2 1,2 2,1 2,1 1)), ((-1 -1,-1 -2,-2 -2,-2 -1,-1 -1))).
- GEOMETRYCOLLECTION(POINT(2 3),LINESTRING(2 3,3 4)).

Die OGC-Spezifikation verlangt zusätzlich zum Geometrietyp auch die Angabe eines sogenannten Spatial Reference System Identifier (SRID). Diese Identifikationsnummer wird benötigt, um der Geometrie ein Bezugssystem zuzuweisen, und sie ordnungsgemäß in der Datenbank verspeichern zu können. Da in der OGC-Spezifikation nur 2D-Geometrien unterstützt werden wurde in PostGIS ein erweitertes Format eingeführt, das extended Well Known Text (EWKT)-Format bzw. auch das extended Well Known Binary (EWKB) Format. Beispiele für Geometrieobjekte im EWKT Format sind:

- POINT(0 0 0).
- SRID=32632;POINT(0 0) – XY with SRID.
- POINTM(0 0 0) – XYM.
- POINT(0 0 0 0) – XYZM.
- SRID=4326;MULTIPOINTM(0 0 0, 1 2 1) – XYM with SRID.
- MULTILINESTRING((0 0 0, 1 1 0, 1 2 1), (2 3 1, 3 2 1, 5 4 1)).
- POLYGON((0 0 0, 4 0 0, 4 4 0, 0 4 0, 0 0 0),(1 1 0, 2 1 0, 2 2 0, 1 2 0, 1 1 0)).

- MULTIPOLYGON(((0 0 0, 4 0 0, 4 4 0, 0 4 0, 0 0 0),(1 1 0, 2 1 0, 2 2 0, 1 2 0, 1 1 0)),((-1 -1 0, -1 -2 0, -2 -2 0, -2 -1 0, -1 -1 0))).
- GEOMETRYCOLLECTIONM(POINTM(2 3 9), LINESTRINGM(2 3 4, 3 4 5)).

### Spatial Reference System Identifier (SRID)

Die SRID ist eine eindeutige Kennzahl zur Definition eines Koordinatensystems. In räumlichen Datenbanken wird die SRID benötigt, um einer Geometrie ein Koordinatensystem zuzuweisen. Ursprünglich hatten alle großen Anbieter von Geodaten eigene Kennzahlen oder waren an die European Petroleum Survey Group (EPSG) angelehnt. Seit 2005 werden die SRIDs von der Oil & Gas Producers (OGP) Surveying und dem Positioning Committee gemeinsam gewartet und festgelegt. Eine bekannte SRIDs ist zum Beispiel 4326 für WGS84.

### Analysewerkzeuge

Die PostGIS-Erweiterung zu PostgreSQL bietet die Möglichkeit, die Analysewerkzeuge aus Abschnitt 2.4.2 umzusetzen. Damit können räumliche Beziehungen, sowie räumliche Analysen zwischen Objekten durchgeführt werden.

## 3.4 Werkzeuge zur grafischen Aufbereitung der Wetterdaten

Nachdem die entsprechenden Daten in der PostgreSQL-Datenbank verwaltet werden, müssen sie in einer für den Nutzer repräsentativen Form dargestellt werden. Für die Darstellung der Daten können nun mehrere Werkzeuge verwendet werden. Dabei wurde in dieser Arbeit besonders darauf Rücksicht genommen, dass es sich um frei zur Verfügung stehende Software (OSS) handelt. Dieser Abschnitt stellt einige Möglichkeiten vor, die auch bei der Umsetzung des „Weather Twitter“ verwendet wurden, um die Wetterdaten grafisch entsprechend aufzubereiten.

### 3.4.1 MapServer

Der MapServer ist eine Open Source Rendering-Engine für Geodaten. Hauptanwendung von MapServer ist die Anzeige von dynamischen Karten im Internet. Die Engine unterstützt die Anzeige und Abfrage der gängigsten Raster-, Vektor- und Datenbankformate, verschiedene Betriebssysteme (Windows, Linux, Mac OS X etc.) und mehrere Skriptsprachen und Entwicklungsumgebungen (PHP, Python, Perl, Ruby etc.).

Entwickelt wurde MapServer ursprünglich im Rahmen des Fornet Projekts an der University of Minnesota (UMN) in Kooperation mit der NASA und dem Minnesota Department of Natural Resources (MNDNR). Später wurde MapServer von der TerraSip, einem NASA geförderten Projekt zwischen UMN und einem Konsortium übernommen. Mittlerweile hat sich das Projekt der OSGeo angeschlossen und wird von einer wachsenden Anzahl an Entwicklern (ca. 20 Stand: März 2011) weltweit gewartet (<http://mapserver.org/about.html> (Zuletzt besucht: 09.Mai 2011)).

Zur Erstellung einer MapServer Anwendung werden folgende Komponenten benötigt:

- *Mapfile*: Konfigurationsdatei mit der Dateiendung *.map*, die strukturierten Text für die MapServer-Anwendung enthält. In dieser Datei wird das Kartengebiet definiert, MapServer die Datenquellen bekannt gegeben sowie das Datenausgabeformat und der Datenausgabeort bekanntgegeben. Darüberhinaus können mehrere Layer definiert, sowie Angaben zur jeweiligen Datenquelle inklusive Filterung, Projektion und Formatierung (Symbole, Farben) gemacht werden.
- *Geodaten*: MapServer unterstützt eine Vielzahl an verschiedenen Raster- und Vektordatenformaten aus verschiedensten Datenquellen. Es können z.B. TIFF/GeoTIFF, Environmental Systems Research Institute (ESRI)-Shapedateien, sowie Daten über eine Verbindung zu einer Datenbank (PostGIS, ESRI ArcSDE, MySQL) weiterverarbeitet werden. Beim „Weather Twitter“ wurden die Daten aus der PostgreSQL-Datenbank mit PostGIS-Erweiterung verwendet.
- *HTML-Seiten*: Als Schnittstelle zwischen dem Benutzer und MapServer dienen HTML-Seiten. In der einfachsten Anwendung kann MapServer ein statisches Kartenbild in eine HTML-Seite einbinden. Mittels HTML-Formularen kann eine Karte jedoch auch interaktiv gemacht werden.

Das Mapfile ist die Zentrale vom MapServer und kann in einem beliebigen Texteditor geschrieben werden. Im Mapfile werden Beziehungen zwischen den verschiedenen Objekten definiert, sowie die Datenquellen und deren Darstellung definiert. Der Aufbau eines Mapfiles gliedert sich in mehrere Elemente. Diese Elemente werden mithilfe eines Ausschnittes der Mapfiledefinition der „Weather Twitter“-Startseite, die in den Listings 3.4 und 3.5 dargestellt ist, exemplarisch erklärt.

Zu Beginn des Files werden die grundlegenden Elemente der Karte (MAP) definiert. Diese werden in Listing 3.4 aufgelistet. Hierzu gehören der Kartenname (NAME), der Status der Karte (STATUS), das Bildausgabeformat (IMAGETYPE), die Kartenausdehnung (EXTENT), Einheit der Karte (UNITS) und die Größe der Ausgabekarte in Pixel (SIZE) sowie einige Zusatzinformationen (SHAPEPATH, SYMBOLPATH etc.). Anschließend wird die Ausgabeprojektion der Karte definiert (PROJECTION).

**Listing 3.4** – Grundelemente des Mapfiles (MAP-Objekt)

```
MAP #Beginn des Mapfiles
NAME " 'Steiermarkwetter' "#Name des Mapfiles
STATUS ON #ON oder OFF
IMAGETYPE PNG #Ausgabeformat der Karte
EXTENT 3 42 5 43 #Ausdehnung der Karte
UNITS METERS #Einheit
SIZE 1024 768 #Größe der Karte in Pixel
```



```

PROJECTION #
    "init=epsg:900913" #Zielprojektion der Karte
    z.B. Sphärische Zylinderprojektion
END
...
...
END

```

Nachdem alle wichtigen Eigenschaften der Karte definiert wurden, können die Geodaten, die in der Karte vorkommen sollen, entsprechenden Layern zugewiesen werden. Die Definition eines Layers erfolgt analog zu Listing 3.5. Dabei können jedem Layer bestimmte Eigenschaften zugewiesen werden (NAME, TYPE, STATUS, DATA, PROJECTION). Als Typ (TYPE) können dem Layer verschiedene Attribute (point, line, polygon, circle, annotation, raster, query, chart) zugewiesen werden. Die Datenquelle kann sowohl als Pfad zu einer Datei als auch als Verbindung zu einer Tabelle in einer Datenbank (Postgis, Oracle) angegeben werden. Hierfür müssen dem Mapfile die Parameter (CONNECTIONTYPE und CONNECTION) hinzugefügt werden. Außerdem können unter anderem noch der Transparenzgrad des Layers (TRANSPARENCY) sowie bei Bedarf ein Datenfilter (FILTER) zur Darstellung bestimmter Datenpunkte zum Layerobjekt hinzugefügt werden. Die Projektion bezieht sich im Layerobjekt im Gegensatz zum MAP-Objekt nicht auf die Zielprojektion, sondern auf die Ausgangsprojektion der eingebundenen Daten.

**Listing 3.5** – Grundelemente des Mapfiles (LAYER-Objekt)

```

MAP #Beginn des Mapfiles
...
...
LAYER
    NAME "Temperaturverteilung" #Layername
    TYPE POLYGON #Geometriety
    STATUS ON #Layerstatus
    CONNECTIONTYPE postgis #(optional) hier erfolgt ein Datenbankzugriff
    CONNECTION "user=... password=... dbname=witter host=129.27.89.66
        port=5431" #(optional) Verbindungsparameter
    TRANSPARENCY 100 #(optional) Transparenzeinstellung des Layers
    CLASSITEM "temperatur_hoehe_h" #(optional) Nach diesem Wert
        erfolgt die Zuordnung der Klasse
    LABELITEM #(optional) Wert für die Beschriftung
    DATA "utm_poly_geom from wt_interpolation_utm using unique id
        using srid=32633" #Datenspalte in der Datenbank

    PROJECTION #Projektion in der die Daten in der Datenbank vorliegen
        "init=epsg:32633" #Ausgangsprojektion der Karte hier: UTM 32N
    END

CLASS

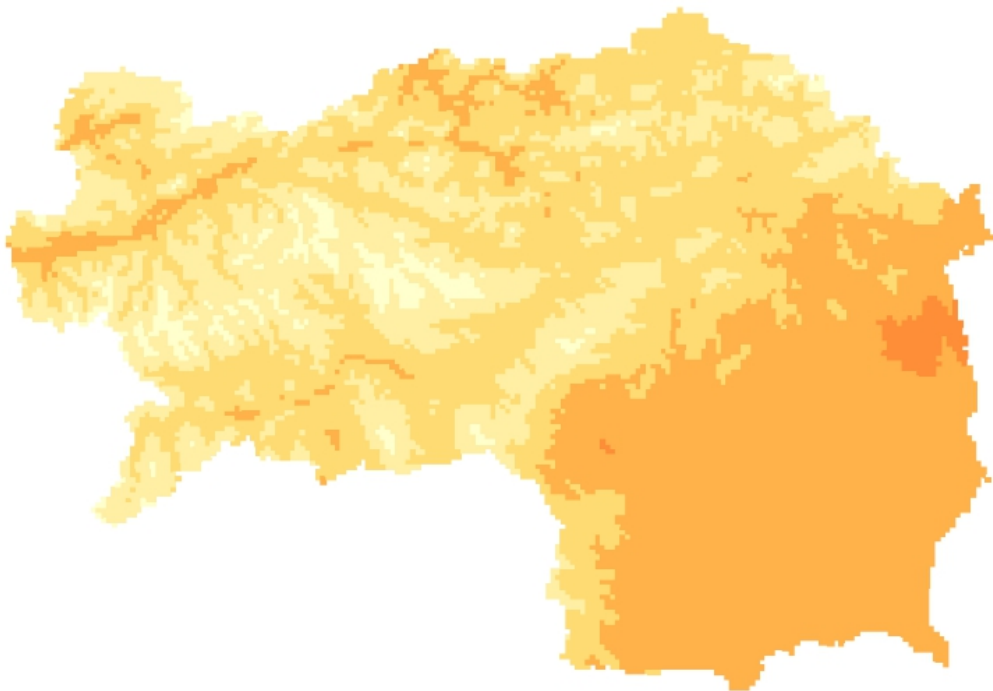
```

```

NAME "'weniger als -15°C'" #Name der Klasse, wird auch für die
    Legende verwendet
EXPRESSION ([temperatur_hoehe_h]<-15) #Zugriff auf das CLASSITEM
    und Klassifizierung nach dem Kriterium des Ausdrucks
STYLE #Aussehen der Klasse
    OUTLINECOLOR 37 52 148 #Farbe der Polygonaußenlinie
    COLOR 37 52 148 #Füllfarbe des Polygons
END #Style
END #Class
... #Weitere Klassen um alle Temperaturwerte abzudecken
... (von kleiner -15°C bis größer 31°C)
...
END #Layer
END #Map

```

Zur Darstellung der Daten können diese in mehrere Klassen (CLASS) unterteilt werden, denen wiederum verschiedene Stile (STYLE) zugewiesen werden. Optional können im Layerobjekt verschiedene Datenquellen definiert werden nach denen die Daten klassifiziert (CLASSITEM) bzw. beschriftet (LABELITEM) werden. Das in diesem Abschnitt definierte Mapfile liefert nach Abfrage über den MapServer mit der URL [http://129.27.89.66/cgi-bin/mapserv?map=F:/ms4w/Apache/htdocs/student/master\\_witter/Weather\\_Twitter/Maps/index\\_map.map&mode=map&layer=Temperaturverteilung](http://129.27.89.66/cgi-bin/mapserv?map=F:/ms4w/Apache/htdocs/student/master_witter/Weather_Twitter/Maps/index_map.map&mode=map&layer=Temperaturverteilung) das Bild in Abbildung 3.4, welches die Temperaturverteilung der Steiermark repräsentiert.



