

Bachelorarbeit

**Entwurf und Implementierung eines Webservice zur
Visualisierung von Daten und Analyse von Parametern einer
satellitengestützten Erdbeobachtungsmission**

Silke Kerkhoff

2012

**Fakultät für Mathematik
und Informatik** 

Prof. Dr. Ralf Hartmut Güting
Lehrgebiet Datenbanksysteme für
neue Anwendungen



**Deutsches Zentrum
DLR für Luft- und Raumfahrt e.V.**

in der Helmholtz-Gemeinschaft

Dipl. Phys. Daniele Dietrich

DFD

Deutsches Fernerkundungsdatenzentrum

Matrikelnummer: 7499221

Eigenständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst und keine anderen als die im Literaturverzeichnis angegebenen Hilfsmittel verwendet habe. Insbesondere versichere ich, dass ich alle wörtlichen und sinngemäßen Übernahmen aus anderen Werken als solche kenntlich gemacht und mit genauer Quellenangabe dargelegt habe.

Oberpfaffenhoffen, den _____ Unterschrift _____

Danksagung

An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mich bei der Anfertigung dieser Bachelorarbeit unterstützt haben. Ein besonderer Dank gilt meinen beiden Korrektoren, im Speziellen Ralf Cordes.

Des Weiteren möchte ich mich bei Frau Daniele Dietrich bedanken, die mich im Rahmen der Bachelorarbeit im Deutschen Zentrum für Luft- und Raumfahrt betreut hat und mir jederzeit mit Rat und Tat zur Seite gestanden ist.

Auch gilt mein Dank vielen meiner Kollegen, die mir während der Anfertigung dieser Abschlussarbeit in meiner Arbeitsgruppe den Rücken frei gehalten haben, die ich zu jeder Zeit um Rat fragen konnte und mich immer unterstützen.

Nicht zuletzt möchte ich mich bei meiner Familie und Freunden bedanken, die mir bei dieser Arbeit und während meines Studiums moralisch beiseite standen.

Inhaltsverzeichnis

Eigenständigkeitserklärung	iii
Danksagung	iv
Inhaltsverzeichnis	v
Abbildungsverzeichnis	vii
1 Einleitung	1
1.1 Motivation	1
1.2 Zielsetzung	2
1.3 Aufbau der Arbeit	2
2 Grundlagen	5
2.1 TanDEM-X-Mission	5
2.1.1 Der Satellit	6
2.1.2 Synthetisches Apertur Radar (SAR)	6
2.1.3 Bodensegment	7
2.1.4 Anwendung und Nutzungen	9
2.2 Grundlagen zu Geodatenbanken	11
2.2.1 Komponenten eines Datenbanksystems	11
2.2.2 Geodaten	12
2.3 Webanwendungen zur Darstellung von Geodaten	15
2.3.1 Geographische Informationssysteme	15
2.3.2 OGC-Webservice	17
2.3.3 UMN MapServer	18
2.3.4 GeoServer	19
2.3.5 Mapbender	20
2.3.6 OpenLayers	21
2.3.7 ArcIMS	22
2.3.8 Google Earth	23
3 Entwurf	25
3.1 Stand der Technik	26
3.2 Anforderungen	27
3.3 Kriterien für die Auswahl der Software	27
4 Implementierung	29
4.1 Logische Struktur	29
4.2 Physikalische Struktur	31
4.2.1 Hardware	31

4.2.2	Software	31
4.3	Vorgehensweise	33
4.3.1	Verwaltung der Software mit Maven	33
4.3.2	GDAS-Installation	34
4.3.3	Datenbank	35
4.4	GeoServer	38
4.5	Visualisierung	44
4.5.1	Provider	45
4.5.2	EGP	48
4.5.3	API	48
4.5.4	GWT	49
4.5.5	WEB	51
5	Schlussfolgerung	55
A	Glossar	57
B	Abkürzungen	59
C	Literaturverzeichnis	61

Abbildungsverzeichnis

2.1	TanDEM-X-Mission	5
2.2	TanDEM-X-Satellit	6
2.3	Aufnahmemodi des SAR	7
2.4	Stationsnetze TanDEM-X und TerraSAR-X	8
2.5	Anwendungen	10
2.6	Vektor- und Rastermodell	12
2.7	Architektur eines Web-GIS	17
2.8	ArcIMS Ablauf	22
3.1	DEM-Generierung	25
4.1	Logische Struktur	29
4.2	GDAS Hardware	31
4.3	Einstiegsseite GeoServer	39
4.4	Neuen Arbeitsbereich anlegen	40
4.5	Neue Datenquelle anlegen	41
4.6	Layers	42
4.7	Editieren eines Layers	43
4.8	Generelle Architektur	44
4.9	WFS -Nachrichtenaustausch	45
4.10	Model View Presenter	50
4.11	Suchauswahl	52
4.12	Suche	53
4.13	Ergebnisse	53
4.14	Karte mit überlagerten Polygonen	54

Kapitel 1

Einleitung

"The world is complex, dynamic, multidimensional; the paper is static, flat. How are we to represent the rich visual world of experience and measurement on mere flatland?"

[Envisioning Information, Edward R. Tufte]

1.1 Motivation

Mit der satellitengestützten Erdbeobachtung werden wichtige Parameter der Erde erfasst. Diese Parameter dienen als Grundlage, um die Entwicklungen unserer Umwelt besser zu verstehen und Prognosen zu erstellen. Durch den globalen Klimawandel und die Auswirkungen auf das menschliche Leben ist der Bedarf an Erfassung und Auswertung dieser Parameter sehr groß. Durch die globale Erwärmung sind die frostfreien Perioden in den höheren und mittleren Breiten deutlich verlängert. Wichtig für die Natur und die Gesellschaft ist in diesem Zusammenhang die Erhöhung des Meeresspiegels, die mit dem Temperaturanstieg und dem damit verbundenen Abschmelzen von Gletschern einhergeht.

Es wird immer wichtiger, auch räumlich abgeschiedene Regionen regelmäßig zu beobachten und daraus Erkenntnisse zu gewinnen. Erst mit der Satellitenfernerkundung können solche Gebiete wiederholt untersucht werden. Ein großes Projekt der satellitengestützten Erdbeobachtung im DLR (Deutsches Zentrum für Luft- und Raumfahrt) ist die TanDEM-X-Mission (TerraSAR-X add-on for Digital Elevation Measurement). Sie hat zum Ziel, ein hoch genaues, dreidimensionales Abbild unserer gesamten Erde in einheitlicher Qualität und einen globalen Datensatz mit bisher unerreichter Genauigkeit zu erstellen. Ein besonderes Merkmal der TanDEM-X-Mission ist die hohe räumliche Auflösung, die es Wissenschaftlern erlaubt, detaillierte Bodenmerkmale zu erkennen. Die Anwendungen der Daten für wissenschaftliche Zwecke obliegt dem DLR.

Interessierte Wissenschaftler können Projektvorschläge beim DLR einreichen und erhalten die Daten, sobald das Projekt angenommen ist, zu Kopierkosten. Alle Daten werden registrierten Nutzern im DLR-Katalog EOWEB (Earth Observation on the WEB) zur Verfügung gestellt und können dort bestellt werden. Für die Erstellung dieses digitalen Geländemodells der gesamten Erde werden interferometrische SAR-Daten über einen Zeitraum von 3 Jahren parallel zum operativen Betrieb von TerraSAR-X empfangen. TerraSAR-X ist der Zwillingssatellit von TanDEM-X und arbeitet schon seit 2007 als Erdbeobachtungssatellit im Weltraum.

Die vorliegende Arbeit bezieht ihre Motivation aus den Möglichkeiten, die sich aus der Technologie der Webservice ergeben. In den letzten Jahren wurden zunehmend GIS-Technologien entwickelt, um die Geodaten einer breiten Nutzerschicht zugänglich zu machen. Auf Basis von standardisierten Geo-Webservices sollen die Möglichkeiten im Bereich anwendungsorientierter Entwicklung von Software im Hinblick auf die Geodaten, die aus der TanDEM-X-Mission gewonnen werden, genutzt werden. Die Komplexität der Geodaten und deren Nutzung soll jedoch

dem Anwender verborgen bleiben. Diese Arbeit soll Wissenschaftlern einen vereinfachten Zugang zu einer großen Menge an Geodaten bereitstellen und dadurch Auswertungen über globale Veränderungen möglich machen.

1.2 Zielsetzung

Die systematische Verarbeitung der empfangenen SAR-Rohdaten zu sogenannten Raw DEMs wird in einem einzigen Verarbeitungsschritt durchgeführt. Dieses System wird integrierter TanDEM-X-Prozessor (ITP) genannt. Als Raw DEM wird das Digitale Geländemodell bezeichnet, das nach der interferometrischen Verarbeitung der SAR-Rohdaten entsteht. Die interferometrische Verarbeitung liefert Informationen über das Objekt, in unserem Fall die Erde, in dem die unterschiedlichen Effekte der Strahlungen ausgewertet werden. Das Raw DEM ist das Eingabeprodukt für ein zweites System, in dem die endgültige Höhe global kalibriert wird. Dieses System heißt DEM Mosaicking and Calibration Processor (MCP) und generiert das fertige dreidimensionale Höhenmodell.

Eine Initiative des Deutschen Fernerkundungsdatenzentrums (DFD) für den standardisierten Zugriff auf raumbezogene Daten ist das Projekt "Geoservice" mit seiner Softwarekomponente GDAS (Geospatial Data Access Service). GDAS bietet standardisierte Web-Dienstleistungen für den Zugriff auf Geodatenätze und Fernerkundungsdatenprodukte. GDAS unterstützt die Erstellung von interoperablen Geodaten-Diensten und die Integration von kommerzieller und Open-Source-Software.

Um die Entwicklung des integrierten TanDEM-X-Prozessors zu unterstützen, soll ein Webservice entwickelt werden, der es Wissenschaftlern ermöglicht, eine Qualitätskontrolle der verarbeiteten TanDEM-X-Daten durchzuführen. Die Grundidee liegt hierbei in der visuellen Darstellung der Daten. Mit Hilfe einer computergestützten Visualisierung können große Datenmengen in ihrer Gesamtheit bewertet und analysiert werden. Anhand geeigneter Parameter, die bei der Generierung des Raw DEMs eingeflossen sind, sollen bestimmte Geodaten ausgewählt werden und mit einer Software, zum Beispiel Google Earth oder OpenLayers, dem Anwender visuell über das Internet zur Verfügung gestellt werden. Die Geodaten sollen in einer Datenbank abgelegt werden, auf die mittels GDAS standardisiert zugegriffen werden kann. Wissenschaftlern soll damit ein einfacher Webservice bereitgestellt werden, der die Daten visuell aufbereitet und es ihnen ermöglicht, einfache Analyse-Tätigkeiten durchzuführen.

1.3 Aufbau der Arbeit

Die Bachelorarbeit ist folgendermaßen aufgebaut:

Kapitel 1: beschreibt die Motivation, Problemstellung und die Zielsetzung der Arbeit.

Kapitel 2: erläutert das TanDEM-X-Projekt näher, und es wird auf den Satelliten, das Bodensegment und die Anwendungsfelder eingegangen. Außerdem wird das grundlegende Konzept des RADAR-Sensors mit einer synthetischen Apertur (SAR) knapp beschrieben. Darüber hinaus werden die Grundlagen für die Verwendung von Geodaten dargestellt und damit die Grundlagen für die folgenden Kapitel gelegt. Es wird beschrieben, was Geodaten sind und wie man mit ihnen arbeiten kann. Der letzte Abschnitt des 2. Kapitels gibt eine Übersicht über die Webanwendungen für die Verarbeitung von Geodaten.