**Abbildung 3.4** – Temperaturverteilung als Ausgabebild des MapServers

Weiterführende Information zum Aufbau von Mapfiles können der MapServer-Webseite<sup>6</sup> entnommen werden.

### 3.4.2 GeoServer

Ein Werkzeug, um Geodaten zu verwalten, zu bearbeiten und anzubieten ist GeoServer, der im Jahr 2001 von The Open Planning Project (TOPP) gestartet wurde. TOPP hatte die Vision, analog zum World Wide Web ein sogenanntes Geospatial Web zu entwickeln, wo die Benutzer räumliche Daten suchen und downloaden können. Datenanbietern sollte die Möglichkeit geschaffen werden ihre Daten in diesem Netz anzubieten und den Usern den direkten Zugriff darauf zu gewähren. Diese Möglichkeit steht im Gegensatz zu den indirekten und umständlichen Methoden des Datenaustausches heutzutage. GeoServer wurde in der Programmiersprache Java entwickelt und steht als Open Source Software Server zur freien Nutzung zur Verfügung.

Zur gleichen Zeit als die Entwicklung von GeoServer startete, wurde auch an der Entwicklung des WFS-Standards gearbeitet. Auch andere Projekte wie die Datenbankeerweiterung PostGIS sowie das Open Source Kartendarstellungstool OpenLayers wurden zur gleichen Zeit entwickelt und flossen in das Projekt GeoServer mitein. Die Daten aus GeoServer können heute auch in sehr viele andere Projekte z.B. Google Earth importiert werden. Derzeit wird daran gearbeitet GeoServer-Daten direkt über Google Maps suchbar zu machen und somit den Zugriff auf Geodaten für die Allgemeinheit im Sinne der Interoperabilität weiter zu erleichtern.

Darüberhinaus kann GeoServer alle Geodatenformate verarbeiten, die den OGC-Standards genügen, und dient als Referenzentwicklung für die OGC-Standards WFS und Web Coverage Service (WCS). Natürlich wird auch der grundlegende WMS-Standard unterstützt (GeoServer-Dokumentation (Zuletzt besucht: 09.Mai 2011)).

## 3.5 Grafische Darstellung der Wetterdaten im Webbrowser mit Hilfe von OpenLayers

Zur entgeltigen Darstellung der Wetterdaten in der WebGIS-Applikation „Weather Twitter“ wurde die JavaScript-Bibliothek OpenLayers verwendet. Ähnlich wie Google Maps und Microsoft Virtual Earth wurde OpenLayers entwickelt, um webbasierte Geoapplikationen zu erstellen. Der Hauptunterschied zu seinen Konkurrenzprodukten liegt bei OpenLayers jedoch darin, dass sein Sourcecode als freie Software jedem zur Verfügung steht. Außerdem wird es von einer breiten Community verwendet, getestet und weiterentwickelt.

Von OpenLayers werden mehrere Schnittstellen zur Verfügung gestellt, um Geodaten einzubinden. Darunter befinden sich die OGC-Spezifikationen WFS und WMS. Neben den OSM-Daten können auch geschlossene Formate, wie Google Maps oder Virtual Earth, in OpenLayers eingebunden werden. Zusätzlich stellt OpenLayers verschieden Werkzeuge zur Verfügung, die es

---

<sup>6</sup><http://mapserver.org/mapfile>

ermöglichen in der Karte zu navigieren (pannen, zoomen). Außerdem wird eine Layerauswahl oder ein Übersichtsfenster und einige weitere Funktionen bereitgestellt.

### 3.5.1 Aufbau eines OpenLayers-Dokumentes

Der Aufbau eines OpenLayers-Dokumentes ist äquivalent zur Struktur einer Javascript-Datei, so werden auch die OpenLayers-Dokumente als JavaScript (*name.js*) abgespeichert. Um auf die Elemente der JavaScript-Bibliothek zugreifen zu können, muss diese entweder in der JavaScript-Datei oder im Header des HTML-Dokumentes geladen werden. Bei JavaScript handelt es sich um eine objektorientierte Skriptsprache, wodurch auch in OpenLayers Objekte eine entscheidende Rolle spielen. Die wichtigsten Objekte werden im nachfolgenden Abschnitt vorgestellt.

### 3.5.2 Objekte in OpenLayers

Der Hauptgrund, warum man OpenLayers in eine Webanwendung einbindet, ist die Möglichkeit, Kartendarstellungen schnell und einfach in eine Anwendung einzubauen. Das Grundelement jedes OpenLayers-Dokumentes ist somit die Definition der Karte (*map*). Dafür dient in OpenLayers das sogenannte Mapobjekt (Listing 3.6).

#### Mapobjekt

Das Mapobjekt enthält alle wichtigen Eigenschaften der Karte. Grundsätzlich ist zu beachten, dass zur Erzeugung eines neuen Objektes in OpenLayers, bis auf die Optionsparameter, alle Parameter benötigt werden.

**Listing 3.6** – Das Mapobjekt in OpenLayers

```
var map = new OpenLayers.Map("map-name", options);
```

Im Fall des Mapobjektes ist nur der Kartename, der zusätzlich die id des `<div>`-Tag in HTML ist, ein vorgeschriebener Parameter. Alle anderen Einstellungen sind optional und können dem Mapobjekt auch im Laufe des OpenLayers-Dokumentes noch hinzugefügt werden. Darunter befinden sich unter anderem (OpenLayers-Bibliothek (Zuletzt besucht: 07.Juli 2011)):

- *events*: Ein Eventobjekt beinhaltet jegliche manipulation der Karte (Move, Click etc.).
- *controls*: Navigationsleiste, PanZoomBar, Attribution etc.
- *layers*: Enthält eine geordnete Liste von Layern in der Karte.
- *baseLayer*: Definiert den derzeit gewählten Layer.
- *tileSize*: Bestimmt die Größe der Kartenblätter.
- *maxResolution*: Maximale Auflösung der Karte.
- *maxExtent*: Maximale Ausdehnung der Karte.

- *restrictedExtent*: Einschränkung der Kartenausdehnung.
- *units*: Karteneinheit.
- *projection*: Projektion der Karte.

Die Definition eines dieser optionalen Objekte erfolgt jeweils mit dem Befehl *new OpenLayers.Object*. Beispielhaft werden neben dem Mapobjekt einige in dieser Arbeit verwendete Objekte vorgestellt.

### Layerobjekt

Ein Layerobjekt wird mit dem in Listing 3.7 dargestellten Befehl erstellt.

**Listing 3.7** – Layerobjekt in OpenLayers

```
var layer = new OpenLayers.Layer.Layertype("layer-name", options);
```

Dabei kann zwischen mehreren Layertypen unterschieden werden:

- *WMS*: Web Map Service-Layer, Parameter: Layername (name), Layerpfad (url), Layerparameter (params) enthält Mapfilepfad, Mapfilelayer und optional die Mapfileoptions (transparent, styles, format, filter etc.). Ebenfalls optional sind die Layeroptionen (options).
- *WFS*: Web Feature Service-Layer, Parameter: Layername (name), Layerpfad (url), Layerparameter (params) und Layeroptionen (options).
- *OSM*: OpenStreetMap-Layer, Parameter: Layername (name).
- *ArcGIS*, *GML*, *Google*, *Grid*, *Image*, *Markers*, *Text*, *Vector*, *VirtualEarth*, *WorldWind*, *Yahoo* uvm.

Beispielhaft wird der Aufbau des Layerobjektes mit der Temperaturverteilung über die Steiermark (Listing 3.8) dargestellt, der als WMS-Layer realisiert ist.

**Listing 3.8** – Layerobjekt mit Temperaturverteilung der Steiermark in OpenLayers

```
var stmk_temperaturverteilung = new OpenLayers.Layer.WMS(  
    "Temperaturverteilung", //Layername  
    "http://129.27.89.66/cgi-bin/mapserv?" //Pfad zum Mapserver  
    {  
        map: ".../Weather_Twitter/Maps/index_map.map" //Pfad zum Mapfile  
        layers: "Temperaturverteilung" //Layername im Mapfile  
        transparent: "true" //Transparenzeinstellung  
    },  
    {  
        sphericalMercator: "true" //Projektionseinstellung  
    }  
);
```

## Controlobjekt

Ein Controlobjekt wird mit dem in Listing 3.9 dargestellten Befehl erstellt.

**Listing 3.9** – Controlobjekt in OpenLayers

```
var control = new OpenLayers.Control.Controltype();
```

Ein Controlobjekt hat die Aufgabe, die Darstellung bzw. das Verhalten der Karte zu beeinflussen. Dabei kann zwischen verschiedenen Controltypen unterschieden werden, wobei an dieser Stelle nur eine kleine Auswahl vorgestellt wird (OpenLayers-Controls (Zuletzt besucht: 07.Juli 2011)):

- *Navigation()*: Dieses Control steuert die Navigation in der Karte mittels diverser Mausbefehle (Drag, Klick, Doppelklick, Scrollen des Mousrads). Es wird defaultmäßig der Karte hinzugefügt.
- *PanZoomBar()*: Links oben in der Ecke des Kartenfensters werden 4 Navigationspfeile und ein vertikaler ZoomBar dargestellt.
- *LayerSwitcher()*: Erlaubt dem Nutzer die Anzeige von verschiedenen Layern ein- und auszuschalten.
- *MousePosition()*: Anzeige der Koordinaten der Maus.
- *OverviewMap()*: Ermöglicht die Anzeige einer Übersichtskarte.
- *Click()*: Klickhandler, der nach einem Mausklick in die Karte bestimmte Funktionen aufruft und ausführt.
- *Geolocate()*: Ermöglicht den Einbau der W3C Geolocation API in die OpenLayers Karte.

Beispielhaft wird der Aufbau eines Controls anhand des Geolocation-Controls (Listing 3.10) erklärt. Dieses Control wurde erst Anfang des Jahres 2011 in die Liste der OpenLayers-Controls aufgenommen und ermöglicht die Einbindung der W3C Geolocation API in eine OpenLayers-Anwendung. Damit kann die Position des Nutzers der Anwendung bestimmt werden.

**Listing 3.10** – Controlobjekt Geolocate zur Bestimmung der Position

```
//Erzeugen des Geolocation-Controls
var geolocation = new OpenLayers.Control.Geolocate();
//Hinzufügen des Geolocation-Controls zur Karte
map.addControl(geolocation);

OpenLayers.Control.Geolocate = OpenLayers.Class(OpenLayers.Control, {
    //mögliche Eventtypen
    EVENT_TYPES: ["locationupdated", "locationfailed", "locationuncapable"],

    //Positionsobjekt
    geolocation: navigator.geolocation,
```

```
//API Properties -> Eigenschaften der Schnittstelle
bind: true,
watch: false,
geolocationOptions: null,

//Initialisieren eines neuen OpenLayers.Control
initialize: function(options) {
    this.EVENT_TYPES =
        OpenLayers.Control.Geolocate.prototype.EVENT_TYPES.concat(
            OpenLayers.Control.prototype.EVENT_TYPES
        );
    this.geolocationOptions = {};
    OpenLayers.Control.prototype.initialize.apply(this, [options]);
},
//Method: activate
//Activate the control
activate: function() {
    if (!this.geolocation) {
        this.events.triggerEvent("locationuncapable");
        alert("Ihre Position konnte nicht bestimmt werden!  
Bitte mit Klick positionieren!");
        return false
    }
    if (!this.active) {
        if (this.watch) {
            this.watchId = this.geolocation.watchPosition(
                OpenLayers.Function.bind(this.geolocate, this),
                OpenLayers.Function.bind(this.failure, this),
                this.geolocationOptions
            );
        } else {
            this.geolocation.getCurrentPosition(
                OpenLayers.Function.bind(this.geolocate, this),
                OpenLayers.Function.bind(this.failure, this),
                this.geolocationOptions
            );
        }
    }
    return OpenLayers.Control.prototype.activate.apply(
        this, arguments
    );
},
//Method: deactivate
deactivate: function() {
    if (this.active && this.watchId !== null) {
        this.geolocation.clearWatch(this.watchId);
    }
    return OpenLayers.Control.prototype.deactivate.apply(
        this, arguments
    );
},
//Method: geolocate
geolocate: function(position) {
    var center = new OpenLayers.LonLat(
        position.coords.longitude,
        position.coords.latitude
    );
}
```

```

    ).transform(
        new OpenLayers.Projection("EPSG:4326"),
        this.map.getProjectionObject()
    );
    if (this.bind) {
        this.map.setCenter(center, 16); //Center Zoom
        //document.getElementById('posname').title = position.coords.latitude;
        document.getElementById("lat").value = position.coords.latitude;
        document.getElementById("lon").value = position.coords.longitude;

        //SQL_Command(Abfrage von Gemeinde, Temperatur und Icon)
        var sql_command = "SELECT gemnam, round(temperatur_hoehe_h), weather_icon
        FROM wt_interpolation_utm, gemeinden
        WHERE st_contains (transform(utm_poly_geom,4326), geometryfromtext(
        'POINT('+position.coords.longitude+" "+position.coords.latitude+"')',4326))
        AND st_contains (transform(the_geom,4326), geometryfromtext(
        'POINT (" +position.coords.longitude+" "+position.coords.latitude+"')',4326))";
        var id = "1";
        processSpatialQuery(sql_command, id);

        map.addLayer(vlayer);
        //Style des Geometrieobjektes
        var style = OpenLayers.Util.extend({}, OpenLayers.Feature.Vector.style['default']);
        style.externalGraphic = document.getElementById("myimg").src;

        style.graphicOpacity = 0.8;
        style.graphicWidth = 64; //style.externalGraphic;
        style.graphicHeight = 64;

        var point = new OpenLayers.Geometry.Point(center["lon"], center["lat"]);
        var pointFeature = new OpenLayers.Feature.Vector(point, null, style);
        vlayer.removeAllFeatures();
        vlayer.addFeatures([pointFeature]);

        alert("Ihre Position wurde mittels W3C Geolocation API bestimmt!\n\n
        Falls die Position nicht stimmt, mit einem Klick in die Karte verbessern!");
    }
    this.events.triggerEvent("locationupdated", {
        position: position,
        point: new OpenLayers.Geometry.Point(
            center.lon, center.lat
        )
    });
},
// Method: failure
failure: function (error) {
    this.events.triggerEvent("locationfailed", {error: error});
    alert("Ihre Position konnte mittels Geolocation API leider nicht bestimmt werden!
    Bitte mit Klick in die Karte positionieren!");
},

CLASS_NAME: "OpenLayers.Control.Geolocate"
});

```



## Stylemap

Mit einem Stylemapobjekt (Listing 3.11) können einer Geometrie in OpenLayers mehrere verschiedene Stile zugewiesen werden. Dafür werden der Stylemap für bestimmte Eigenschaften (z.B. „default“, „select“) Styleobjekte zugewiesen

**Listing 3.11** – Stylemapobjekt in OpenLayers

```
var stylemap = new OpenLayers.Stylemap(style, options);
```

## Style

Die Erzeugung eines Styleobjektes in OpenLayers ist in Listing 3.12 aufgezeigt.

**Listing 3.12** – Styleobjekt in OpenLayers

```
var style = new OpenLayers.Style(style, options);
```

Ein Styleobjekt definiert das Aussehen einer Geometrie. Dabei können eine Vielzahl an Eigenschaften definiert werden. Unter anderem lassen sich folgende Kriterien festlegen:

- *pointRadius*: Radius eines Punktes.
- *fillColor*, *fillOpacity*: Füllfarbe und Opazität der Füllfarbe (0-1) eines Objektes.
- *strokeColor*, *strokeOpacity*, *strokeWidth*, *strokeLinecap*: Farbe, Opazität (0-1) und Stärke der Umrandung eines Objektes, sowie Form des Linienabschlusses (abgerundet, quadratisch).
- *backgroundGraphic*: relativer Pfad zu einer Hintergrundgrafik.
- *graphicWidth*, *graphicHeight*, *graphicOpacity*, *graphicXOffset*, *graphicYOffset*, *rotation*, *graphicName*: Größe und Ausrichtung der Grafik, sowie Grafikname.
- *label*: Hintergrundbeschriftung.
- *labelAlign*, *labelXOffset*, *labelYOffset*: Ausrichtung der Beschriftung.
- *fontColor*, *fontSize*, *fontWeight*: Schriftfarbe, Schriftgröße und Schriftstärke.

Die Beschriftung eines WFS-Layers kann auch anhand der Werte in einer Datenbankspalte umgesetzt werden. Dafür kann im *label* mittels \$(spaltenname) auf die Datenbankspalte zugegriffen werden.

## 3.6 W3C Geolocation API – Positionierung mittels Webbrowser

In folgendem Abschnitt wird nun auf die W3C Geolocation API und die damit verbundene Möglichkeit der Positionsermittlung mit Hilfe eines Webbrowsers eingegangen. Dabei wird der Begriff „Geolocation“ ebenso erklärt, wie die Entwicklung der Positionsbestimmung im World Wide Web. Die Grundlagen zu diesem Kapitel wurden bereits in Abschnitt 2.3 beschrieben.

### 3.6.1 Einleitung

Unter Geolocation versteht man im Allgemeinen die Zuordnung einer geographischen Position zu einem Objekt. Dabei kann es sich um ein Mobiltelefon genauso, wie um einen mit dem Internet verbundenen Desktop PC handeln. Der Begriff „Geolocation“ bezieht sich hier auf die Feststellung der aktuellen Position, wird jedoch auch als Bezeichnung für die aktuelle Position verwendet. Ein wesentlicher Unterschied zur Positionierung besteht darin, dass der Schwerpunkt bei der Geolocation auf dem Auffinden bzw. der Zuordnung einer Lokation (Adresse, Ortsname etc.) liegt, und nicht alleine die Bestimmung der geographischen Koordinaten von Bedeutung ist.

Bei der Geolocation wird versucht jegliche verfügbare Information, die Aufschluss über den Aufenthaltsort des benutzten Gerätes liefert, zu verarbeiten und mit einer Position in Verbindung zu bringen. Hierfür können die verschiedensten Technologien eingesetzt werden. Sowohl GPS, WLAN, Bluetooth oder die Mobilfunk Cell-ID, als auch Methoden der terrestrischen Radionavigation wie z.B. Time of Arrival (ToA) oder Time Difference of Arrival (TDoA) (siehe Abschnitt 2.3) werden verwendet. Darüberhinaus kommt auch der Positionierung mittels IP-Adresse eine große Bedeutung zu. Die einzelnen Technologien haben jedoch gewisse Vor- und Nachteile, wodurch sie sich nur für bestimmte Anwendungszwecke eignen.

### 3.6.2 Vorteile und Nachteile der verschiedenen Technologien

Wie bereits in der Einleitung zur Geolocation (3.6.1) vorgestellt, können zur Positionierung eines Gerätes, das auf eine Webseite zugreift, mehrere Technologien verwendet werden (Presentation-Slides-Ryan-Sarver (Zuletzt besucht: 03.Juli 2011)):

1. IP Geolocation.
2. Triangulation:
  - GPS: Genauigkeit 10m.
  - Wi-Fi: Genauigkeit 20m.
  - Cell-ID: Genauigkeit 1000m.

Diese Methoden unterscheiden sich durch spezifische Vorteile (+) und Nachteile (–), die in folgendem Abschnitt kurz erläutert werden.

#### IP-Geolocation

Die IP-Geolocation wird aktuell von drei großen Unternehmen (Maxmind<sup>7</sup>, Quova<sup>8</sup>, IP2Location<sup>9</sup>) angeboten, die dafür Dienste und Server bereitstellen. Das Prinzip der IP Geolocation wurde in Abschnitt 2.3.1 vorgestellt. Folgende charakteristische Eigenschaften beschreiben die Positionierung mithilfe der IP-Adresse:

---

<sup>7</sup><http://www.maxmind.com>

<sup>8</sup><http://www.quova.com>

<sup>9</sup><http://www.ip2location.com>

- + *Verbreitung*: Weit verbreitet, da jeder Internetbenutzer über eine IP-Adresse verfügt, die einer Position zugeordnet werden kann.
- + *Methode*: Serverbasierte Detektierung.
- *Genauigkeit*: IP-Geolocation ist nur in Städten genau. In den USA wird immerhin in ca. 85% der Fälle der richtige Staat detektiert.
- *Kosten*: Teuer, aufgrund der benötigten Serverkapazitäten und deren Wartung.
- *Zuverlässigkeit*: Die Methode liefert auch grob falsche Ergebnisse, die jedoch nicht immer auf den ersten Blick erkannt werden.

### Geolocation mittels GPS

GPS bietet im Vergleich zur IP-Geolocation vor allem in wenig bebauten Regionen wesentlich genauere Ergebnisse und gewinnt durch die Verbreitung von Mobiltelefonen mit GPS-Empfängern immer mehr an Bedeutung.

- + *Genauigkeit*: Sehr genau, sobald die erste Positionslösung (Position Fix) erhalten wird. Gute Genauigkeit in ländlichen Regionen (wenige Störfelder).
- + *Aufwand*: Geringer Aufwand. Nur ein GPS-Empfänger wird benötigt, der heute nahezu in jedem neuen mobilen Gerät eingebaut ist.
- *Zuverlässigkeit*: Probleme ergeben sich im städtischen Bereich. Außerdem funktioniert auch der Betrieb in Gebäuden nicht zufriedenstellend.
- Dauert oft lange, bis die erste Positionslösung ermittelt werden kann.

### Wi-Fi-Positioning

Die Positionierung mittels WLAN-Netz wurde im Abschnitt 2.3.1 vorgestellt. Ein Tool zur Positionierung mittels WLAN-Zugriffspunkten wird von Loki<sup>10</sup> angeboten. Ein Anbieter von Wi-Fi-Access Points ist die Firma Skyhook<sup>11</sup>. Skyhook wurde im Jahr 2003 gegründet, um den steigenden Bedarf zur genauen Positionsbestimmung für LBS zu nutzen. Die bis dahin verwendeten Technologien wie GPS oder die Zellpositionierung lieferten nur unbefriedigende Ergebnisse. So wurde mit der Entwicklung eines Systems (Core Engine SDK) begonnen, das WLAN-Access Points nutzt, um auch Indoorpositionen zu bestimmen. Dafür wurden auch weltweit Aufzeichnungen der Netzwerkzugangspunkte<sup>12</sup> (siehe Abbildung 3.5) durchgeführt. Die Abbildung zeigt auch schön die Probleme der Wi-Fi-Positionierung durch fehlende WLAN-Zugangspunkte bzw. nicht aufgezeichnete Punkte im ländlichen Bereich auf.

Die Vor- und Nachteile der Wi-Fi-Positionierung werden nachfolgend angeführt.

---

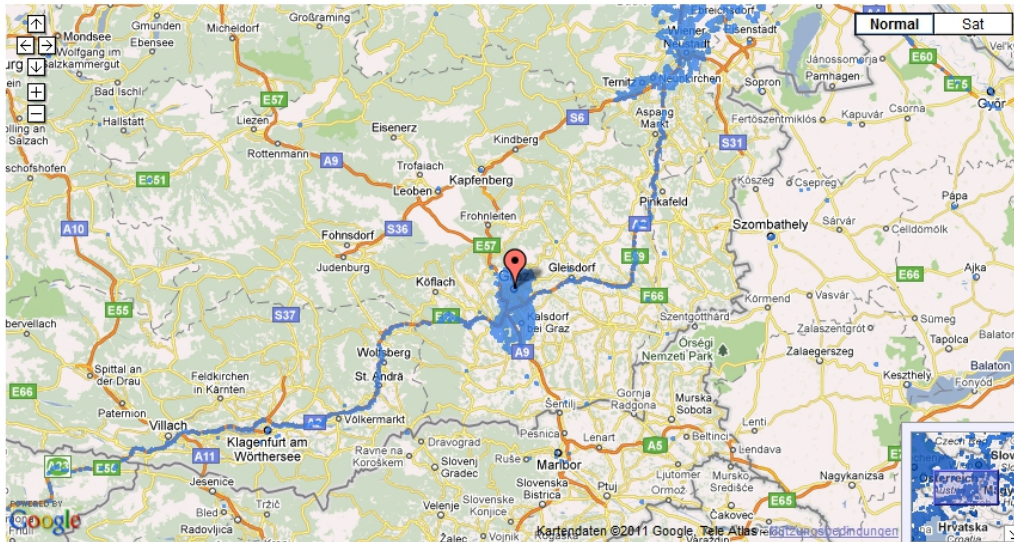
<sup>10</sup><http://www.loki.com>

<sup>11</sup><http://www.skyhookwireless.com>

<sup>12</sup><http://www.skyhookwireless.com/howitworks/coverage.php>

## Check Coverage in Your Area

Type in your address and click Find It.



**Abbildung 3.5** – Blaue Punkte zeigen die Abdeckung der Steiermark mit WLAN-Zugangspunkten

- + *Genauigkeit*: Hohe Genauigkeit, wenn Wi-Fi-Zugangspunkte zur Verfügung stehen.
- + Schnelle erste Positionslösung (Position Fix).
- + *Aufwand*: Niedrig, da die bestehenden Wi-Fi-Knoten genutzt werden, und nur eine Software zur Auswertung benötigt wird.
- *Zuverlässigkeit*: Probleme im ländlichen Bereich aufgrund weniger WLAN Zugriffspunkte.