Kapitel 3: erläutert die bisherigen Gegebenheiten, welche Anforderungen an die Software gestellt und welche Kriterien für die Auswahl der verwendeten Werkzeuge in Betracht gezogen werden.

Kapitel 4: beschreibt die Umsetzung und Implementierung des Webservice. Hier wird auf die Installation der GDAS-Software, das Einrichten der Geodatenbank und des GeoServers, sowie die Erstellung des Webservice zur Visualisierung der Daten eingegangen.

Kapitel 5: enthält Schlussfolgerungen und gibt Empfehlungen.

Kapitel 2

Grundlagen

Dieses Kapitel gibt eine Einführung über die TanDEM-X-Mission. Gleichzeitig werden die Besonderheiten bei der Verarbeitung von Daten mit räumlichem Bezug herausgestellt und erläutert. Abschließend wird auf das Konzept und die Verwendung dieser Daten im Internet eingegangen. Dieses Kapitel soll als Grundlage für die folgenden Kapitel betrachtet werden.

2.1 TanDEM-X-Mission

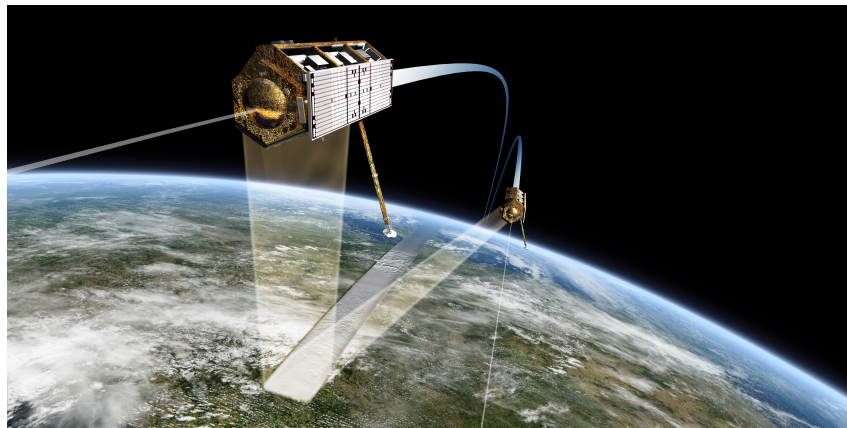


Abbildung 2.1: TanDEM-X-Mission (Quelle: DLR)

Die TanDEM-X-Mission (TerraSAR-X add-on for Digital Elevation Measurement) basiert auf zwei nahezu identischen Erdbeobachtungssatelliten, TerraSAR-X und TanDEM-X [DLR] (siehe Abbildung 2.1).

Der TerraSAR-X-Satellit ist bereits am 15. Juni 2007 ins All gestartet, TanDEM-X folgte ihm am 21. Juni 2010. Beide Satelliten fliegen mit einer Geschwindigkeit von über 27.000 Kilometern pro Stunde mit einem Abstand von 200 bis 500 Metern in einer Höhe von 514 Kilometer hintereinander her. Das Ziel dieser Mission ist die Erstellung eines globalen digitalen Geländemodells mit einer bisher nicht erreichten Genauigkeit. Für ein Zwölf-Meter-Raster (etwa die Breite einer Straße) werden von dem System Höheninformationen geliefert, die bis auf zwei Meter genau sind. Mittels der durchgehenden Homogenität bietet die TanDEM-X-Mission die Basis für ein einheitliches Kartenmaterial ohne Brüche durch Landesgrenzen und unterschiedliche Messverfahren. Die beiden Satelliten sind das erste konfigurierbare SAR-Interferometer im Weltraum. Mit Hilfe von SAR-Interferometrie ist es anhand von zwei Radarbildern möglich, Höhenunterschiede zu ermitteln. Hierfür müssen die Radarbilder dasselbe Gebiet in unterschiedlichen Winkeln

abbilden. Bei der TanDEM-X-Mission wird dies durch die zwei Satelliten realisiert. Agiert ein Satellit als Sender und beide empfangen die von der Erdoberfläche zurückgestreuten Signale, befinden sich die beiden Satelliten im bistatischen Betrieb. Es wird ein dreijähriger Parallelbetrieb im Formationsflug benötigt, um eine vollständige globale Abdeckung zu erhalten. Innerhalb dieser 3 Jahre kann das Satellitenpaar die gesamte Landoberfläche der Erde von 150 Millionen Quadratkilometern vermessen und erzeugt dabei eine Datenmenge von etwa 1,5 Petabyte.

2.1.1 Der Satellit

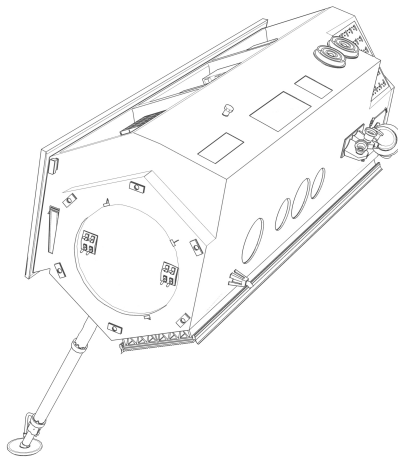


Abbildung 2.2: TanDEM-X-Satellit (Quelle: DLR)

Der Satellit, der in Abbildung 2.2 zu sehen ist, ist 5 Meter lang und hat einen Durchmesser von 2,4 Metern. Da sich der Satellit auf einer sonnensynchronen Kreisbahn befindet, kann die Energieversorgung über ein an der linken Seite des Satelliten fest installiertes Solarpanel erfolgen. Um den Formationsflug zu ermöglichen, musste der TanDEM-X-Satellit, im Gegensatz zum TerraSAR-X-Satelliten, ein wenig verändert werden. TanDEM-X besitzt zusätzlich ein Antriebssystem, das mit Hochdruck-Stickstoffgas arbeitet und feine Bahnkorrekturen unterstützt, um den Formationsflug aufrechtzuerhalten. Um Bahnkorrekturen auszuführen, erhält der TanDEM-X-Satellit Informationen zur Position und Geschwindigkeit von TerraSAR-X. Die Positionen beider Satelliten werden über spezielle GPS-Empfänger auf den Satelliten bestimmt. Außerdem können beide Satelliten über 6 Hornantennen gegenseitig Informationen austauschen. Obwohl die Position der beiden Satelliten bekannt ist, müssen weitere Sicherheitsvorkehrungen getroffen werden, um das Kollisionsrisiko zu minimieren. Zur Regelung der Lage wird das Magnetfeld der Erde und ein künstliches Magnetfeld auf beiden Satelliten benutzt. Dadurch können Drehraten der Satelliten stabilisiert werden. Eine weitere Sicherheitsmaßnahme verhindert, dass sich die Satelliten gegenseitig bestrahlen. Es existieren Abschnitte in der Umlaufbahn, in denen einer der beiden Satelliten keine Radarpulse senden darf. Diese Bereiche nennt man Sperrzonen (engl. *exclusion zones*).

2.1.2 Synthetisches Apertur Radar (SAR)

Radarsatelliten im herkömmlichen Sinne arbeiten nach dem Prinzip des Synthetischen Apertur Radars (SAR). Hierbei wird die Erde mit einem Strahl kohärenter Mikrowellenstrahlung, wie bei einem Laser, beleuchtet. Das Radar sendet Mikrowellenpulse aus, die von der Erdoberfläche reflektiert werden, um dann wieder von dem Radar empfangen zu werden. Der Abstand des

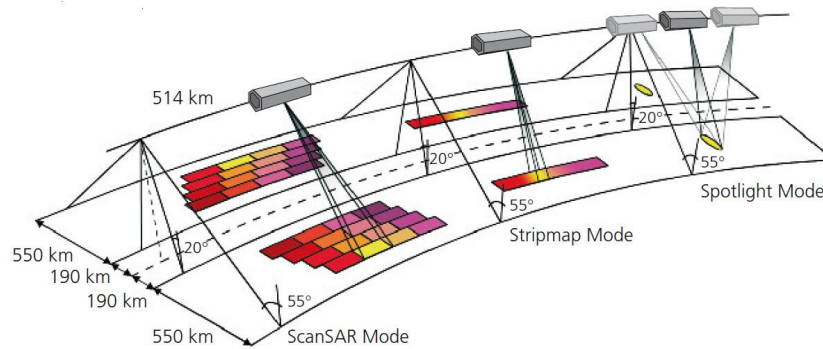


Abbildung 2.3: Aufnahmemodi des SAR: Spotlight, Stripmap und ScanSAR (Quelle: DLR)

Satelliten zur Erdoberfläche ergibt sich aus der Laufzeit der Signale. Das Radar zeichnet die Signale sequenziell auf und beleuchtet einen Streifen am Boden, da sich der Satellit um die Erde bewegt. Die Echo-Signale enthalten auch Informationen über die Oberflächenbeschaffenheit, und nach einer aufwendigen Signalverarbeitung entsteht ein zweidimensionales Bild des Gebietes. Mit Radar wird in einem Wellenlängenbereich gearbeitet, in dem die Strahlen ungehindert durch Nebel, Wolken oder Regen durchdringen. Radarsysteme beleuchten die Erdoberfläche aktiv, so dass sie unabhängig von der Tageszeit eingesetzt werden können. Diese beiden Eigenschaften machen den Einsatz von Radar zeit- und wetterunabhängig. Bei der TanDEM-X-Mission haben wir nun zwei Satelliten, die ein Gebiet von zwei unterschiedlichen Positionen abbilden. Ähnlich wie beim menschlichen Auge erhält man zusätzlich gewünschte Höheninformationen, um ein digitales Geländemodell zu erstellen.

Es gibt unterschiedliche Betriebsmodi, siehe Abbildung 2.3 bei der TanDEM-X-Mission. Der Standardbetriebsmodus ist der sogenannte "Stripmap-Modus", bei dem der Satellit kontinuierlich 30 km breite Streifen mit einer Auflösung von ca. 3 Metern aufnimmt. Beim "ScanSAR-Modus" werden komplizierte Sequenzen elektronischer Antennen-Schaltungen genutzt, um einen großen Streifen, über 100 km breit, mit einer Auflösung von 18 Metern am Boden auszuleuchten. Der "Spotlight-Modus" besitzt eine Auflösung von bis zu einem Meter. Hierbei werden innerhalb von Sekunden viele Schaltungen an der Antenne durchgeführt, um die Radarstrahlen auf ein etwa 5 - 10 Kilometer großes Gebiet auf der Erde gerichtet zu halten. Würde man nur eine Antenne zur Verfügung haben, wäre ein leicht versetzter, zweiter Überflug notwendig, um unterschiedliche Abstandsdifferenzen zu einem Punkt am Boden zu erhalten. Der Satellit benötigt jedoch bei seiner Erdumkreisung mehrere Tage, bis er wieder denselben Punkt auf der Erde aufnehmen kann. Zwischenzeitliche Veränderungen auf der Erdoberfläche, zum Beispiel Pflanzenwuchs, machen die Höhenbestimmung dadurch ungenau. Bei der TanDEM-X-Mission wird dies vermieden, da die zwei Bilder fast gleichzeitig (max. 0,08 Sekunden Zeitversatz) entstehen. Die Höhengenaugigkeit des digitalen Geländemodells ist somit unabhängig von zeitlichen Veränderungen.