### Cell-ID

Die Positionierung über die Cell-ID in Zellularen Netzwerken wurde in Abschnitt 2.3.1 behandelt. Auch dieses Verfahren bietet bestimmte Vor- und Nachteile.

- + *Zuverlässigkeit*: Funktioniert, sobald es Mobilfunkmasten gibt.
- + Schnelle erste Positionslösung.
- + *Aufwand*: Gering, nur Software wird benötigt.
- *Genauigkeit*: Sehr ungenau in großen Mobilfunkzellen. Verbesserung durch Zusatzverfahren.

### Zusammenfassung

Aus den Vor- und Nachteilen der verschiedenen Positionierungsmethoden (Tabelle 3.1) lässt sich der Schluss ziehen, dass eine möglichst optimale Positionslösung nur mit der Nutzung von mehreren Technologien erzielbar ist. Somit müssen hybride Ansätze verfolgt werden, die verschiedene Technologien vereinen. Genau diesen Ansatz verfolgt die W3C Geolocation API.

**Tabelle 3.1** – Vor- und Nachteile verschiedener Positionierungsmethoden

Methode	Genauigkeit		Einsatzbereich		Sonstiges	
	Stadt	Land	Indoor	Outdoor	Kosten	Position fix
IP Geolocation	+	–	+	+	–	+
GPS	–	+	–	+	+	–
WPS	+	–	+	+	+	+
Cell-ID	+	–	+	+	–	+

### 3.6.3 Entwicklung der W3C Geolocation API

Der Versuch eine Geolocation Schnittstelle zu definieren, um den Standort eines Gerätes (Mobiltelefon, PDA, PC etc.) mittels Browser zu ermitteln, wurde zu Beginn des 21. Jahrhunderts begonnen. Viele große Internetdienstleister haben den Nutzen von Geolocation erkannt und schlossen ihre Projekte zu einer gemeinsamen Initiative zusammen. Die großen Internetdienstleister Google, Mozilla, Microsoft, Opera und Apple begannen 2009 mit einer Kooperation, um mit der W3C Geolocation API eine Schnittstelle zur Ermittlung der Userposition über einen Webbrowser zu entwickeln.

Dabei konnte auf eine Vielzahl bereits bestehender Entwicklungen zurückgegriffen werden. So existieren bereits seit längerem Dienste, um die Position eines Rechners über die IP-Adresse feststellen zu können. Auch die Positionierung über die WLAN-Zugangspunkte wird seit längerer Zeit verfolgt. Die Grundlagen der W3C Geolocation API gehen auf Arbeiten von Personen und Firmen zurück, die sich bereits mit der Thematik befasst haben. Vor allem die Arbeiten von Aza Raskin, die Google Gears Geolocation API, sowie die Arbeiten von Ryan Sarver stellen die Grundlage der W3C Geolocation API dar.

#### Google Gears Geolocation API

Die Google Gears Geolocation API gilt als eine der wichtigsten Grundlagen zur Entwicklung der W3C Geolocation API. Die erste Version von Google Gears wurde im Mai 2007 veröffentlicht, wobei Gears erst im August 2008 um die Geolocation Funktion erweitert wurde. In der Gears Geolocation API wurden eine Reihe von ECMAScript konformen Objekten definiert mit deren Hilfe der Standort eines Gerätes bestimmt werden kann.

Webseiten können diese Schnittstelle auf zwei Arten nutzen. Entweder direkt über den Webbrowser oder über das Google Gears Plugin (siehe <http://code.google.com/intl/de-DE/apis/gears> (Zuletzt besucht: 27. April 2011)). Ohne Plugin wird Google Gears Geolocation API derzeit (Juni 2011) von folgenden Browsern unterstützt:

- Firefox 3.5+.
- GoogleChrome.
- Opera 10.6.

Auf Mobiltelefonen unterstützen die API:

- iPhone.
- Android (firmware 2.0+).

Die Gears Geolocation API erlaubt es einer Web-Applikation, die aktuelle Position des Benutzers zu ermitteln (*getCurrentPosition*), Veränderungen in der Position festzustellen (*watchPosition*) bzw. die letzte ermittelte Position des Benutzers anzuzeigen (*lastPosition*). Der Zugriff auf die im Gerät vorhandene Positionsinformation muss vom Benutzer erst mittels Bestätigung in einem Dialogfeld erlaubt werden. Danach bietet die API die bestmögliche Positionslösung an, indem vom System alle verfügbaren Informationen von verschiedenen Positionsanbietern (Location Provider) verwendet werden. Diese Anbieter können im Gerät implementiert sein (z.B. GPS-Empfänger), aber auch als serverbasierte Positionsanbieter (Network Location Provider (NLP)) arbeiten.

Die Gears Geolocation API benutzt alle Informationen, die von den Location Providern zur Verfügung gestellt werden, und Hinweise auf die aktuelle Position des Clients liefern. Sowohl GPS-Daten, wie auch die IP-Adresse oder andere zur Positionsbestimmung verwendbaren Daten, wie Cell-ID's oder Wi-Fi-Knotenpunkte werden gesammelt. Diese Parameter werden an einen Network Location Server gesendet (in diesem Fall jener von Google (code.l.google.com)), um eine Positionslösung zu erhalten. Der Server gleicht die übermittelten Parameter mit einer Liste von Mobile Cell Towers sowie deren Signalstärken, oder einer Liste von detektierten Wi-Fi-Netzwerken und deren Signalstärken ab. In Zusammenhang mit den Wi-Fi-Netzwerken, sei die Problematik<sup>13</sup> rund um die Google Street View Autos erwähnt, welche bei ihren Kamerafahrten ohne Genehmigung auch Informationen zu den Wi-Fi-Zugangspunkten gesammelt haben.

Die Kommunikation zwischen dem Server und dem Client funktioniert über HTTP, wobei Gears HTTP POST-Requests verwendet. Sowohl die Anfrage (Listing 3.13) als auch die Antwort darauf (Listing 3.14) sind im JavaScript Object Notation-Format (JSON) definiert.

**Listing 3.13** – Google Gears Request

```
{
  "version": "1.1.0",
  "host": "maps.google.com",
  "access_token": "2:k7j3G6LaL6u_lafw:4iXOeOpTh1glSXe",
  "home_mobile_country_code": 310,
  "home_mobile_network_code": 410,
  "radio_type": "gsm",
  "carrier": "Vodafone",
  "request_address": true,
  "address_language": "en_GB",
  "location": {
    "latitude": 51.0,
```

<sup>13</sup>[http://diepresse.com/home/techscience/internet/google/609381/Street-View\\_USBehoerde-ermittelt-wegen-WLANDaten](http://diepresse.com/home/techscience/internet/google/609381/Street-View_USBehoerde-ermittelt-wegen-WLANDaten)

```
    "longitude": -0.1
  },
  "cell_towers": [
    {
      "cell_id": 42,
      "location_area_code": 415,
      "mobile_country_code": 310,
      "mobile_network_code": 410,
      "age": 0,
      "signal_strength": -60,
      "timing_advance": 5555
    },
    {
      "cell_id": 88,
      "location_area_code": 415,
      "mobile_country_code": 310,
      "mobile_network_code": 580,
      "age": 0,
      "signal_strength": -70,
      "timing_advance": 7777
    }
  ],
  "wifi_towers": [
    {
      "mac_address": "01-23-45-67-89-ab",
      "signal_strength": 8,
      "age": 0
    },
    {
      "mac_address": "01-23-45-67-89-ac",
      "signal_strength": 4,
      "age": 0
    }
  ]
}
```

JSON ist ein textbasierter Open Standard, der zum Datenaustausch definiert wurde und sowohl vom Computer als auch vom Menschen leicht gelesen werden kann. Die Darstellung der Daten ist stark an die Objektdefinition in JavaScript angelehnt. In der Anfrage an Google sind alle verfügbaren Informationen zur möglichen Position des Mobiltelefons enthalten.

**Listing 3.14** – Google Gears Server Response

```
{
  "location": {
    "latitude": 51.0,
    "longitude": -0.1,
    "altitude": 30.1,
    "accuracy": 1200.4,
    "altitude_accuracy": 10.6,
```

```
"address": {
  "street_number": "100",
  "street": "Amphibian Walkway",
  "postal_code": "94043",
  "city": "Mountain View",
  "county": "Mountain View County",
  "region": "California",
  "country": "United States of America",
  "country_code": "US"
},
"access_token": "2:k7j3G6LaL6u_lafw:4iXOeOpTh1glSXe"
}
```

Basierend auf die übermittelten Parameter berechnet der Network Location Provider eine Position, wobei der genaue Algorithmus von Google nicht verraten wird (Geolocation-API-Network-Protocol (Zuletzt besucht: 07.Juli 2011)).

### 3.6.4 Die W3C Geolocation API

#### Einleitung

Vom World Wide Web Consortium (W3C) wurde im Jahr 2008 aufbauend auf die in Abschnitt 3.6.3 vorgestellten Entwicklungen begonnen an einem Standard zu arbeiten, der das Thema Geolocation zum Inhalt hat. Die sogenannte Geolocation API definiert eine High-Level-Schnittstelle zur Positionsinformation (Längen- und Breitengrad) des Gerätes, auf dem eine Applikation ausgeführt wird. Das Ziel der Schnittstelle ist es, aufbauend auf bereits bestehende Technologien eine Spezifikation zu schaffen, die möglichst unabhängig von der zugrunde liegenden Positionsinformation ist. In der Spezifikation sind zwar Funktionen und Datenstrukturen definiert, jedoch wird nicht vorgeschrieben, wie die Schnittstelle ihre Positionsinformation erhält. Die genaue Umsetzung der W3C Geolocation API ist nicht spezifiziert, jedoch baut sie sehr stark auf bestehende Technologien wie die Google Gears Geolocation API auf. Auch die Firefox Geolocation Umsetzung greift auf den Network Location Provider von Google zurück (WebScanNotes (Zuletzt besucht: 23.Juni 2011)).

Die W3C Geolocation API unterstützt sowohl einfache Positionsabfragen, als auch Positionsupdates um die Position von bewegten Objekten zu verfolgen. Die Positionsinformation wird dabei als Längen- und Breitenkoordinate im WGS84 System aufbereitet.

#### Sicherheitsaspekt

Wichtiges Anliegen bei der Entwicklung der Schnittstelle ist die Berücksichtigung der Privatsphäre und Sicherheit der Anwender. So kann die Position eines Gerätes, und damit verbunden des Users, nur mit dessen Zustimmung erfolgen. Dafür wird im Webbrowser eine eigene Toolbar geöffnet, in der die Positionierung bestätigt werden muss.



## Beschreibung der W3C Geolocation API

Um die Positionsinformation eines angebundenes Gerätes zu erhalten und dessen Position zu bestimmen, wird von verschiedenen Skripten das *Geolocation* Objekt (Tabelle 3.15) verwendet. Die Positionsinformation wird über einen dem entsprechenden Endgerät angepassten Algorithmus gewonnen, indem im *Geolocation* Objekt ein Positionsobjekt (*Position*) erzeugt und mit Daten befüllt wird.

**Listing 3.15** – *Geolocation* Objekt

```
[NoInterfaceObject]
interface Geolocation {
    void getCurrentPosition(in PositionCallback successCallback ,
                           in optional PositionErrorCallback errorCallback ,
                           in optional PositionOptions options);

    long watchPosition(in PositionCallback successCallback ,
                      in optional PositionErrorCallback errorCallback ,
                      in optional PositionOptions options);

    void clearWatch(in long watchId);
};

[Callback=FunctionOnly , NoInterfaceObject]
interface PositionCallback {
    void handleEvent(in Position position);
};

[Callback=FunctionOnly , NoInterfaceObject]
interface PositionErrorCallback {
    void handleEvent(in PositionError error);
};
```

Dabei wird zwischen drei verschiedenen Positionsobjekten unterschieden:

- *getCurrentPosition*.
- *watchPosition*.
- *clearWatch*.

Das *getCurrentPosition* Objekt umfasst ein bis drei Argumente, die nach Aufruf der Methode an das Gerät zurückgegeben werden:

- *PositionCallback*: Im Fall einer erfolgreichen Positionsbestimmung wird von der API an das Gerät ein *successCallback* gesendet, der die aktuelle Position des Gerätes beinhaltet.
- *PositionErrorCallback*: Wenn die Position nicht ermittelt werden kann, liefert die Schnittstelle einen *errorCallback*, der den Grund für die fehlgeschlagene Positionsermittlung aufzeigt.

- *PositionOptions*: Dieses Objekt liefert die Optionen der Positionsbestimmung. Dabei kann zwischen *enableHighAccuracy* (Anweisung an die Anwendung, die besten möglichen Positionsergebnisse zu liefern. Diese Einstellung benötigt mehr Zeit und kostet mehr Energie), *timeout* (maximale Dauer der Anfrage) und *maximumAge* (maximal gültiges Alter einer gespeicherten Position) unterschieden werden.