2.1.3 Bodensegment

Die Messdaten der beiden Satelliten müssen zum Boden übermittelt, dort empfangen und an ein Datenzentrum weitergeleitet werden, um dort zu fertigen Produkten verarbeitet zu werden. Diese Datenempfangs- und Verarbeitungskette wird in extra hierfür entwickelten Anlagen am Boden, dem sogenannten Bodensegment, durchgeführt. Über ein Netz von vier Bodenstationen, Neustrelitz in Deutschland, Kiruna in Schweden, Inuvik in Kanada und O'Higgins in der

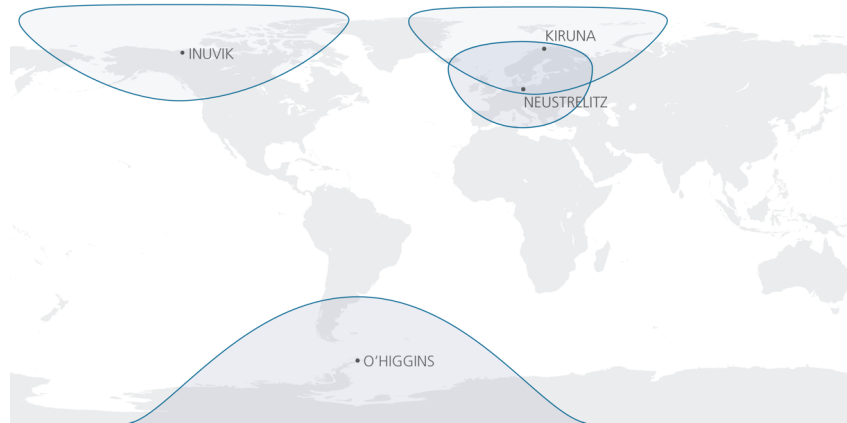


Abbildung 2.4: Stationsnetze TanDEM-X und TerraSAR-X
(Quelle: DLR)

Antarktis werden die Daten der beiden Satelliten empfangen. In Abbildung 2.4 sind die Empfangsbereiche der unterschiedlichen Stationen dargestellt. Das DLR-Bodensegment besteht aus drei Hauptelementen: dem Missionsbetriebssegment (MOS), dem Instrumentenbetriebs- und Kalibrierungssegment (IOCS) und dem Nutzlastbodensegment (PGS). Hauptaufgaben des Missionsbetriebssegments ist die Kommandierung und Kontrolle der Satelliten. Außerdem führt es Analysen über die Umlaufbahn durch, plant die Aufnahmen und stellt Zusatzdaten für die Verarbeitung der Daten bereit. IOCS ist für den Betrieb der Instrumente zuständig, überwacht also die Instrumente und die Kalibrierung des SAR-Systems. Das Nutzlastbodensegment ist für den Datenempfang und die Archivierung der Daten, wie auch für die Generierung der Basisprodukte zuständig. Für die Aufnahme des globalen Geländemodells bedarf es einer bestimmten Strategie für die Datenaufnahme. Sie ist abhängig von der Art und Weise, wie die Satelliten fliegen. Für die Planung ist eine schnelle Rückmeldung und Beurteilung der Daten sehr wichtig. Dies wird bei langen zusammenhängenden Aufnahmen erschwert, da sie wegen der großen Datenmenge an unterschiedlichen Stationen empfangen werden. Der Versand der Daten erfolgt per Postweg an das Deutsche Fernerkundungsdatenzentrum und kann bei der Antarktisstation mehrere Wochen dauern.

Im Folgenden wird der Arbeitsablauf in diesem komplexen System von der Aufnahme bis zur Auslieferung der Daten in einer Übersicht dargestellt. Dies dient zum Verständnis und zur Erklärung einiger Begriffe, die für den weiteren Verlauf der Arbeit benötigt werden.

Die Erstellung eines digitalen Geländemodells ist das Hauptziel der TanDEM-X-Mission. Die hierfür benötigten Aufnahmen haben Vorrang vor TerraSAR-X-Daten und TanDEM-X-Daten, die von Wissenschaftlern bestellt wurden. Wie oben beschrieben erfolgt die Planung beim MOS und ist äußerst kompliziert, da viele Parameter einfließen. Das Ergebnis der Planung sind genaue Anweisungen für den Satelliten, mit welchen Einstellungen und wann er bei einem Überflug die Aufnahme von SAR-Daten ausführen soll.

Ein Data Take bezeichnet die Datei, die die innerhalb eines bestimmten zusammenhängenden Zeitraums aufgenommenen SAR-Daten enthält. Die gewünschten Aufnahmen kann man als eine Bestellung ansehen, die in der TanDEM-X-Mission als Akquisition bezeichnet wird. Ein Data Take wird unter Umständen mehrere Akquisitionen enthalten. An Bord werden die Data Takes in einem Arbeitsspeicher, dem Solid State Mass Memory (SSMM), gespeichert. Befindet sich der Satellit im Empfangsbereich einer Bodenstation, werden mehrere Data Takes auf dem SSMM ausgewählt, verschlüsselt und zur Empfangsstation übermittelt. Diesen Prozess nennt man Downlink. An den Bodenstationen werden die Daten wieder entschlüsselt.

Der erste Schritt für die Verarbeitung dieser Rohdaten wird von dem integrierten TanDEM-

X-Prozessor (ITP) durchgeführt, er ist sozusagen das Herzstück bei der Generierung des digitalen Geländemodells aus den TanDEM-X-Daten. Zu seinen Aufgaben gehört das Screening, die bistatische Fokussierung und die interferometrische Verarbeitung der Daten. Bei der Interferometrie im Radarbereich wird von mindestens zwei SAR-Aufnahmen die Phasendifferenz ausgewertet. Somit kann bei der Erfassung der zurückkommenden Signale die Veränderung der Erdoberfläche gemessen werden. Bistatisch bedeutet, dass sich der Sender und der Empfänger auf unterschiedlichen Plattformen befinden. Im Falle des TanDEM-X-Projekts sind dies zwei unterschiedliche Satelliten. Die Untersuchung der Rohdaten heißt Screening von TanDEM-X-Aufnahmen und wird an den Bodenstationen für jedes entschlüsselte Data-Take-Segment durchgeführt. Es werden hierbei qualitäts- und akquisitionsbedingte Parameter aus den Rohdaten extrahiert und in Dateien herausgeschrieben, wobei die Rohdaten selbst nicht verändert werden. Die Datenmenge dieser Dateien ist sehr gering, und sie werden über Ethernet von den Bodenstationen zum DLR nach Oberpfaffenhofen übertragen. Die Rohdaten dagegen werden verschlüsselt auf Bänder geschrieben und per Post nach Oberpfaffenhofen in das Institut Deutsches Fernerkundungsdatenzentrum (DFD) des DLR geschickt.

Im DFD werden die Dateien und Parameter, die via Ethernet übertragen worden sind, im ITP einem interferometrischen Qualitäts-Pre-Check (IQPC) unterzogen. Der IQPC umfasst unter anderem die Bestimmung des Zeitpunktes, der geometrischen Lage (Position und Orientierung) und der räumlichen Abdeckung der Aufnahmen. Er erstellt Informationen zu der Konfiguration der beiden Satelliten und führt weitere Analysetätigkeiten durch. Für die Herstellung der digitalen Geländemodelle werden noch zusätzliche Informationen, zum Beispiel die Position oder die Lage der Satelliten, benötigt. Sobald alle benötigten Informationen und die Rohdaten für die Herstellung eines DEMs (Digital Elevation Model) in Oberpfaffenhofen vorhanden sind, beginnt der ITP mit dessen Generierung. Zuerst werden die Rohdaten in Blöcken von 8 Sekunden mit einem geringen Überlappungszeitraum aufgeteilt (PRCSCR). Diese Blöcke heißen Szenen. Aus jeder dieser Szenen wird ein Rohdaten-DEM generiert (RDEMGEN). Die einzelnen Szenen der Rohdaten-DEMs werden im nächsten Verarbeitungsschritt (MCP) zu einem flächendeckenden DEM zusammengefügt. Am Ende der Verarbeitungskette werden alle erzeugten Produkte und Zwischenprodukte in einer Langzeit-Datenbank abgelegt. Von dort aus wird die Auslieferung der Produkte an die Wissenschaftler überwacht.

2.1.4 Anwendung und Nutzungen

Die bei der TanDEM-X-Mission erhaltenen Daten können in mehreren Bereichen Anwendung finden. Je nachdem, welche Anwendungsbereiche betrachtet werden sollen, gibt es zwei unterschiedliche Aufnahmetechniken. Für beide Techniken wird das gleiche Radarsignal von beiden Satelliten aufgenommen und die Differenz der Weglänge gemessen.

Bei der sogenannten Across-Track-SAR-Interferometrie bewegen sich die Satelliten nebeneinander. Dieses Verfahren wird für die Erstellung von Topografischen Karten, der Vermessung von Landschaftsnutzung und Vegetation verwendet und hilft bei der Navigation von Flugzeugen bei eingeschränkter Sicht. Ein großer Anwendungsbereich findet sich auch im Krisenmanagement wieder. Überschwemmungsgebiete können mit Hilfe von hochauflösenden digitalen Geländemodellen besser untersucht werden, ebenso Erdbeben und Vulkanregionen. Mittels TanDEM-X-Daten können Veränderungen der Eismassen von Gletschern erkannt werden, und es lassen sich Erkenntnisse über die Dynamik der polaren Eiskappen erschließen.

Die zweite Aufnahmetechnik nennt sich Along-Track-SAR-Interferometrie, hier ist die Ausrichtung der Antennen des Radars und die Flugrichtung voneinander unabhängig. Hiermit lassen sich die Geschwindigkeiten von bewegten Objekten messen. Diese Messungen können von hohem Interesse bei Verkehrsbeobachtungen sein. Auch können damit die Fließgeschwindigkeit von Gletschern näher bestimmt oder Rückschlüsse auf Wasserströmungen gezogen werden.