Auch das *watchPosition* Objekt kann ein bis drei Argumente enthalten, die äquivalent zu den zuvor vorgestellten Argumenten bei *getCurrentPosition* sind (*PositionCallback*, *PositionErrorCallback*, *PositionOptions*).

Die Antwort der Geolocation API wird in einem Positionsinterface (*Position*), das in Listing 3.16 gezeigt wird, zurückgegeben, das die Positionskordinaten *Coordinates* und die Zeit der Positionsbestimmung *DOMTimeStamp* beinhaltet.

**Listing 3.16** – *Position* Objekt

```
[NoInterfaceObject]
interface Position {
  readonly attribute Coordinates coords;
  readonly attribute DOMTimeStamp timestamp;
};
```

Das Koordinateninterface (*Coordinates*) in Listing 3.17 liefert neben den Koordinaten der geographischen Länge (*longitude*) und Breite (*latitude*) auch die GPS-Höheninformation (*altitude*), Genauigkeitsinformationen (*accuracy*, *altitudeAccuracy*) sowie bei *watchPosition* auch die Richtung (*heading*) und die Geschwindigkeit (*speed*). Die optionalen Attribute des Koordinateninterface sind mit einem „?“ gekennzeichnet.

**Listing 3.17** – Koordinateninterface *Coordinate*

```
[NoInterfaceObject]
interface Coordinates {
  readonly attribute double latitude;
  readonly attribute double longitude;
  readonly attribute double? altitude;
  readonly attribute double accuracy;
  readonly attribute double? altitudeAccuracy;
  readonly attribute double? heading;
  readonly attribute double? speed;
};
```

Das *PositionError* Objekt (Listing 3.18) beinhaltet im Fall einer fehlgeschlagene Positionsbestimmung deren Grund. Sowohl ein Integerwert (*PERMISSION\_DENIED* = 1, *POSITION\_UNAVAILABLE* = 2 oder *TIMEOUT* = 3) als auch eine Textnachricht wird an das Programm zurückgegeben.

**Listing 3.18** – *PositionError* Objekt

```
[NoInterfaceObject]
interface PositionError {
```

```
const unsigned short PERMISSION_DENIED = 1;
const unsigned short POSITION_UNAVAILABLE = 2;
const unsigned short TIMEOUT = 3;
readonly attribute unsigned short code;
readonly attribute DOMString message;
};
```

Nachdem die API keine Rückschlüsse auf ihre zugrunde liegenden Standortinformationsquellen liefert, gibt sie auch keine Garantie, dass der richtige Standort des Gerätes ausgegeben wird (W3C-Geolocation-Spezifikation (Zuletzt besucht: 03.Juli 2011)).

Die W3C Geolocation API wurde Anfang des Jahres 2010 als HTML5-Spezifikation übernommen. Außerdem wurde sie auch schon in weitere Applikationen eingebaut, so ist sie seit Anfang 2011 auch in OpenLayers umgesetzt.

### 3.6.5 Vergleich zwischen IP Geolocation und W3C Geolocation API

Bei näherer Betrachtung der verschiedenen Anbieter von IP-Geolocationdiensten fällt auf, dass die Dienste für den Rechner mit der IP-Adresse 129.27.89.45 am Institut für Geoinformation, Steyrergasse 30 in Graz unterschiedliche Positionslösungen liefern. Die Abbildung 3.6 zeigt einen Vergleich der Positionen<sup>14</sup> der Dienste Quova.com, Maxmind.com und IP2Location.com mit der W3C Geolocation API.

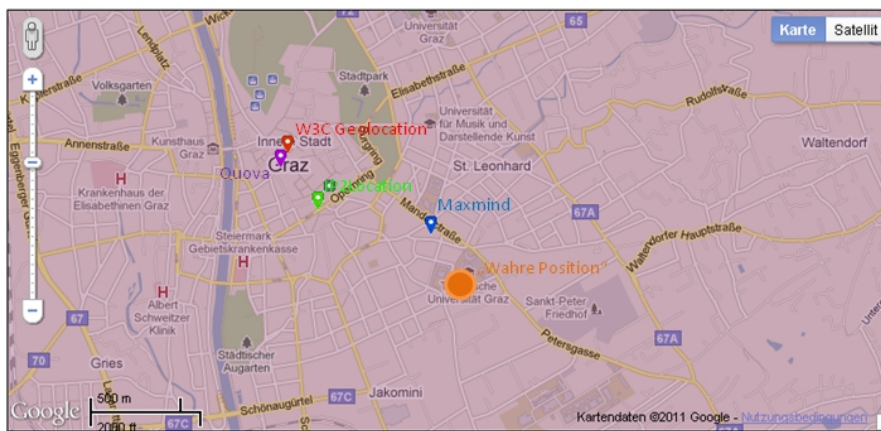
Ein großes Problem dieser Dienste, bzw. auch der W3C Geolocation API wurde auch in Abschnitt 2.3.1 bereits angesprochen. So bezieht sich die von den hier betrachteten Diensten erhaltene Position meist nicht auf den jeweiligen Rechner, sondern vielmehr auf den zentralen Server eines Dienstleisters z.B. dem entsprechenden Internetanbieter.

Somit werden die Ergebnisse der W3C Geolocation API vor allem bei Geräten ohne GPS-Empfänger bzw. aktiviertem WLAN sehr oft verfälscht, da Internetnutzer über zentrale Server ins Netz einsteigen (z.B. mobiles Internet über Mobilfunknetz).

---

<sup>14</sup><http://whatismyipaddress.com/update-location>

Geolocation Map for 129.27.89.45



Geolocation Details for 129.27.89.45

	📍 W3C Geolocation	📍 Provider A	📍 Provider B	📍 Provider C
Latitude	47.0707	47.07	47.0667	47.0679
Longitude	15.4395	15.439	15.45	15.4417
Accuracy (m)	18000			
Country		austria	austria	austria
State/Region		steiermark	steiermark	steiermark
City		graz	graz	graz
Postal Code		8010		-
Organization		technische universitaet graz	technische universitaet graz	technische universitaet graz
ISP		technische universitaet graz, at	technische universitaet graz	
Connection		high (b)	unknown	comp
Accuracy	Select Accuracy ▾	Select Accuracy ▾	Select Accuracy ▾	Select Accuracy ▾
Feedback				

Abbildung 3.6 – Darstellung des Positionsergebnisses verschiedener Geolocationdienste

## Kapitel 4

# Der „Weather Twitter“

Nachdem in den vorangegangenen Kapiteln die Grundlagen zum Aufbau einer WebGIS-Applikation vorgestellt wurden, wird in diesem Kapitel auf die eigentliche Applikation, den „Weather Twitter“ eingegangen. Der „Weather Twitter“ erlaubt es dem Nutzer seine aktuelle Wettersituation mitzuteilen. Je mehr Menschen ihr Wetter dem System mitteilen, desto detailliertere Informationen können vom System generiert werden.

### 4.1 Grundgedanken bzw. Konzept des „Weather Twitter“

Gerade das aktuelle Wetter, sowie die Wettersituation in naher Zukunft, sind für die meisten Menschen von großer Bedeutung. Diese Information liegt jedoch meist nur sehr großflächig vor, da sich ein Netz mit Wetterstationen nur in einer bestimmten Dichte realisieren lässt. Warum sollte man daher nicht den Menschen selbst als Sensor benutzen, um Information zu generieren, und damit ein System zu erstellen, dass wiederum bessere Information liefert.

### 4.2 Aufbau des Webdienstes

Der „Weather Twitter“ ist als Webseite mit angebundener GDI realisiert, die auf einem Webserver abgelegt ist, und mit einem Internetbrowser aufgerufen werden kann. Dafür wurden die in Kapitel 2 vorgestellten Technologien verwendet.

#### 4.2.1 Startseite

Die Startseite des Weather Twitter (Abb. 4.1) ermöglicht dem Besucher einen Überblick über die aktuelle Wetterlage der Steiermark.

Zentrales Element der Seite ist das Kartenfenster, in dem die aktuelle Wetterinformation für die Steiermark dargestellt ist. Die Darstellung der Daten erfolgt mithilfe der Open Source Anwendung OpenLayers, wobei als Hintergrundkarte eine von OSM gerenderte Karte verwendet wird. Angezeigt werden dabei acht, nach regionalen Kriterien ausgewählte, Orte und die dort vorherrschende Temperatur, samt Wetterlage.

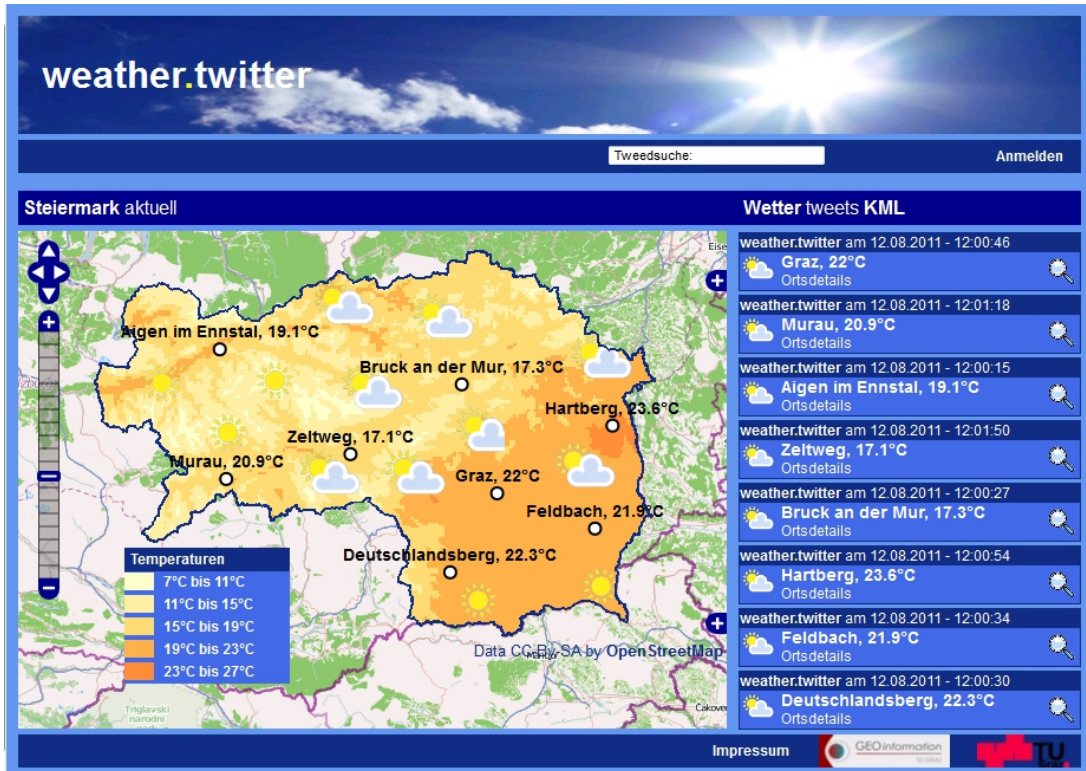


Abbildung 4.1 – Startseite von „Weather Twitter“

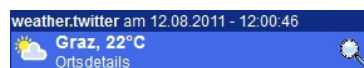


Abbildung 4.2 – Tweet für Graz

Diese Orte werden auch in der Tweetliste am rechten Bildrand gezeigt. Ein Tweet (siehe Abbildung 4.2) umfasst mehrere Informationen zum entsprechenden Ort, wobei mit einem Klick auf die Lupe auch zum entsprechenden Ort gezoomt werden kann. Der Tweet kann auch über ein „Mouseover“-Event in der Karte aufgerufen werden. Der Link „Ortsdetails“ verweist auf eine Seite, wo zusätzliche Informationen (Wettervorschau, Temperaturentwicklung, Kartenausschnitt etc.) zum Ort angezeigt werden.

#### 4.2.2 Home

Die Seite „Home“ (Abb. 4.3) zeigt in der Karte alle Wettermeldungen an, die innerhalb der letzten Stunde eingetragen wurden. Die angezeigten „Tweets“, in der Tweetspalte rechts neben der Karte, richten sich dabei nach der Bounding Box des Kartenausschnittfensters.