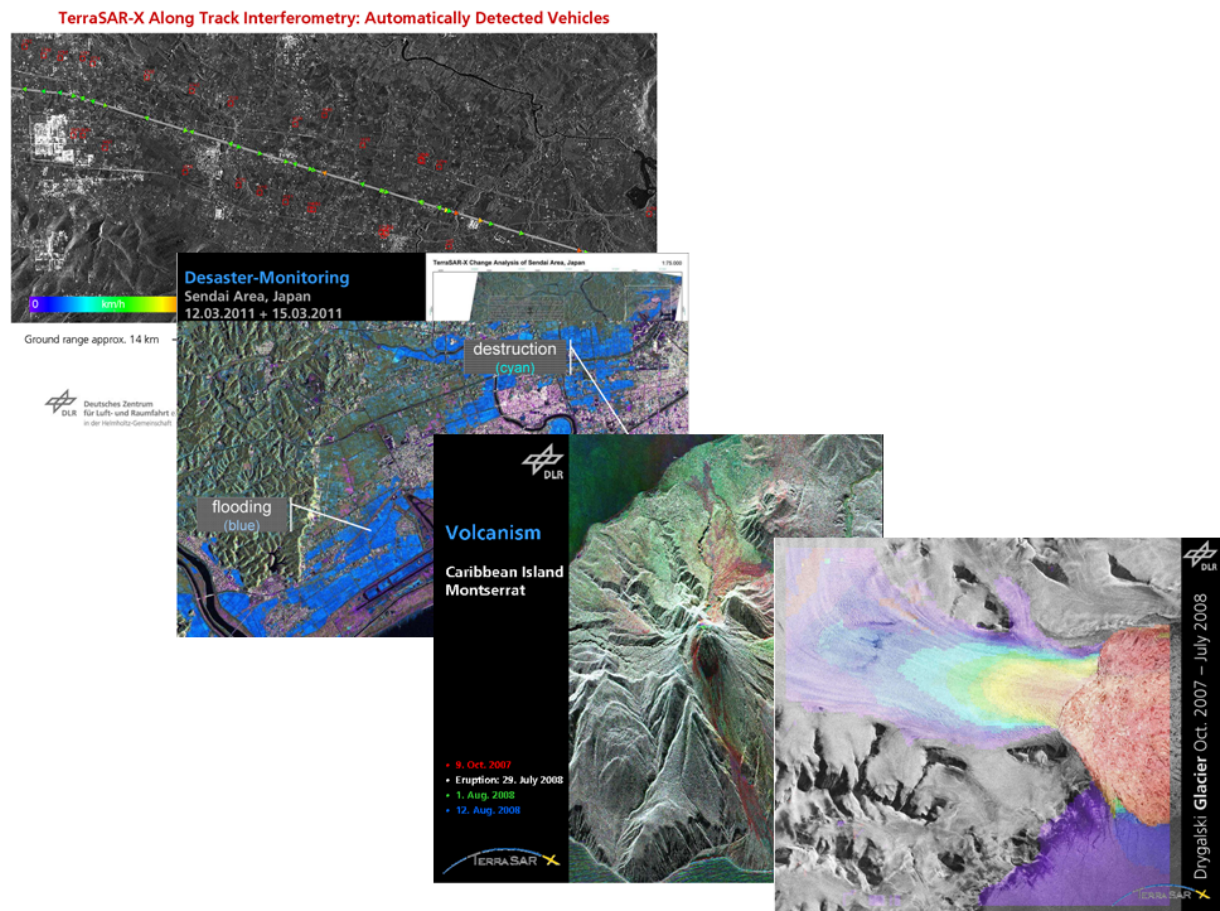


Abbildung 2.5: Anwendungen (Quelle: DLR)

Mit der TanDEM-X-Mission wird eine innovative Radartechnik Anwendung finden. Diese Technik heißt POL-InSAR und bedeutet, dass die Radarwellen polarisiert abgestrahlt werden können. Es ergeben sich dadurch Unterschiede bei den reflektierten Wellen je nach Material, auf das die Strahlen auftreffen. Die Wissenschaftler erhoffen sich mit dieser Technik neue Erkenntnisse in den unterschiedlichsten Bereichen. In Abbildung 2.5 werden einige Anwendungsbereiche dargestellt.

2.2 Grundlagen zu Geodatenbanken

Datenbanken finden in der heutigen Zeit in fast allen Bereichen der Datenverarbeitung Anwendung. Im Alltag ist man einer regelrechten Informationsflut ausgesetzt. Damit die gesammelten Informationen als Entscheidungsgrundlagen dienen können, müssen sie gespeichert, geordnet und zur Auswertung bereitgestellt werden. Datenbanksysteme vereinfachen den Umgang mit den Daten und haben gegenüber einfachen Dateisystemen mehrere Vorteile. Durch die zentrale Verwaltung der Daten in der Datenbank können Redundanzen und Inkonsistenzen vermieden werden. Mit der persistenten Speicherung der Daten wird dem Datenbankmanagement die Möglichkeit gegeben, die Datenbank nach dem Auftreten von Fehlern wieder in einen konsistenten Zustand zu versetzen. Auch können mehrere Benutzer gleichzeitig auf den Datenbestand zugreifen, ohne sich gegenseitig zu stören, um nur einige Vorteile von Datenbanken zu nennen. Um eine einfache Datenmanipulation zu realisieren, werden den Nutzern Werkzeuge zur Verfügung gestellt. In dieser Arbeit wird auf räumliche Datenbanken eingegangen, die die Verarbeitung und Abspeicherung von geometrischen Daten, auch Geodaten genannt, erleichtern. Geodaten sind sehr komplex, und die Struktur einer herkömmlichen Datenbank ist den Anforderungen bei der Verarbeitung von Geodaten nicht gewachsen. Aus diesem Grund wurden hierfür spezielle Datenbanksysteme entwickelt, die die Speicherung, Modellierung und Anfragen auf räumliche Daten ermöglichen. In [güt94] werden zwei Arten von Systemen unterschieden: Geodatenbanksysteme und Bilddatenbanksysteme. Bilddatenbanksysteme enthalten Analysetechniken, um räumliche Objekte aus Bildern herauszuziehen, sowie einige Funktionalitäten von Geodatenbanken und können Rasterbilder verarbeiten. Wie in [güt94] beschrieben verfügt ein Geodatenbanksystem über weitere Möglichkeiten, Geodaten zu verarbeiten. Diese Datenbanksysteme stellen Operationen bereit, die auf geometrische Datentypen angewendet werden können, zum Beispiel die Schnittmenge von zwei Flächen oder die Berechnung der Streckenlänge zwischen zwei Punkten. Die verbreitetsten Anfragen auf Geodaten sind Punkt-, Rechteck-, Regions-, Abstand-, nächster-Nachbar- (engl. *Nearest-Neighbor*) Anfrage und der räumliche Verbund (engl. *Spatial Join*). Die Interoperabilität und der Datenaustausch zwischen verschiedenen Systemen wird durch eine standardisierte Datenhaltung der Geobasis- und der Geofachdaten (siehe Unterabschnitt Geodaten) und durch ein standardisiertes Datenaustauschformat ermöglicht. Weitere Erklärungen zur Standardisierung befinden sich im Unterabschnitt Standardisierung von Geodaten.

Die Begriffe Datenbanksystem und Datenbank werden in der Alltagssprache häufig nicht eindeutig verwendet. Um eine korrekte Vorstellung dieser Begriffe zu bekommen, werden nachfolgend die Komponenten einer Datenbank kurz erläutert, um danach die Besonderheiten mit dem Umgang von Geodaten hervorzuheben.

2.2.1 Komponenten eines Datenbanksystems

Ein Datenbanksystem besteht aus zwei Komponenten, der Datenbank, die die Daten enthält, und dem Datenbankmanagementsystem (DBMS), dessen Aufgabe der Zugriff auf die Daten sowie deren Verwaltung und Organisation ist.

Mithilfe des DBMS lassen sich Daten abfragen, hinzufügen, löschen und ändern, wobei das DBMS dafür sorgt, dass keine Konflikte entstehen, wenn mehrere Anwender die Datenbank gleichzeitig nutzen. Das DBMS orientiert sich am Datenmodell, das die Struktur der Datenbank beschreibt, d. h. die Art und Weise, wie die Daten in der Datenbank abgelegt sind, wie sie zueinander in Beziehung stehen, welche Datentypen verwendet werden und welche Bedingungen die Daten erfüllen müssen. Eine besonders wichtige Anforderung an das Datenmodell ist, dass die Persistenz und Integrität der Daten gewährleistet ist.

2.2.2 Geodaten

Um eine möglichst genaue Abbildung der Erde zu gewinnen, benötigt man Daten mit einem räumlichen Bezug. Diese Daten werden Geodaten genannt, die sich über den räumlichen Bezug miteinander verbinden lassen. Gemäß [geoport] lassen sich Geodaten in Geobasisdaten und Geofachdaten unterteilen, die wie folgt definiert werden:

"Geobasisdaten sind grundlegende amtliche Geodaten, welche die Landschaft (Topographie) sowie die Liegenschaften anwendungsneutral in einem einheitlichen geodätischen Bezugssystem beschreiben. Geobasisdaten werden durch die Vermessungsverwaltungen der Länder erhoben, bereitgestellt und gepflegt. Sie sind Grundlage für Geofachdaten sowie für amtliche Kartenwerke (Deutsche Grundkarte, Topographische Karten)."

"Geofachdaten sind thematische Daten mit Raumbezug, die einerseits die geographischen Koordinaten (Geobasisdaten) und andererseits eine dazugehörige Information angeben."

Damit diese Daten in einer Geodatenbank abgelegt werden können, müssen die Eigenschaften der Geodaten näher beschrieben werden. Anhand dieser Eigenschaften können Datenbankmodelle entwickelt werden, die einen möglichst effektiven und einfachen Zugriff auf die Geodaten ermöglichen.

Modellierung von Geodaten

Um die Modellierung zu verfeinern, werden im Folgenden Eigenschaften der Geodaten und ihre Verwendung dargestellt. [Bri 2008] unterteilt diese Eigenschaften in:

Geometrische Eigenschaften Sie dienen zur Beschreibung der Lage und Ausdehnung von Objekten im Raum. Durch ein räumliches Bezugssystem wird dem Objekt die Position im Raum eindeutig zugewiesen. Die geometrischen Informationen werden entweder in einem Vektor- oder einem Rastermodell dargestellt.

Beim Vektormodell ist der Punkt das Grundelement. Das Objekt wird durch seine Koordinaten auf der XY-Achse definiert, womit sich die Geometrie des Objektes sehr genau beschreiben lässt.

Rastermodelle bestehen aus einer durch die Auflösung bestimmte Anzahl von Bildpunkten. Der gesamte Datenraum wird in Rasterzellen aufgeteilt. Die Geometrie des Objekts hängt von der Basisgröße der Rasterzelle ab. In Abbildung 2.6 sind die beiden unterschiedlichen Modelle dargestellt. Im Vektormodell wird das Objekt im Raum durch die Punkte P_1 bis P_5 eindeutig beschrieben. Jeder Punkt wird wiederum durch Koordinaten in dem verwendeten räumlichen Bezugssystem bestimmt.

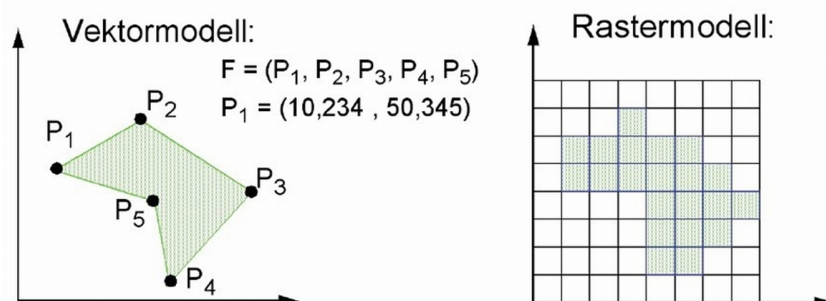


Abbildung 2.6: Vektor- und Rastermodell (Quelle [Bri 2008]):

Topologische Eigenschaften Hier werden die Beziehungen mehrerer Objekte im Raum zueinander beschrieben, also ob die Objekte übereinander/aneinander liegen, sich schneiden oder sich umgeben.

Thematische Eigenschaften Darunter versteht man fachlich relevante Eigenschaften, die wiederum in nominale, qualitative und quantitative Eigenschaften unterteilt werden können. Eine solche Eigenschaft wird durch eine Attributmenge mit Attributwerten beschrieben und entweder mit einem Layer- oder einem Objektmodell dargestellt. Beim Layermodell werden die Attribute in separaten thematischen Ebenen (Layern) erfasst, die je nach Bedarf übereinander gelegt werden können. Im objektbasierten Datenmodell betrachtet man das Geobjekt an sich.