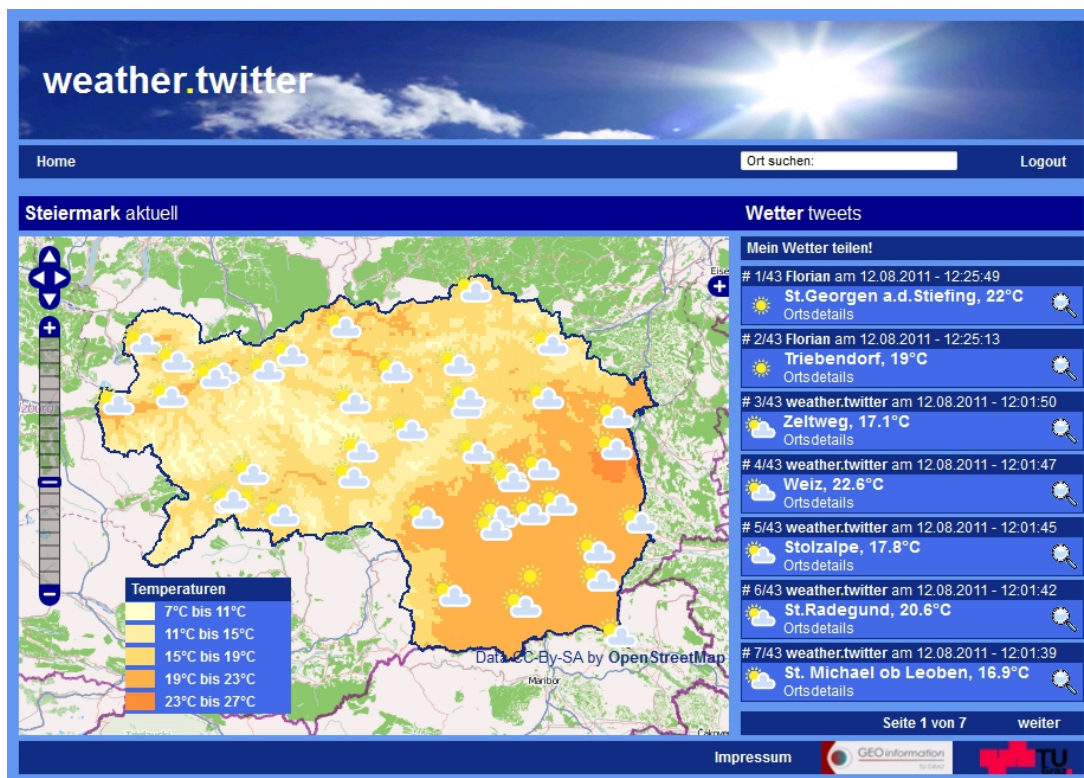


Abbildung 4.3 – Übersichtsseite „Home“ des „Weather Twitter“

Neben den Funktionen, die bereits in der Startseite ausgeführt werden können (Lupe, „Mouseover“-Event), kann der Benutzer, mit einem Klick auf den Button „Mein Wetter teilen!“, sein aktuelles Wetter an das System übermitteln.

#### 4.2.3 Wetter mitteilen

Diese Seite (Abbildung 4.4) dient dem Benutzer dazu, sein aktuelles Wetter an das System zu übermitteln.





Abbildung 4.4 – Seite um dem System sein Wetter mitzuteilen

Zuerst wählt der User die entsprechende Wettersituation (Iconauswahl). Danach muss sich der Benutzer positionieren. Die Positionierung kann sowohl mit einem Klick auf den Button Geolocation, als auch mit einem Klick in die Karte erfolgen. Anhand der bestimmten Position ermittelt „Weather Twitter“ aus dem Wetterraster eine Näherungstemperatur. Diese Lufttemperatur kann anschließend mithilfe eines Schiebereglers um  $\pm 5^{\circ}\text{C}$  verändert werden. Nachdem die Wettermeldung in einer Vorschau angezeigt wird, kann über den Button „Wetter mitteilen!“ das aktuelle Wetter an das System gesendet werden.

#### 4.2.4 Wetterdetails

Die Seite „Wetterdetails“ zeigt die Wettervorhersage (Abb. 4.5) für einen Ort in den nächsten Tagen. Außerdem kann die Temperaturentwicklung, sowie ein Kartenausschnitt mit der Position des Ortes angezeigt werden.

Die Informationen für die Wettervorhersage stammen vom Norwegischen Meteorologischen Institut<sup>1</sup>, wobei neben der vorhergesagten Wetterlage, die minimale und maximale Temperatur, die Windentwicklung, sowie der Tagesniederschlag angezeigt werden.

<sup>1</sup><http://www.yr.no>



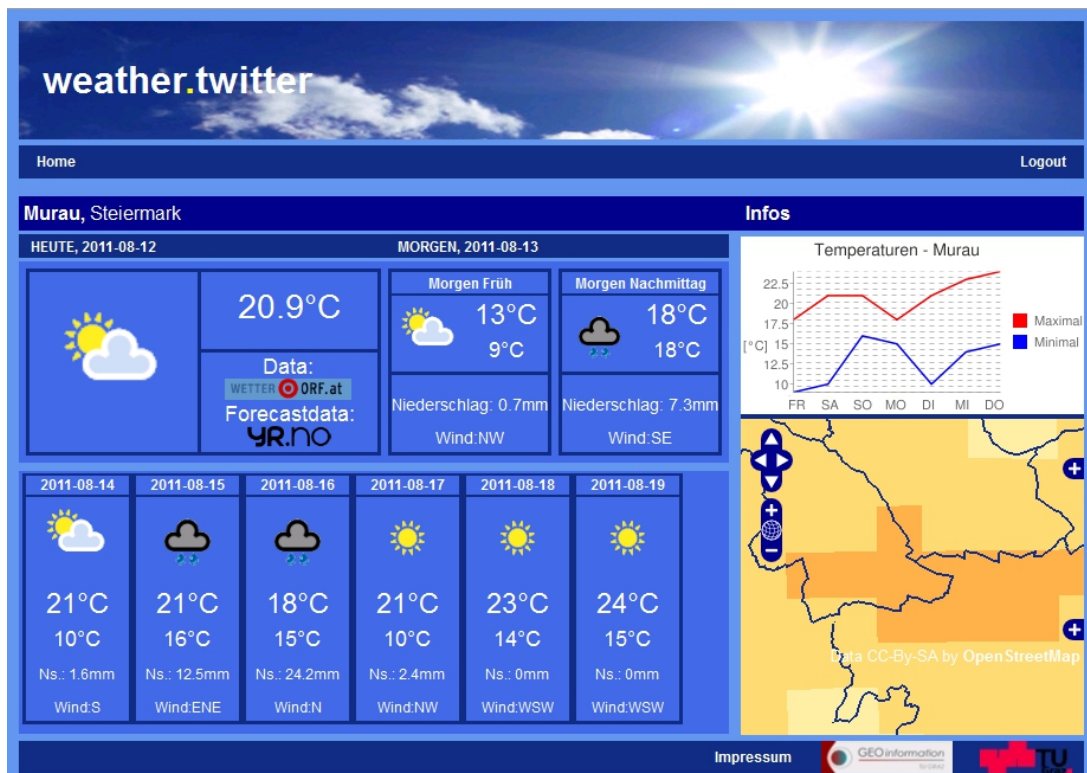


Abbildung 4.5 – Detailsseite von „Weather Twitter“

### 4.3 Datenbankinfrastruktur mittels PostgreSQL

Die dem Wetterdienst zugrunde liegende Datenbankinfrastruktur wird mittels einer PostgreSQL-Datenbank realisiert. In Abbildung 4.6 wird die Datenbankinfrastruktur als Entity Relationship Modell (ERM) dargestellt. Zum Aufbau der Applikation werden mehrere Tabellen benötigt:

- Tabelle für Informationen zu den einzelnen Wettermeldungen (*wt\_user\_tweet*).
- Tabelle für die Wettervorschauinformation (*wt\_user\_forecast*).
- Tabelle für den Wetterraster (*wt\_interpolation\_utm*).

#### 4.3.1 Wettertweets

Die einzelnen Wettertweets sind das Herzstück der Applikation. Ohne Wettermeldungen, die in die Anwendung einfließen, kann der Dienst nicht funktionieren. Dabei werden sowohl die Grundlagenmeldungen, die von der ORF-Wetterseite bzw. von der ZAMG stammen, als auch die vom Nutzer mitgeteilten Wettermeldungen (Tweets) in der Tabelle *wt\_user\_tweet* gespeichert. Die Tabelle besteht aus einer Reihe an Datenspalten (Tweet-ID, User-ID, Username, Zeit des Eintrags, Ortsname, Wettersituation, Temperatur, Höhe des Ortes, Vorschaudaten), sowie einer Geometriespalte zur Verspeicherung der Position der Mitteilung.

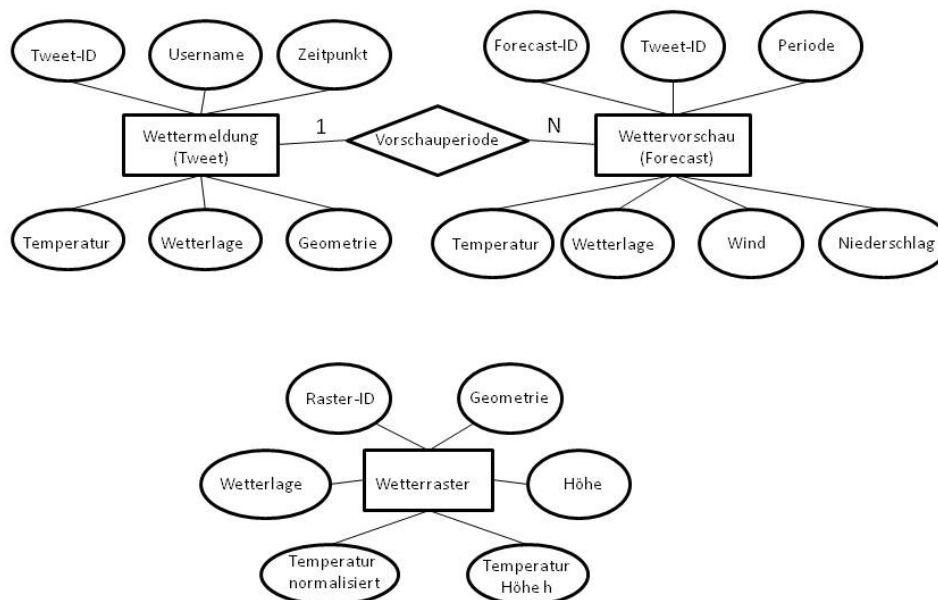


Abbildung 4.6 – Entity Relationship Modell des "Weather Twitter"

### 4.3.2 Wettervorschauinformation

Neben dem aktuellen Wetter bietet die Applikation auch die Möglichkeit, für bestimmte Orte Wettervorschauinformation abzurufen. Diese Information stammt vom norwegischen Dienst yr.no und wird in der Tabelle *wt\_user\_forecast* gespeichert. Die Tabelle besteht aus der Wettervorhersage für die nächsten neun Tage (Forecast-ID, Tweet-ID, Zeitperiode, Temperatur, Wetterlage, Wind, Niederschlag, Druck).

### 4.3.3 Temperaturreaster

Um die Temperaturdaten auch flächenhaft darstellen zu können, werden die Temperaturen aus den Wettermeldungen in einen 1000×1000 Meter Raster interpoliert. Die interpolierten Daten werden in der Tabelle *wt\_interpolation\_utm* abgespeichert. Diese Tabelle beinhaltet Datenspalten zur jeweiligen Rasterzelle (ID, Latitude, Longitude, Temperatur, Höhe, Wettersituation), sowie zwei Geometriespalten (Punktgeometrie und Polygoneometrie).

## 4.4 Einbinden von Mapfiles in den Webdienst

Die zur Darstellung im „Weather Twitter“ bestimmten Daten, werden mithilfe von Mapfiles aufbereitet. Mapfiles sind das zentrale Element des MapServer. In ihnen werden die Beziehungen

zwischen Objekten, sowie die einzubindenden Daten und verschiedene Arten der Darstellung definiert. Der Aufbau eines Mapfiles ist in Abschnitt 3.4.1 näher beschrieben.

Anschließend werden die Mapfiles in ein entsprechendes OpenLayers-Objekt als WMS eingebunden (siehe Abschnitt 3.5.2).

## 4.5 Einbindung von OpenLayers in den Webdienst

Die Darstellung der Wetterinformation im Webbrowser erfolgt mit der JavaScript-Bibliothek OpenLayers. Die Einbindung von OpenLayers in HTML erfolgt über einen `<div>`-Tag, der als Platzhalter für das in OpenLayers erzeugte Mapobjekt dient. Alle Objekte, wie diverse Controls oder Layers werden dem Mapobjekt hinzugefügt und damit gemeinsam mit dem Mapobjekt in den Webdienst übernommen.

## 4.6 Positionsbestimmung im „Weather Twitter“

Die Positionsbestimmung im „Weather Twitter“ erfolgt mithilfe der in Abschnitt 3.6 vorgestellten W3C Geolocation API. Je nachdem welches Endgerät der Benutzer verwendet, um auf die Applikation zuzugreifen, werden zur Positionsbestimmung verschiedene Positionsquellen herangezogen. Dadurch sind auch im Ergebnis der Positionsermittlung, analog zu den Vor- und Nachteilen der verschiedenen Positionierungsmethoden (Abschnitt 3.6.2) einige Unterschiede sichtbar.