Temporale Eigenschaften beschreiben die zeitliche Veränderung von Eigenschaften. Ein konzeptionelles Modell für temporale Eigenschaften ist das Snapshot-Modell. Zu einem gewählten Zeitpunkt werden die zeitlich variierenden Attribute mit einem Zeitstempel festgehalten. Es entstehen Momentaufnahmen eines zeitlichen Zustandes. Der exakte Zeitpunkt der Änderungen einzelner Attributwerte ist nicht festzustellen, denn die Wertänderungen zwischen zwei Snapshots bleiben unberücksichtigt. [OTT, SWIACZNY 2001]

Standardisierung von Geodaten

Geodaten finden mittlerweile in vielen Bereichen Verwendung. Die hierbei benutzten Daten haben unterschiedliche Herkunft und sind in unterschiedlichen Formaten gespeichert. Um diese Daten für einen weitgefächerten Anwendungsbereich nutzen zu können, ist ein einfacher und effizienter Austausch der Daten von großer Bedeutung. Dazu ist eine Standardisierung der Geodatenformate und -schnittstellen sowie der Geodatenmodelle notwendig. Eine der wichtigsten Standardisierungsorganisationen im GIS (geographische Informationssysteme)-Bereich ist das Open Geospatial Consortium (OGC), das im Jahre 1994 als Open GIS Consortium gegründet und 2004 in das Open Geospatial Consortium umbenannt wurde. Es ist ein internationales Industrie-Konsortium von 423 (Stand 24.07.2011) Unternehmen, Behörden und Hochschulen. Es hat sich zur Aufgabe gemacht, öffentlich zugängliche Standardschnittstellen für den Austausch von Geodaten auf Basis frei verfügbarer Spezifikationen zu entwickeln [ogc]. Nachfolgend werden die OGC-Standards erläutert, die in dieser Arbeit Anwendung finden. Im Kapitel 2.3 Webanwendungen zur Darstellung von Geodaten, werden weitere OGC-Standards vorgestellt.

Feature-Geometry-Modell Das Feature-Geometry-Modell wird für die Beschreibung der Geometrie und der Darstellung von topologischen Eigenschaften bis zur 3. Dimension, also den räumlichen Eigenschaften von Geodaten, verwendet. In dem Modell sind Methoden für Anfrage, Austausch, Verarbeitung, Verwaltung und Zugriff von Geobjekten definiert. Zugrundegelegt wird dem Modell die Feature-Geometry-Spezifikation, die zwei Hauptpakete umfasst. Das Paket "Geometry" beschreibt die geometrischen Eigenschaften und das Paket "Topology" beschreibt die topologischen Eigenschaften der Geobjekte.

Simple-Feature-Modell Das Simple-Feature-Modell ist eine Spezifikation zur Beschreibung der Geometrie von Objekten im zweidimensionalen Koordinatensystem. Als Grundlage hierfür dient das Feature-Geometry-Modell, wobei es Bezeichnungsunterschiede zwischen den Modellen gibt. Für das Simple-Feature-Modell gibt es zwei Formen der Repräsentation, Well-Known Text (WKT) und Well-Known Binary (WKB). WKT wird verwendet, um Geometriedaten im ASCII-Format auszutauschen. Es gibt verschiedene Geometrietypen, POINT, LINE, MULTILINE, POLYGON, MULTIPOLYGON. Ein Punkt mit XY-Koordinaten wird beispielsweise folgendermaßen definiert: Point (1 1). Zusätzlich muss angegeben werden, in welchem Be-

zugssystem, also in welchem Koordinatensystem die Geometriedaten angegeben werden. Hierfür wird der sogenannte Spatial Reference Identifier (SRID) benutzt. Dies ist eine eindeutige Nummer des Bezugssystems und wird von der European Petroleum Survey Group (EPSG) in einem Katalog, der alle EPSG-Codes enthält, definiert. Das World Geodetic System 1984 (WGS84) besitzt beispielsweise den EPSG-Code 4326.

Bei der WKB-Präsentation werden die Geometriedaten als Binärströme übertragen. Die geometrische Information wird in BLOB-Werten gespeichert, wobei das WKT sich in Ein-Byte-Integer ohne Vorzeichen, Vier-Byte-Integer ohne Vorzeichen und Acht-Byte-Zahlen mit doppelter Genauigkeit (IEEE 754 Format) aufteilt. Der Punkt Point(1 1) wird folgendermaßen in Hexadezimalziffern dargestellt:

```
010100000000000000000000F03F000000000000F03F
```

Diese Folge lässt sich in folgende Bestandteile zerlegen:

Byte Order: 01

WKB Type: 01000000

X: 000000000000F03F

Y: 000000000000F03F

Byte Order gibt an, ob die Byte-Reihenfolge Little-Endian=0 oder Big-Endian=1 ist. WKB Type gibt den Geometriotyp an: 1 für Punkt, 2 für Linie, etc. X gibt den Wert auf der X-Achse und Y den Wert auf der Y-Achse an. [MySQL WKB]

SQL/MM Spatial SQL/MM Spatial ist eine Erweiterung von SQL, bei der vordefinierte objektrelationale Datentypen festgelegt werden, die den Simple-Feature-Standard in SQL abbilden. MM steht für Multimedia. SQL/MM ist ein ISO/IEC Standard 13249 und teilt sich in einen allgemeinen Bereich und in mehrere Unterbereiche auf. Der allgemeine Teil wird als Framework bezeichnet, der Definitionen und allgemeine Konzepte zur Verfügung stellt, die in den anderen Teilen verwendet werden. "SQL/MM Full Text" dient zur Verarbeitung und Speicherung von komplexen Texten und bietet verschiedene Arten von Suchmustern an. "SQL/MM Still Images" wird für die Speicherung und Verarbeitung von Bildern in Datenbanken verwendet. "SQL/MM Data Mining" strukturiert benutzerdefinierte Typen, einschließlich der zugehörigen Methoden, um Data Mining zu unterstützen. Und zu guter Letzt "SQL/MM Spatial" definiert Datentypen zu Speicherung von Geodaten.

SQL/MM definiert zwei Tabellen. Die Tabelle GEOMETRY_COLUMNS enthält zusätzliche Informationen für Spalten in Tabellen der Datenbank, die räumliche Daten enthalten. Die Tabelle SPATIAL_REF_SYS speichert Informationen über Referenzsysteme, sie enthält also den EPSG-Code mit den dazugehörigen Parametern der Projektion. In der Implementierung wird auf die Verwendung dieser beider Tabellen näher eingegangen.

Geography Markup Language Die Geography Markup Language (GML) wurde vom Open Geospatial Consortium entwickelt, um einen Standard für die Darstellung von räumlichen Objekten und ihrer Zusammenhänge zu erstellen. Mit GML können geographische Informationen zwischen verteilten Computersystemen ausgetauscht werden, auch wenn die Computersysteme unterschiedliche Software benutzen. Es wird für die Modellierung, die Speicherung und den Austausch von Objekten mit und ohne räumlichen Bezug verwendet. GML basiert auf XML, und wie bei den meisten XML-Sprachen gibt es ein Dokument und ein Schema. Mit dem Schema wird die Struktur der Dokumente spezifiziert, mit dem die Geodaten anwendungsspezifisch beschrieben werden, um somit ein einheitliches Verständnis zu erhalten. Für die Beschreibung der Geometrie existieren drei Basisgeometrien: Points, Lines und Polygons. Mit diesen Basisgeometrien können die meisten Anwendungen bedient werden. Für eine Anwendung muss auf Basis der vordefinierten Strukturen und Typen ein domänenspezifisches GML-Applikationsschema definiert werden. Für

die Bearbeitung und Handhabung der Datenmodelle und Daten können Standard-XML-Tools wie XML-Parser und XML-Editoren eingesetzt werden.

Kartenprojektion Um mit Geodaten arbeiten zu können, muss man wissen, in welchem Koordinatensystem die Daten vorliegen. Generell kann die Lage eines Punktes auf der Erdoberfläche durch die Festlegung der X- und Y-Koordinate beschrieben werden. Die X-Koordinate bestimmt die geographische Länge, die Y-Koordinate die geographische Breite. Linien und Polygone bestehen aus mehreren Punkten und können anhand der Punkte eindeutig bestimmt werden. Ein weiterer Begriff im Zusammenhang mit Geodaten ist die Bounding Box, die mit zwei Koordinatenpunkten (minx,miny) und (maxx,maxy) ein Rechteck beschreibt, das den gewünschten Kartenausschnitt definiert. Die Koordinaten können jedoch nur bezogen auf ihr Projektionssystem eindeutig interpretiert werden. Es existieren viele unterschiedliche Projektionssysteme, in denen versucht wird, die gekrümmte Erdoberfläche mit möglichst geringer Verzerrung auf eine plane Fläche abzubilden. Von der bereits erwähnten Organisation EPSG wurden erstmals alle vorhandenen Koordinatensysteme mit einem eindeutigen Namen, dem sogenannten EPSG-Code oder auch Spatial Reference Identifier (SRID), versehen.

2.3 Webanwendungen zur Darstellung von Geodaten

In diesem Kapitel werden die grundlegenden Konzepte von internetgestützten Geoinformationssystemen beschrieben. Außerdem wird ein kurzer Überblick über freie und kommerzielle Software für die Nutzung und Verbreitung von Geodaten im Internet gegeben.

2.3.1 Geographische Informationssysteme

Ein geographisches Informationssystem (GIS) ist ein Computerprogramm, das raumbezogene Daten verarbeitet (siehe OpenLayers). Es stellt Funktionen bereit, um Abstraktionen von Objekten auf der Erdoberfläche zu erfassen, zu verwalten, auszuwerten und anzuzeigen. Mit Hilfe von GIS können komplexe räumliche Zusammenhänge erkannt werden. Eine klassische Gliederung für GIS ist nach [Bill, R 1999] Erfassung, Verwaltung, Analyse und Präsentation.

Wie in [Ming] beschrieben gibt es 3 Typen von GIS-Architekturen:

- Traditionelles oder monolithisches GIS
- Client/Server GIS
- verteiltes GIS

Das **traditionelle** oder auch **monolithische GIS** ist ein Mainframe oder Standalone Desktop GIS, das aus den Komponenten Benutzerschnittstelle, Dateneingabe und Geodatenbank besteht. Hierbei kann der Nutzer nur auf Daten, die in seinem System abgelegt worden sind, zugreifen. Die Daten waren zumeist in einem proprietären Format abgelegt und für einen Datenaustausch so gut wie gar nicht einsetzbar.

Verbessert worden ist diese Standalone-Lösung durch ein **Client/Server GIS**, dem ein auf LAN basierter Desktop GIS zugrunde liegt. Jedoch ist der einfache Austausch und die gemeinsame Nutzung der Daten durch die unterschiedlichen Formate weiterhin erschwert, so dass webbasierte geographische Informationssysteme entwickelt wurden. Die Internettechnologien sind immer weiter vorangeschritten, so dass eine Vielzahl von Anwendungen entstanden ist, um mit räumlichen Daten unter Nutzung des Webs zu arbeiten. Durch die Bereitstellung im Internet wurden die Daten einer großen Anzahl von Anwendern zur Verfügung gestellt und eine einfache Nutzung gemeinsamer Daten ermöglicht. Es haben sich unterschiedliche Begriffe, wie Web-Mapping, Web-GIS und Internet-GIS, für diese GIS-Anwendungen im Internet etabliert. In dieser

Arbeit wird der Begriff Web-GIS für alle Anwendungen verwendet, die mit geographischen Daten agieren und ihre Informationen über das Internet austauschen. Das Internet basiert auf dem Zusammenspiel von Client und Server.

Gemäß [LeuRei 2000] gibt es zwei verschiedene Arten, um internetbasierte Client/Server-GIS-Anwendungen zu entwickeln, entweder client-seitig oder auf der Seite der Server. Bei einer client-seitigen GIS-Anwendung wird der Client, also der Browser erweitert, um GIS-Funktionalität zu unterstützen. Während in einer server-seitigen GIS-Anwendung ein Browser nur für die Zugriffe auf den Server und die Anzeige der Ergebnisse verwendet wird. Ein Webbrowser ist normalerweise in der Lage, HTML-Seiten, JavaScript und Bilder im Rasterformat zu verarbeiten. Soll der Browser erweitert werden, um GIS-Funktionalitäten und somit auch andere Formate handhaben zu können, müssen zusätzliche Funktionalitäten zur Verfügung gestellt werden. Auf Seiten des Clients gibt es dazu vier Möglichkeiten:

- Plug-Ins: Dies sind Programme, die von Herstellern zur Verfügung gestellt werden, um die Funktionalität von Browsern zu erweitern. Sie müssen auf dem Client installiert werden.
- JavaScript ist eine Programmiersprache, die client-seitig genutzt wird, um Webseiten dynamisch zu manipulieren. Die Anwendungen von JavaScript werden zur Laufzeit vom Webbrowser interpretiert und laufen in einer so genannten "Sandbox". Dies verhindert, dass den Programmen auf dem Rechner, auf dem die Anwendung läuft, kein Schaden zugefügt werden kann.
- ActiveX sind Softwarekomponenten, die von Microsoft für verschiedene Anwendungen entwickelt wurden, unter anderem auch um den Internet Explorer zu erweitern. Auch ActiveX-Komponenten müssen auf dem Client installiert werden.
- Java Applets sind Programme, die in Webseiten ablaufen können und direkt mit dem Benutzer interagieren.

Sowohl Plug-Ins als auch ActiveX müssen auf dem Client installiert werden und bilden somit eine Sicherheitslücke, da der Client der Herkunft der Programme nicht hundertprozentig vertrauen kann.

Auf der Seite des Servers gibt es fünf Möglichkeiten, um GIS-Anwendungen mit dem Internet zu verbinden:

- Common Gateway Interface (CGI) ist ein Interface, um Daten zwischen dem Anwender und dem Server auszutauschen. Auf dem Server werden Programme mit den Anfragedaten ausgeführt und die Antworten zurück an den Anwender geschickt. Eine häufig benutzte Sprache ist hierbei Perl.
- Webserver Application Programming Interface (API) ist eine Schnittstelle, die für die Kommunikation und den Datenaustausch mit einer Webseite zuständig ist. Sie ermöglicht einer Webseite, Informationen bei einer anderen Webseite anzufordern und selbständig in die eigene Webseite einzubauen.
- Java Servlets sind Java-Applikationen, die eine Serverkomponente aufweisen. Java wird durch den Einsatz von plattformabhängigen Java Virtual Machines (JVM) in einen plattformunabhängigen Zwischencode kompiliert. Die Applikationen können dann auf dem Server oder dem Client ausgeführt werden.
- Active Server Pages (ASP) ist von Microsoft entwickelt worden, um Skripte, die in HTML-Seiten eingebettet sind, auf dem Server auszuführen.

- JavaServer Pages (JSP) ist das Pendant von Java zu den Active Server Pages von Microsoft. Das Programm, das in HTML eingebettet ist, wird durch den Server in Java Servlets umgewandelt und dann auf dem Server ausgeführt.

Zusätzlich zum Webserver wird noch ein Mapserver benötigt. Der Internetnutzer stellt eine Anfrage an den Webserver, der die Anfrage an den Mapserver weitergibt. Der Mapserver fragt die gewünschten Daten aus der Geodatenbank ab und erstellt die Antwort, meistens in Form einer Karte. Die Karten werden über den Webserver an den Nutzer zurückgeschickt. Bei jeder Anfrage wird von dem Mapserver eine neue Karte aus den vorhandenen Daten generiert und in eine HTML-Seite eingefügt. Der Webserver dient nur zur Übertragung der Karte zum Browser. Weiterhin stellt ein Mapserver Funktionen zur Verfügung, die für die Navigation in der Kartenapplikation auf dem Client benötigt werden. Auch werden bestimmte Zusatzinformationen über die Karte von dem Mapserver dem Nutzer zugänglich gemacht. Eine Architektur eines Web-GIS ist z. B. folgendermaßen aufgebaut:

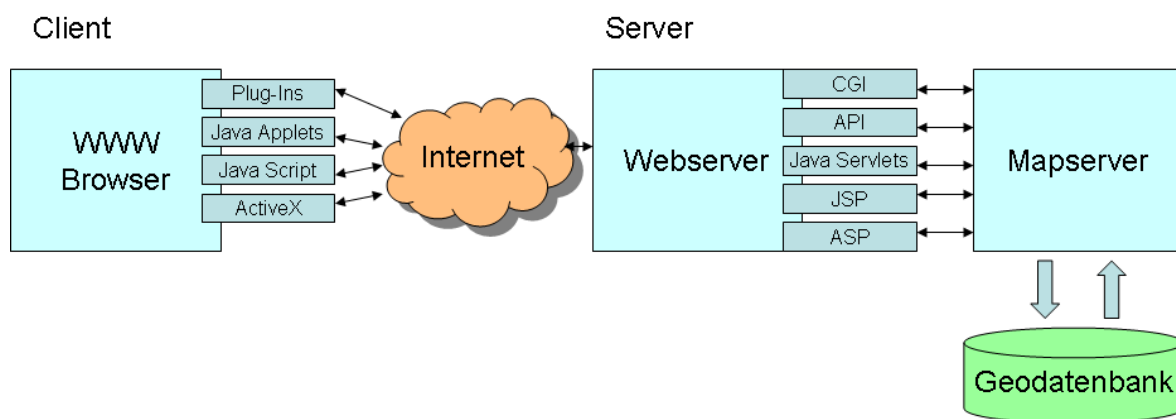


Abbildung 2.7: Architektur eines Web-GIS

Eine Form des **verteilten GIS** sind mobile GIS, bei denen die Clients mobile Geräte z. B. Laptops, Handys oder PDAs sind. Die Kommunikation zwischen Client und Server erfolgt über drahtlose Netzwerke. Aufgrund der verwendeten Clients entstehen Einschränkungen bei der Bildschirmgröße, der Rechnerleistung und den Datenübertragungsraten. Neben den Komponenten Webserver, Applikationsserver, Mapserver und Datenserver ist ein Gatewayserver notwendig, der das mobile Gerät mit dem Webserver verbindet und die Anfrage des Benutzers in HTTP übersetzt.

2.3.2 OGC-Webservice

Immer mehr verbreitet im Internet ist die Nutzung von Diensten. Ein Webservice funktioniert so, dass ein Client eine Internetadresse (URL) aufruft, die die Anfrage entgegennimmt, bearbeitet und die Antwort an den Client zurücksendet. Ein OGC-Webservice ermöglicht insbesondere den Zugriff auf Geodaten in unterschiedlichen geographischen Informationssystemen, indem die Kommunikation zwischen den Systemen standardisiert wird. Geodienste heißen auch OpenGIS Web Services (OWS) oder Geodatendienste. Dies sind Softwareanwendungen, die durch den Austausch von XML-basierten Nachrichten über das Internetprotokoll HTTP mit anderen Softwareanwendungen kommunizieren. Für die Bereitstellung von Bilddaten im Web-GIS gibt es von OGC drei standardisierte Geodienste, den Web Map Service (WMS), den Web Coverage Service (WCS) und den Web Feature Service (WFS).

Web Map Service (WMS) Dieser Service ist ein OGC-Standard zur Darstellung einer Rasterdatei, die über eine Webschnittstelle angefordert werden kann. Sowohl die Syntax der Anfrage als auch das Format der Antwort wird definiert. Bei der Anfrage werden Parameter mitgeliefert, die an den WMS-Server weitergegeben werden. Der WMS-Server wertet die Anfrage aus und greift auf die Geodaten zu, die in mehreren verteilten Geodatenbanken liegen können. An den Client zurückgeliefert wird ein Bild in dem angeforderten Format, zum Beispiel gif, png oder jpeg. [ogc] spezifiziert folgende Operationen:

- GetCapabilities gibt eine XML-Datei mit den Diensten zurück, die der Web Map Service anbietet, und welche Anfrageparameter akzeptiert werden.
- GetMap liefert ein Bild mit den Karteninformationen.
- GetFeatureInfo ruft Informationen über die verfügbaren Layer ab und kann optional verwendet werden.

Web Coverage Service (WCS) Bei einem Coverage handelt es sich um Rasterdaten, die noch zusätzliche Informationen enthalten, oder um multidimensionale Rasterdaten. Im WCS sind folgende Operationen verfügbar:

- GetCapabilities antwortet mit einer kurzen Beschreibung über die Dienste.
- DescribeCoverage liefert eine vollständige Beschreibung eines oder mehrerer Coverages.
- GetCoverage gibt ein Coverage oder mehrere angefragte Coverages in einem bekannten Format zurück.

Web Feature Service (WFS) Der WFS definiert eine Schnittstelle und ihre Operationen für das Lesen und Editieren von geographischen Vektordaten. Die Geodaten liegen in einer GML-Datei vor und können in einem Client angezeigt werden. Die Daten sind nicht wie bei WMS fertige Karten. Vom [ogc] werden folgende Operationen spezifiziert:

- GetCapabilities ermittelt die Features, die zur Verfügung stehen.
- DescribeFeatureType liefert Informationen über die angebotenen Features.
- GetFeature gibt das angefragte Feature an den Client zurück.
- Transaction: Mit Transaction WFS oder auch Web Feature Service Transaction (WFS-T) kann der Client lesend und schreibend auf die Features zugreifen und sogar Features löschen. Diese Funktionalität wird nicht von allen Servern unterstützt.

Aus der Vielzahl der Entwicklungen, die sowohl client- als auch server-seitig getätigt wurden, wird auf die verbreitesten im Folgenden näher eingehen. Als Web-GIS Server werden GeoServer und MapServer, als Web-GIS-Client OpenLayers und Mapbender und zum Abschluss die zwei kommerziellen Web-GIS-Anwendungen ArcIMS und Google Earth vorgestellt.