### 4.6.1 Positionierung am Desktop PC

Am Desktop PC bzw. Notebook erfolgt die Positionierung hauptsächlich über die IP-Adresse bzw. wenn möglich über vorhandene WLAN-Zugangspunkte. Dadurch stimmt die ermittelte Position des Nutzers meist nur sehr schlecht, bzw. überhaupt nicht mit der tatsächlichen Position des Computers und damit Nutzers überein. So kann im schlechtesten Fall nicht einmal das richtige Bundesland ermittelt werden. Nur in Städten, in denen es bereits gute Aufzeichnungen der WLAN-Zugangsknoten (Abbildung 3.5) gibt (z.B. Graz, Wien), sind zufriedenstellende Ergebnisse zu erwarten. Ganz im Gegensatz dazu können am Handy bzw. Smartphone sehr gute Positionslösungen erzielt werden.

### 4.6.2 Positionierung am Handy bzw. Smartphone

Hier bedient sich die W3C Geolocation API meist an Informationen des Mobilfunknetzes (Handymasten), oder an den vom Gerät überlieferten GPS-Daten. Diese Datenquellen liefern für die Eingabe von Wetterdaten mehr als ausreichende Genauigkeiten, die bis in den Meterbereich reichen. Bei der Nutzung der W3C Geolocation API am Smartphone wurde festgestellt, dass die erste Positionslösung nie mittels der GPS-Daten ermittelt wird, sondern die API immer eine Näherungslösung über das Mobilfunknetz bestimmt. Erst bei einer weiteren Positionser-

mittlung (erneutes *getCurrentPosition* oder *watchPosition* (siehe Abschnitt 3.6.4)) wird auf die GPS-Daten zugegriffen.

## Kapitel 5

# Anwendung und Einsatz des „Weather Twitter“

In diesem Kapitel werden einige Anwendungsgebiete des in Kapitel 4 vorgestellten „Weather Twitter“ beleuchtet. Dabei wird neben der ursprünglichen WebGIS-Applikation im Besonderen auf eine im Rahmen der Lehrveranstaltung „AK Medientechnologien“ am Institut für Informationssysteme und Computer Medien an der TU Graz erstellte iPhone bzw. Android Applikation eingegangen, der die Daten des „Weather Twitter“ zugrunde liegen.

### 5.1 Anwendung der Applikation

Der „Weather Twitter“ steht als WebGIS-Applikation am Server des Institutes für Geoinformation zur Verfügung. Dabei können die in Kapitel 4 vorgestellten Seiten besucht, und die entsprechenden Wettermeldungen eingegeben werden. Neben diesen Webseiten dient der „Weather Twitter“ jedoch auch als Grundlage für eine weitere Anwendung.

### 5.2 Interoperabilität mit der mobilen Applikation „wettR“

Im Rahmen der Lehrveranstaltung „AK Medientechnologien“ wurde von den Studierenden Stephan Moser, Horst Possegger, Florian Salbrechter und Raphael Spreitzer der TU Graz die iPhone und Android Applikation „wettR“ realisiert. Die Applikation baut auf der Idee und dem Konzept von „Weather Twitter“, das in den vorigen Kapiteln vorgestellt wurde, auf. Die mobile Anwendung erlaubt es dem Benutzer, sein aktuelles Wetter mitzuteilen und die Wettermeldung in den „Weather Twitter“ einfließen zu lassen. Außerdem können die aktuellen Wettermeldungen des „Weather Twitter“ von „wettR“ abgerufen und dargestellt werden.

### 5.2.1 Definition der Schnittstellen zwischen „Weather Twitter“ und der mobilen Applikation

Der Datenaustausch zwischen dem „Weather Twitter“ und der mobilen Anwendung erfolgt mittels KML bzw. XML. Dafür werden die aktuellen Wettermeldungen im KML-Format ausgegeben, womit sie sich in Google Earth darstellen lassen. Die entsprechenden Anfragen an den Dienst werden mittels HTTP-Requests und PHP realisiert.

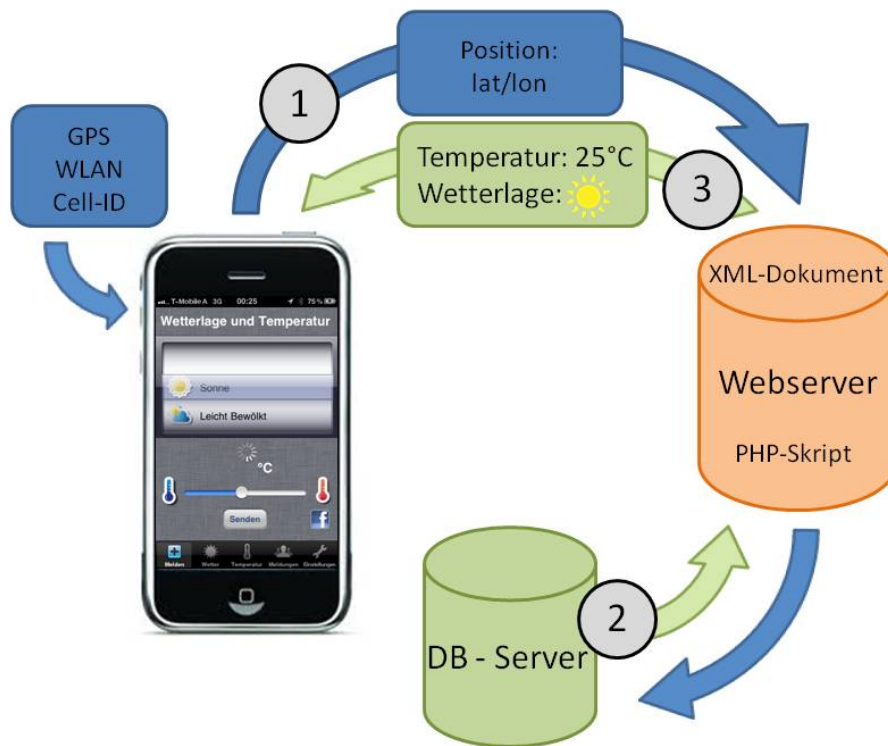
**Listing 5.1** – XML-Anfrage der mobilen Applikation an den „Weather Twitter“

```

<weatherdata>
- <request>
  <position lat="15" lon="47" />
</request>
- <current_condition_next_tweet>
  <time>12.08.2011 - 12:00:25</time>
  <time_kml>2011-08-12T12:00:25Z</time_kml>
  <tweet_by_user>Weather Twitter</tweet_by_user>
  <tweet_location>Bad Radkersburg</tweet_location>
  <tweet_lon>15.988611</tweet_lon>
  <tweet_lat>46.689722</tweet_lat>
  <tweet_altitude height="211" unit="meters" />
  <tweet_temperature temp="20.9" unit="degree Celsius" />
  <tweet_icon_url>
    .../wsymbols01_png_64/wsymbol_0002_sunny_intervals.png
  </tweet_icon_url>
  <distance_to_next_tweet distance="4785.145" unit="kilometers" />
</current_condition_next_tweet>
- <current_condition_raster>
  <time>12.08.2011 - 12:00:25</time>
  <raster_location>Ash Shiqq</raster_location>
  <raster_lon>16.032612</raster_lon>
  <raster_lat>46.660899</raster_lat>
  <raster_altitude height="208" unit="meters" />
  <raster_temperature temp="24" unit="degree Celsius" />
  <raster_icon_url>
    .../wsymbols01_png_64/wsymbol_0002_sunny_intervals.png
  </raster_icon_url>
  <distance_to_next_rasterpoint distance="4780499" unit="meters" />
</current_condition_raster>
</weatherdata>

```

Die mobile Applikation nimmt mit dem „Weather Twitter“ Kontakt auf und übermittelt seine Position in Form von zwei GET Parametern *lat* und *lon* an ein PHP-Skript ((1) in Abb. 5.1). Eine Anfrage für die Position  $lat=47^\circ$  und  $lon=15^\circ$  ([http://129.27.89.66/student/master\\_weather/Weather\\_Twitter/XML/iphone\\_communication.php?lon=47&lat=15](http://129.27.89.66/student/master_weather/Weather_Twitter/XML/iphone_communication.php?lon=47&lat=15)) liefert das Ergebnis in Listing 5.1.



**Abbildung 5.1** – Wettervorschlag für die aktuelle Position

In diesem Skript werden Wetterdetails (Lufttemperatur, Wettersituation) zur übermittelten Position vom Datenbankserver abgefragt (2) und als XML-Dokument an die Applikation „wettR“ zurückgesendet (3).

In weiterer Folge gibt der User der mobilen Applikation die aktuelle Wetterlage ein ((4) in Abb. 5.2), wobei nach Bestätigung der Eingabe die Parameter für die Position (*lat*, *lon*) sowie die Lufttemperatur (*temp*) und die Wettersituation (*icon*) mithilfe des folgenden Requests (`http://129.27.89.66/student/master_witter/Weather_Twitter/XML/iphone_communication.php?lon=47&lat=15&temp=20&icon=01`) an ein PHP-Skript übermittelt (5), und in der Datenbank gespeichert werden (6).

### 5.2.2 Darstellung der Daten in der mobilen Applikation

Die zur Darstellung des Wetters in der mobilen Applikation benötigten Daten liegen in Form von KML-Dokumenten am Webserver des Institutes für Geoinformation vor. So können, wie Schritt (7) in Abbildung 5.3 zeigt, die aktuelle Temperaturverteilung über die Steiermark (Heatmap) bzw. die einzelnen Wettertweets vom Server abgerufen werden, wobei die Anzahl der dargestellten Meldungen variabel festgelegt werden kann.

Außerdem liegt eine KML-Datei vor, die das Wetter zu 100 zufällig gewählten Standpunkten in der Steiermark beinhaltet, und ebenfalls zur Darstellung dient. Diese KML-Files werden über Google Earth in den Dienst „wettR“ eingebunden. Zur besseren Visualisierung können diverse

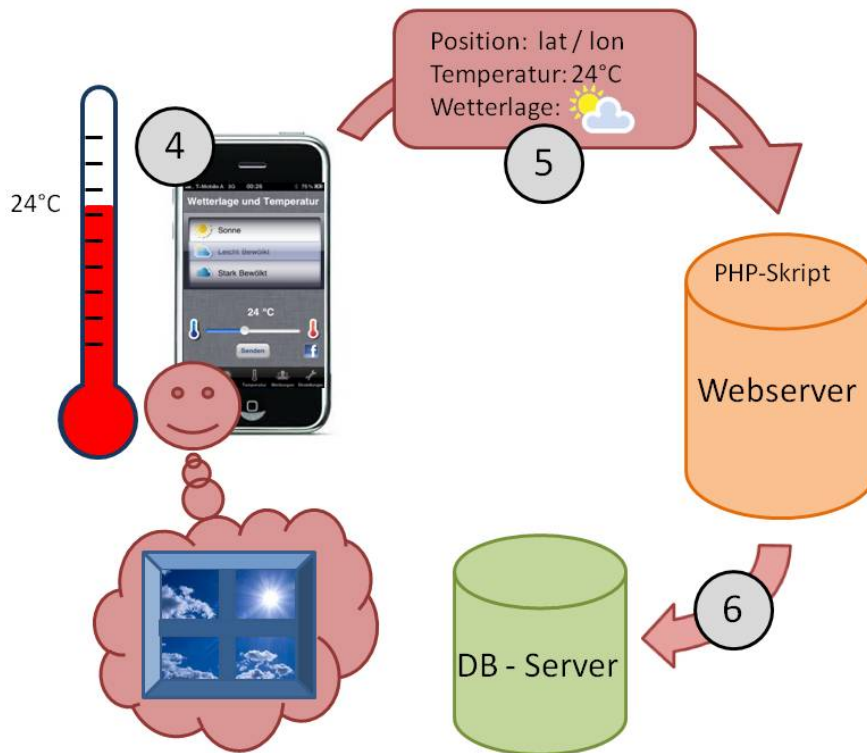


Abbildung 5.2 – Übertragung des vorherrschenden Wetters

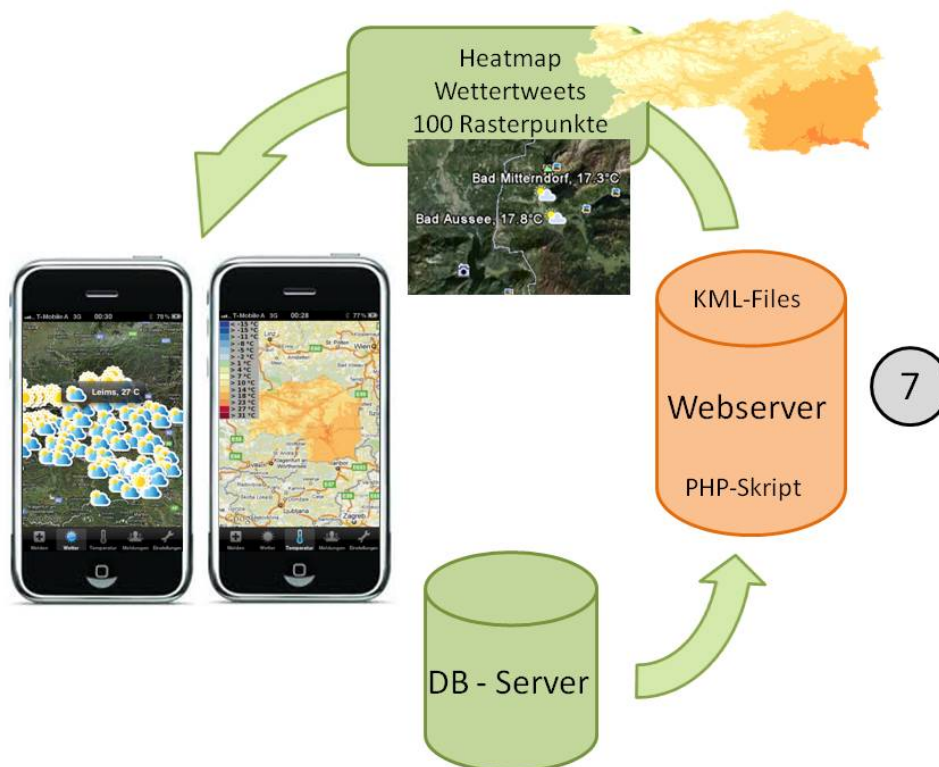


Abbildung 5.3 – Darstellungsmöglichkeiten von „wettR“



Parameter, wie die Transparenz des Rasters, vom Nutzer verändert werden.