2.3.3 UMN MapServer

UMN MapServer ist ein Kartenserver, der Mitte der 90er mit der Unterstützung der NASA an der Universität von Minnesota (UMN) im Projekt ForNet entwickelt wurde. 1994 wurde die erste Version von UMN MapServer veröffentlicht und danach von dem NASA-Projekt TerraSIP betreut. Seit 2007 gehört UMN MapServer zu den OSGeo-Projekten und wird von dem OSGeo MapServer Project Steering Committee und mehreren Entwicklern weltweit betreut. Die

Hauptanwendung von MapServer besteht darin, dynamische Karten über das Internet anzuzeigen, wobei viele Datenformate benutzt werden können. MapServer ist eine Open Source Software, die interaktiv in einem Webserver eingebunden wird. Standardmäßig führt UMN MapServer ein CGI-Programm in der Schnittstelle des Webserver aus. Der Benutzer stellt eine kartographische Anfrage an den Webserver, die an den MapServer weitergeleitet wird. Der MapServer benutzt die Parameter aus der URL-Anfrage und die Konfigurationen aus einer Konfigurationsdatei, der sogenannten MAP-Datei, um ein Bild gemäß den angeforderten Daten zu erstellen. Die MAP-Datei ist die zentrale Konfigurationsdatei des UMN MapServers, mit der die kartographische Darstellung und die Kartenausgabe bestimmt wird. Sie enthält Informationen über die einzelnen Layers, deren Projektion, welche Dienste angeboten werden, die verwendete Datenquelle, und unter Umständen werden Klassifizierungen und Ausgestaltungen in dieser Datei definiert. Die MAP-Datei besteht aus mehreren Abschnitten, die mit einem Schlüsselwort beginnen und mit einem END abgeschlossen werden. Ausgenommen von dieser Regel ist der Header, der kein einleitendes Schlüsselwort besitzt, jedoch auch mit einem END abgeschlossen wird. Der Header enthält Angaben, die für das gesamte Projekt Gültigkeit haben und somit auch das Aussehen der Karte bestimmen. Im Abschnitt Web werden der Web Map Service (WMS) und verschiedene Templates definiert. Bei den Templates handelt es sich um HTML-Dateien, die das Aussehen der Benutzeroberfläche festlegen. Anschließend kommen Angaben zur Projektion der erstellten Karte und Definitionen für Maßstabsleiste, Zeichenerklärungen und einzelner Kartenschichten (Layer). Da diese Konfigurationsdatei einfach zu handhaben ist, kann sie auch von Anfängern erstellt werden, um somit schwierige Kartengestaltungen zu realisieren. Um die Karte zu erstellen, greift der MapServer auf lokale oder über das Netz verfügbare Geodaten zu. Die Geodaten können hierbei in einem Dateisystem oder einer Geodatenbank abgelegt werden, wobei die Geodatenbank gegenüber einem Dateisystem einen erheblichen Vorteil bezüglich Performance liefert. Die so generierte Karte wird über das Internet zurück an den Webserver und von ihm an den Webbrowser geliefert. Wie oben erwähnt wird in der MAP-Datei ein HTML-Template definiert, das das Aussehen der Benutzeroberfläche im Browser festlegt. Für jede Anfrage, also für jeden Klick auf die Karte, werden alle Parameter aus der MAP-Datei an den MapServer geschickt. Dieser erzeugt temporäre Karten, Legende, Maßstabsleiste und Übersichtskarten und setzt diese in das Template an der vorher festgelegten Stelle ein. In der MAP-Datei wird auch festgelegt, ob Abfragen auf einen Layer zugelassen sind oder nicht. Ist die URL des UMN MapServer bekannt, kann dieser weltweit aufgerufen werden. Der UMN MapServer liegt sowohl für Windows also auch für Linux vor. Um ein Arbeiten mit dem UMN MapServer überhaupt möglich zu machen, muss zunächst einmal ein Webserver, zum Beispiel Apache, und eine PHP-Version installiert werden. PHP ist eine Skriptsprache, mit der dynamische Webseiten in einem Browser angezeigt werden können. [MapServ]

2.3.4 GeoServer

Das Projekt GeoServer wurde 2001 von The Open Planning Project (TOPP) begonnen. TOPP ist eine Organisation, die gegründet wurde, um die Zusammenarbeit der Regierung mit der Zivilbevölkerung zu fördern und somit nützliche Informationen öffentlich zugänglich zu machen und Entscheidungen im gemeinsamen Einverständnis zu treffen. [p2pfoun]

Den Gründern von GeoServer schwebte ein Geospatial Web analog zum World Wide Web vor. Mit dem World Wide Web lassen sich beliebige Texte suchen und downloaden, mit dem Geospatial Web sollte dies für Geodaten möglich sein. Die Anbieter der Daten sollten in der Lage sein, ihre Daten direkt im Web zu veröffentlichen, und Nutzer sollten direkt darauf zugreifen können. Bei der Entwicklung von GeoServer wurde viel Wert auf offene Standards gelegt, so dass GeoServer ein standardkonformer und funktionsreicher Open-Source-Server für Web-GIS-Anwendungen geworden ist. Der Server ist in Java programmiert und baut auf GeoTools auf, das eine Open

Source Java-Code-Bibliothek ist, die standardkonforme Methoden anbietet, um mit Geodaten zu arbeiten. Jegliche Konfiguration erfolgt im GeoServer über ein Web Administration Tool. Der Zugriff auf das Tool erfolgt über einen Webbrowser anhand der Standardinstallation auf dem lokalen Rechner mit der URL `http://localhost:8080/geoserver/web`. Möchte der Anwender Änderungen in der Konfiguration durchführen, muss er sich zuerst bei GeoServer anmelden. Der Standardnutzer ist `admin`, das zugehörige Passwort kann nur in der Datei `security/users.properties` in dem Dateisystem von GeoServer geändert werden. Nach dem Einloggen hat man Zugriff auf verschiedene Bereiche. Im Bereich "Server" erhält man einen Überblick über den Serverstatus, Kontaktinformationen, globale Einstellungen und Einstellungen für WMS und WCS. Der Abschnitt "Service" ist für fortgeschrittene Anwender, um Abfragefunktionen von WMS, WFS und WCS festzulegen. Über "Layer Preview" lassen sich die Layer in verschiedenen Ausgabeformaten ansehen. Etwa zur gleichen Zeit als GeoServer gegründet wurde, arbeitete das OpenGIS Consortium, das heute unter dem Namen Open Geospatial Consortium (OGC) bekannt ist, an einem Standard für einen Web Feature Service. Hiermit wurde ein Standardverfahren entwickelt, um direkt auf Geodaten im Web zugreifen zu können, wobei Geographic Markup Language (GML) als Datenaustauschformat verwendet wird. Ebenfalls wurde ein Web Map Service erstellt, der als ein Standard für die Erstellung und das Anzeigen von Karten mit Geodaten gilt. Zudem erstellte MetaCarta OpenLayers (siehe [openLayers]). Insgesamt haben all diese Entwicklungen die Funktionalitäten von GeoServer sehr verbessert. GeoServer kann nun Ausgangsdaten generieren, die in vielen Internetbrowsern für Geodaten, zum Beispiel Google Earth [googleearth], angezeigt werden können. Darüber hinaus arbeitet GeoServer derzeit direkt mit Google zusammen, damit Daten von GeoServer in Google Maps gesucht werden können. [GeoservHis]

2.3.5 Mapbender

Mapbender wurde 2001 von der Firma CC-GIS mit dem Produktnamen CCGIS Client Suite entwickelt. 2003 wurde die Software mit dem Namen Mapbender unter der GNU GPL License frei zugänglich gemacht und bei SourceForge registriert. Im Jahre 2006 gehörte Mapbender zu einem der acht Startprojekte der Open Source Geospatial Foundation (OSGeo). Die derzeitigen Weiterentwicklungen werden von WhereGroup durchgeführt, die aktuelle (Stand 17. August 2011) Mapbender Version ist 2.7. Mapbender unterstützt Kartenanwendungen, die in PHP, JavaScript und XML implementiert sind, wobei Mapbender nur die Karten verwalten und darstellen kann. Die Karten selbst werden von unterschiedlichen Kartendiensten, die über das Internet erreichbar sind, zur Verfügung gestellt. In Mapbender existieren drei zentrale Objekte:

- Benutzer
- Oberfläche (Applikation)
- Karten- und Datendienste

Um mit Mapbender arbeiten zu können muss ein Benutzer zuerst über die flexible Benutzerverwaltung angelegt werden. Hierbei können dem Benutzer unterschiedliche Rechte für die Dienste und Applikationen zugeteilt werden. Ein Benutzer, der nur Anwenderrechte hat, kann über die Kartenoberfläche Ebenen ein- und ausblenden oder in die Karte hinein und heraus zoomen. Ein Benutzer mit Rechten für eine Oberfläche kann auf den Kartendienst zugreifen und somit unterschiedliche Dienste anfordern. Benutzer können auch in Gruppen zusammengefasst werden, wobei einer Gruppe eine Applikation zugeordnet werden kann. Mit dieser Zuordnung erhalten alle Benutzer der Gruppe Zugriff auf die Dienste dieser Oberfläche. Mapbender besitzt vordefinierte Applikationen, die eine Abfrage mittels OGC-Standarddiensten und eine einfache Navigation ermöglichen. Mit Hilfe dieser Dienste kann die Oberfläche individuell, je nach Problemstellung konfiguriert werden. Die Verwaltung der Nutzer obliegt einem Administrator. Als

Oberfläche wird eine HTML-Seite mit Steuerelementen bezeichnet, wobei jede Oberfläche individuell zusammengestellt wird. Mapbender verfügt über einige Template-Applikationen, die als Vorlagen für die Generierung von eigenen Applikationen dienen können. Jegliche Konfiguration von Mapbender wird über die Oberfläche eingestellt, die Oberfläche dient sozusagen als Bindeglied zwischen Benutzer und Kartendienst. Als Kartendienst oder Datendienst kann jede Datenquelle eingebunden werden, die mit WMS und WFS kompatibel ist. Zusätzlich lässt sich Mapbender als Open Source Software flexibel erweitern, damit andere Nicht-Standarddienste eingebunden werden können. Die Karten werden dynamisch zur Laufzeit aus einer Datenbank gelesen. Für den Betrieb und die Verwaltung ist keine Installation auf dem lokalen Rechner notwendig, ein Webbrowser mit Internet-Zugang reicht völlig aus, um mit Mapbender zu arbeiten. Die Softwarevoraussetzungen sind ein Webserver (z. B. Apache), PHP als Skriptsprache und eine Datenbank. Mittels der Datenbank können alle Aktionen, zum Beispiel Anlegen von Benutzerkonten, protokolliert werden.