Die KML-Dateien mit der Temperaturverteilung der Steiermark, sowie den zufällig verteilten 100 Orten, werden bei jeder durchgeführten Interpolation neu erzeugt. Die KML-Datei, welche die einzelnen Wettermeldungen beinhaltet, wird nach jedem Systemwetterupdate erzeugt.



## Kapitel 6

# Zusammenfassung und Ausblick

Im Schlusskapitel dieser Masterarbeit wird der „Weather Twitter“ noch einmal zusammenfassend betrachtet. Desweiteren wird auf mögliche Ausbauszenarien und Änderungen des Systems eingegangen, da das Konzept des „Weather Twitter“ sicher noch sehr viel Spielraum für Erweiterungen offen lässt.

### 6.1 Ausbau der Applikation

Die Applikation, die im Rahmen der Masterarbeit „WITTER - Weather Twitter“ entstanden ist, bietet dem Nutzer eine lokale Wetterinformation für die Steiermark. Dabei ist die lokale Dichte und Genauigkeit der Information nur von der Anzahl der Nutzer und der Qualität ihrer Beiträge abhängig. Folgende Wetterinformationen können vom Benutzer eingegeben und abgefragt werden:

- Lufttemperatur.
- Aktuelle Wettersituation (Sonne, Regen, bewölkt etc.).

Natürlich lässt diese Tatsache großen Spielraum für mögliche Erweiterungen. Darunter fällt die Möglichkeit, zusätzliche Wetterparameter (Luftdruck, Wind, Niederschlag etc.) in das System einfließen zu lassen. Desweiteren könnte man noch andere Parameter (Pollenbelastung, Feinstaub etc.) in den Onlinedienst integrieren. Da die Aufbereitung und Modellierung solcher Daten jedoch nicht unbedingt in die Kernkompetenz eines Geoinformatikers fällt, wurden diese Parameter nicht berücksichtigt. Dafür müsste wohl meteorologisches Expertenwissen und die Kenntnis diverser Modelle zur Verarbeitung solcher Parameter hinzugezogen werden.

Die große Spannung des „Weather Twitter“ und eines der Hauptargumente für die Umsetzung einer solchen Applikation liegt jedoch nicht direkt in der Modellierung von meteorologischen Daten, sondern in der Umsetzung und Untersuchung der W3C Geolocation API, sowie der Nutzung von sogenanntem „User Generated Content“.

Die W3C Geolocation API lässt sicher noch genug Möglichkeiten für diverse Verbesserungen offen, so wäre es durchaus hilfreich, Daten über die verwendeten Positionierungsmethoden

bzw. den Berechnungsalgorithmus der Position offenzulegen, um die Positionsgenauigkeit besser einschätzen zu können. Derzeit geht mit der Benutzung der API ein großer Unsicherheitsfaktor einher, da ohne weiteres auch falsche Positionslösungen geliefert werden. Somit sind die Kontrolle bzw. manuelle Eingabe der aktuellen Position für viele Applikationen unumgänglich.

Der zweite wichtige Punkt für zukünftige Arbeiten liegt in der Untersuchung des „User-generated Content“. Besonders der Umgang mit solchen Daten in Bezug auf deren Zuverlässigkeit, und damit verknüpft deren Weiterverarbeitung und Filterung, lässt einiges an Untersuchungsspielraum offen. Spannende Fragen, die in naher Zukunft mit großer Wahrscheinlichkeit eine wichtige Rolle spielen werden.

## **6.2 Inhalte und Design**

Das Design der WebGIS-Applikation und die nutzerfreundliche Aufbereitung der Inhalte können ebenso in noch professionellerer Form umgesetzt werden. Außerdem wäre eine Auswahl von interessanten Zusatzinformationen und Statistiken zu treffen, die der Anwendung zusätzlich eine gewisse Attraktivität verleihen. Möglich wären hier Benutzeranmeldungen und auf den jeweiligen Benutzer abgestimmte Inhalte, wie Lieblingsorte, individuelle Wetterberichte, Ableitung des Wetterverhaltens aus den Usermeldungen und ein Vertrauensindex für den User und seine Meldungen.

## **6.3 Kommerzielle Nutzung**

Eine kommerzielle Nutzung des Systems wäre in weiterer Folge sinnvoll, um das Angebot des „Weather Twitter“ auszubauen. Dafür sind vor allem die rechtlichen Fragen bei der Nutzung von Wetterdaten verschiedener Dienste zu klären.

# Literaturverzeichnis

- [Presentation-Slides-Ryan-Sarver Zuletzt besucht: 03.Juli 2011] : *W3C Geolocation API - Making Websites Location-aware*. Zuletzt besucht: 03.Juli 2011. – URL <http://www.slideshare.net/rsarver/w3c-geolocation-api-making-websites-locationaware>
- [W3C-Geolocation-Spezifikation Zuletzt besucht: 03.Juli 2011] : *W3C Geolocation API - Spezifikation*. Zuletzt besucht: 03.Juli 2011. – URL <http://www.w3.org/TR/2010/CR-geolocation-API-20100907/>
- [ZAMG Zuletzt besucht: 03.Juli 2011] : *Zentralanstalt für Meteorologie und Geodynamik*. Zuletzt besucht: 03.Juli 2011. – URL [http://www.zamg.ac.at/wir\\_ueber\\_uns](http://www.zamg.ac.at/wir_ueber_uns)
- [Geolocation-API-Network-Protocol Zuletzt besucht: 07.Juli 2011] : *Geolocation API Network Protocol*. Zuletzt besucht: 07.Juli 2011. – URL [http://code.google.com/intl/de-DE/apis/gears/geolocation\\_network\\_protocol.html](http://code.google.com/intl/de-DE/apis/gears/geolocation_network_protocol.html)
- [Klimedia Zuletzt besucht: 07.Juli 2011] : *Klimedia - Barometrische Höhenformel*. Zuletzt besucht: 07.Juli 2011. – URL <http://www.klimedia.ch/kap3/a6.html>
- [OpenLayers-Bibliothek Zuletzt besucht: 07.Juli 2011] : *OpenLayers - Javascript Mapping Library*. Zuletzt besucht: 07.Juli 2011. – URL <http://dev.OpenLayers.org/apidocs/files/OpenLayers-js.html>
- [OpenLayers-Controls Zuletzt besucht: 07.Juli 2011] : *OpenLayers.Controls - Javascript Mapping Library*. Zuletzt besucht: 07.Juli 2011. – URL <http://dev.OpenLayers.org/apidocs/files/OpenLayers/Control-js.html>
- [Wunderground Zuletzt besucht: 07.Juli 2011] : *Wunderground - About Our Data*. Zuletzt besucht: 07.Juli 2011. – URL <http://www.wunderground.com/about/data.asp>
- [GeoServer-Dokumentation Zuletzt besucht: 09.Mai 2011] : *Geoserver*. Zuletzt besucht: 09.Mai 2011. – URL <http://docs.geoserver.org/>
- [<http://mapserver.org/about.html> Zuletzt besucht: 09.Mai 2011] : *Mapserver*. Zuletzt besucht: 09.Mai 2011. – URL <http://mapserver.org/about.html>
- [PHP-Manual Zuletzt besucht: 09.Mai 2011] : *PHP Manual*. Zuletzt besucht: 09.Mai 2011. – URL <http://www.php.net/manual/de/>

- [W3C-HTML-CSS Zuletzt besucht: 09.Mai 2011] : *W3C HTML und CSS*. Zuletzt besucht: 09.Mai 2011. – URL <http://www.w3.org/standards/webdesign/htmlcss>
- [W3C-Mission Zuletzt besucht: 09.Mai 2011] : *W3C Mission*. Zuletzt besucht: 09.Mai 2011. – URL <http://www.w3.org/Consortium/mission>
- [W3C-XML Zuletzt besucht: 09.Mai 2011] : *W3C Mission*. Zuletzt besucht: 09.Mai 2011. – URL <http://www.w3.org/standards/xml/core>
- [OGC-About Zuletzt besucht: 13.April 2011] : *About OGC*. Zuletzt besucht: 13.April 2011. – URL <http://www.opengeospatial.org/ogc/>
- [Wikipedia-IP-Positionierung Zuletzt besucht: 15.Mai 2011] : *Wikipedia IP Positionierung*. Zuletzt besucht: 15.Mai 2011. – URL [http://code.google.com/intl/de-DE/apis/gears/geolocation\\_network\\_protocol.html](http://code.google.com/intl/de-DE/apis/gears/geolocation_network_protocol.html)
- [Räumliche-Interpolation Zuletzt besucht: 20.August 2011] : *Modellierung von Werteoberflächen - Räumliche Interpolation*. Zuletzt besucht: 20.August 2011. – URL [http://ifgivor.uni-muenster.de/vorlesungen/Geoinformatik/kap/kap7/k07\\_4.htm](http://ifgivor.uni-muenster.de/vorlesungen/Geoinformatik/kap/kap7/k07_4.htm)
- [Macrec Zuletzt besucht: 23.Juni 2011] : *Macrec - Bildformate: kleines Tutorial über GIF, JPEG, PNG, TIFF, EPS und PDF*. Zuletzt besucht: 23.Juni 2011. – URL <http://www.macrec.ch/pdf/bildformate.pdf>
- [WebScanNotes Zuletzt besucht: 23.Juni 2011] : *WebScanNotes - W3C Geolocation API*. Zuletzt besucht: 23.Juni 2011. – URL [http://webscannotes.com/?page\\_id=425](http://webscannotes.com/?page_id=425)
- [<http://code.google.com/intl/de-DE/apis/gears> Zuletzt besucht: 27.April 2011] : *Geolocation API*. Zuletzt besucht: 27.April 2011. – URL [http://code.google.com/intl/de-DE/apis/gears/api\\_geolocation.html](http://code.google.com/intl/de-DE/apis/gears/api_geolocation.html)
- [ISO Zuletzt besucht: 27.April 2011] : *ISO 19133:2005, Geographic information – Location-based services – Tracking and navigation*. Zuletzt besucht: 27.April 2011. – URL [http://www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=32551](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=32551)
- [GISWIKI-LBS Zuletzt besucht: 27.April 2011] : *Location Based Service*. Zuletzt besucht: 27.April 2011. – URL <http://www.giswiki.org/wiki/LBS>
- [OGC-Spezifikation-KML Zuletzt besucht: 27.April 2011] : *OGC Spezifikation - KML*. Zuletzt besucht: 27.April 2011. – URL <http://www.opengeospatial.org/standards/kml>
- [OGC-Spezifikation-SFS Zuletzt besucht: 27.April 2011] : *OGC Spezifikation - Simple Features*. Zuletzt besucht: 27.April 2011. – URL <http://www.opengeospatial.org/standards/sfs>
- [OGC-Spezifikation-WFS Zuletzt besucht: 27.April 2011] : *OGC Spezifikation - WFS*. Zuletzt besucht: 27.April 2011. – URL <http://www.opengeospatial.org/standards/wfs>

- [OGC-Spezifikation-WMS Zuletzt besucht: 27.April 2011] : *OGC Spezifikation - WMS*. Zuletzt besucht: 27.April 2011. – URL <http://www.opengeospatial.org/standards/wms>
- [OpenGIS-Location-Service Zuletzt besucht: 27.April 2011] : *OpenGIS Location Service (Open-LS) Implementation Specification: Core Services Version 1.2.0*. Zuletzt besucht: 27.April 2011. – URL <http://www.opengeospatial.org/standards/ols>
- [Bartelme 2005] BARTELME, Norbert: *Geoinformatik, Modelle-Strukturen-Funktionen*. Springer Verlag, 2005
- [Blankenbach 2007] BLANKENBACH, Jörg: *Handbuch der mobilen Geoinformation, Architektur und Umsetzung mobiler standortbezogener Anwendungen und Dienste unter Berücksichtigung von Interoperabilität*. Herbert Wichmann Verlag, 2007
- [Hofmann-Wellenhof u. a. 2007] HOFMANN-WELLENHOF, Bernhard u. a.: *GNSS - Global Navigation Satellite Systems: GPS, GLONASS, Galileo & more*. Springer Verlag, 2007