2.3.6 OpenLayers

Für Web-Entwicklungen, bei denen Kartendaten in die Anwendung eingebunden werden, ist OpenLayers mittlerweile die meist benutzte Software. OpenLayers ist eine JavaScript-Bibliothek, mit der interaktive Karten in Webseiten integriert werden können, ohne mit dem Server in Abhängigkeit zu stehen. Auch hierbei handelt es sich um eine Open Source Software, die kostenlos von der Homepage von OpenLayers heruntergeladen werden kann. Die Firma MetaCarta entwickelte im Juni 2006 den ersten Prototyp von OpenLayers, um dann ca. 1 Jahr später die erste Version 1.0 zu veröffentlichen. Aktuell ist die Version 2.10 verfügbar. Seit November 2007 gehört OpenLayers der Open Source Geospatial Foundation (OSGeo) an und wird mittlerweile durch eine Reihe von Organisationen auf der ganzen Erde unterstützt. In der HTML-Seite, in der OpenLayers genutzt werden soll, muss die Java-Bibliothek geladen werden, indem man im `<head>`-Teil das `<script>`-Element auf die URL `http://openlayers.org/api/OpenLayers.js` verweisen lässt. Man kann auch OpenLayers aus dem eigenen lokalen Webserver einbinden. Hierfür muss OpenLayers direkt auf den lokalen Server in ein lokales Verzeichnis heruntergeladen werden. In dem zentralen JavaScript-Objekt OpenLayers ist jeder Methodenaufruf, den OpenLayers bereitstellt, enthalten. Der Ablauf jeder Anwendung von OpenLayers ist untergliedert in:

- Anlegen eines Map-Objektes
- Erzeugen von mindestens einem Layer-Objekt
- Hinzufügen der Layer zur Map

Damit man mit dem Layer sinnvoll arbeiten kann, benötigt man noch ein Control-Objekt. Das Map-Objekt enthält Referenzen auf alle Layer- und Control-Objekte und ist die Zentrale der Webanwendung. Die Layer-Objekte enthalten die Themen der Karte, wobei ein Layer-Objekt aus einem Thema oder einer Ansammlung von Layern bestehen kann. Die Control-Objekte sind die Steuerelemente, mit denen man in der Karte navigieren kann und bestimmt wird, wie auf Mausaktionen in der Karte reagiert werden soll. Es können sehr einfach zusätzliche Layer, wie Textlayer, Imagelayer, Vektorlayer oder auch Markers und Boxlayer in die Darstellung der Karte aufgenommen werden. In OpenLayers wird zur Umrechnung von Projektionen eine kartographische Projektionsbibliothek (Proj4) zur Verfügung gestellt. Unterstützt werden verschiedene Standardschnittstellen von OGC, wie Web Map Service (WMS) und Web Feature Service (WFS). OpenLayers ist ein Ajax-Toolkit, dadurch können Daten zwischen Server und Browser asynchron übertragen werden. Es ermöglicht, HTTP-Anfragen an den Server durchzuführen, die Ergebnisse an den Browser zurückzugeben, ohne die komplette Webseite neu zu laden. Damit wird das Konzept von OpenLayers unterstützt, die geographischen Daten und die Anwendungen zur

Darstellung dieser Daten zu trennen. Die einfache Bedienbarkeit und die Interoperabilität mit verschiedenen Datenquellen macht OpenLayers zu einem sehr beliebten Client, um Karten im Web zu visualisieren und mit dem Server zu interagieren. [wikiOpen, osOpen]

2.3.7 ArcIMS

ArcIMS ist eine kommerzielle Software, die von der Firma ESRI entwickelt worden ist. Es ist wohl das geographische Informationssystem, das weltweit am häufigsten verkauft und eingesetzt wird. Es ist relativ einfach zu bedienen und besitzt vordefinierte Anwendungen, die die Zugriffe auf die Daten sehr vereinfachen. Das Publizieren der Daten erfolgt mit einem Map Server. Dieser besteht aus zwei verteilten Komponenten, dem Client und dem Server. Der Client besteht aus einem Viewer. Der Server umfasst mehrere Komponenten, einen Spatial Server, einen Application Server, die Application Server Connectors und den Manager. Weiterhin besteht der Map Server aus einem Webserver, einem Browser und einem Servlet (z. B. Tomcat Apache). Bei einer ArcIMS-Anfrage wird diese zuerst vom Webserver entgegengenommen und dann mittels der Konnektoren zu einem Application Server weitergeleitet. Der Application Server schickt die Anfrage zur Verarbeitung an den Spatial Server weiter. Die Kommunikation zwischen Application Server und Spatial Server erfolgt mittels ArcXML, das eine Implementierung von XML ist, die innerhalb von ArcIMS benutzt wird. Die Antwort an den Client erfolgt einfach in die entgegengesetzte Richtung. Abbildung 2.8 zeigt den Ablauf zur Verarbeitung von Client-Anfragen innerhalb von ArcIMS.

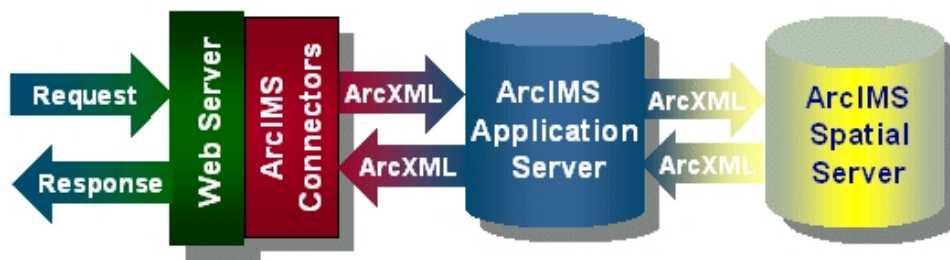


Abbildung 2.8: ArcIMS Ablauf (Quelle: [esriSup])

Der Spatial Server ist das Kernstück des ArcIMS-Map-Servers und für die Generierung von Karten und Bereitstellung von zusätzlichen Daten zuständig. Die Karten und die Daten können in unterschiedlichen Arten zum Client übertragen werden. Hierfür stehen innerhalb des Spatial Server verschiedene Dienste zur Verfügung:

- Image extrahiert und produziert die Karte im JPG-, GIF- und PNG-Format.
- Feature liefert die Rohdaten in sehr stark komprimierter binärer Form. Es liegt beim Client, diese zu entpacken und zu speichern.
- Query liefert die Daten aus der Datenquelle.
- Geocode findet basierend auf Informationen aus Shapefiles oder ArcSDE Adressen auf Karten.
- Extract generiert einen Ausschnitt aus Shapefiles der ArcSDE-Daten als Shapefile.

Der Application Server übernimmt die Lastverteilung der eingehenden Anfragen. Hierfür dokumentiert er, welche Dienste auf welchem Spatial Server ausgeführt werden und ordnet die

Anfragen dem zuständigen Dienst zu. Wie oben erwähnt erfolgt die Kommunikation zwischen Application Server und Spatial Server mit Hilfe von ArcXML. Die Anfragen, die über den Webserver eingehen, müssen deswegen in ArcXML übersetzt werden. Diese Übersetzung wird von speziellen Konnektoren getätigt. Es gibt in ArcIMS vier verschiedene Konnektoren für die unterschiedlichen Clients, den Servlet Connector, den Cold Fusion Connector, den ActiveX Connector und den Java Connector. Der Manager umfasst drei Komponenten, die für den Zugriff auf alle Server-Einstellungen zuständig sind. Der ArcIMS Author erstellt die Konfigurationsdateien für den Map-Service im ArcXML-Format. Der ArcIMS Designer erstellt die Webseiten und der ArcIMS Administrator verwaltet den Spatial Server und veröffentlicht den Webservice. Auf der Seite des Clients kann bei ArcIMS zwischen einem HTML-Viewer und einem Java-Viewer gewählt werden. [ESRI,esrihelp]

2.3.8 Google Earth

Google Earth basiert auf dem Programm Keyhole, das von dem Unternehmen Keyhole Corp. von Michael T. Jones, dem Gründer der Firma, entwickelt wurde. Als 2004 das Unternehmen von Google Inc. übernommen wurde, wurde das Programm in Google Earth umbenannt. [wikearth] Bei der Entwicklung von Google Earth hat sich in den letzten Jahren viel getan, und die Attraktivität von Google Earth steigt zunehmend. [Gros 2006] nennt die Gründe für die Beliebtheit von Google Earth gegenüber anderen vergleichbaren Programmen wie World Wind von der NASA oder GeoPlayer von GeoFusion: "It has captured an enormous interest for a few key reasons: (1) it is free; (2) it is fast; (3) it has its own markup language (KML), which allows anyone to display and easily share their own data; and (4) it is by all accounts fun; this stems from its speed, an easy-to-use interface and its high quality imagery." [Gros 2006] [Höf 2009] untersucht die Vorteile von Google Earth gegenüber anderen Virtual Globes genauer und kommt zum Schluss, dass die Hemmschwelle für den Einstieg mit der Arbeit von Google Earth durch die hohe Benutzerfreundlichkeit sehr gering ist. Die Nutzung der Daten ist für den privaten Benutzer hervorragend, da sie leicht zugänglich sind, einen großen Umfang haben und leicht zu visualisieren sind. Es gibt eine große Anzahl von freiwilligen Benutzern, die nützliche Zusatzinformationen zur Verfügung stellen und somit Google Earth für eine breite Masse mit einem nicht unerheblichen Mehrwert ausstatten. In vielen verschiedenen Fachzeitschriften als auch in Wochenzeitschriften findet man Artikel über Themen bezüglich Google Earth, womit der Bekanntheitsgrad und auch die Beliebtheit von Google Earth feststellbar ist. Google Earth ist eine Server-Client-Anwendung. Google Earth 6 steht in drei unterschiedlichen Versionen zur Verfügung: [googEarth]

- Google Earth, dies ist die kostenlose Basisversion.
- Google Earth Pro für 399 US-Dollar bietet zusätzlich zur Basisversion professionelle Tools, die speziell für Geschäftskunden entwickelt wurden.
- Google Earth Enterprise bietet Unternehmen die Möglichkeit, ihre eigenen Bilddaten und andere geographische Daten ihren Mitarbeitern zugänglich zu machen, wobei dies eine umfangreiche Server-Version und keine Client-Version wie die beiden anderen Versionen ist.

"Bei Aufbau einer Datenbank der gesamten Erde ist es am praktischsten, nur eine einzige globale Projektion zu verwenden. Google Earth verwendet für seine Bilddatenbank eine einfache Zylinderprojektion. Es handelt sich dabei um eine einfache Kartenprojektion, bei der Längen- und Breitengrade abstandstreue, gerade parallele Linien sind, die sich jeweils im rechten Winkel kreuzen. Diese Projektion wird auch als Lat/Lon-WGS84-Projektion bezeichnet." [earthsup]

Als Sprache verwendet Google Earth die Keyhole Markup Language (KML). KML ist eine XML-Sprache, die auf geographische Visualisierung einschließlich der Annotation von Karten und

Bildern konzentriert ist. Geographische Visualisierung umfasst hierbei nicht nur die Präsentation von geographischen Daten auf dem Globus, sondern auch die Kontrolle über die Navigation des Benutzers im Sinne von "wo geht es lang" und "wo muss man suchen". [ogckml] Heutzutage werden viele Daten in Form einer KMZ-Datei, das heißt einer komprimierten KML-Datei zur Verfügung gestellt. Mittels KML bietet Google Earth einige Funktionalitäten an, für deren genaue Beschreibung wegen der Vielzahl der Funktionalitäten auf [earthuserman] verwiesen wird. Zur Visualisierung benutzt Google Earth Raster- oder Vektordaten. Die Rasterdaten können als Image-Overlay in Google Earth angezeigt werden. Vektordaten können mit Drag-and-Drop oder durch das Untermenü *Öffnen* oder *Importieren* im Menü *Datei* importiert werden. Integrierte Daten sind Daten, die von Google Earth auf den Server der Firma hochgeladen werden, um von dort auf die Daten zugreifen zu können. Des Weiteren gibt es importierte Daten, die der Nutzer mit Hilfe der Programmiersprache KML selbst erstellen kann. Diese Daten werden auf dem Rechner des Nutzers gespeichert und können mit der Importfunktion in Google Earth geöffnet werden. Alles in allem ist Google Earth ein sehr erfolgreiches Programm. Seine Vorteile liegen vor allem in der einfachen Bedienbarkeit, dem weltweiten Raumbezug, der globale Daten zur Verfügung stellt und dadurch internationale Vergleiche möglich macht. Da es auf das Internet aufbaut, können eine große Anzahl von Benutzern angesprochen werden. Die Qualität als auch die Quantität der Daten wird durch die aktive Community erhöht. Jedoch sollte man nicht aus den Augen verlieren, dass durch die Monopolstellung von Google Earth und dem Mangel der Transparenz Google Inc. allein über die weiteren Entwicklungen und technischen Neuerungen entscheiden kann. [Höf 2009]

Kapitel 3

Entwurf

Wie im Kapitel Grundlagen im Abschnitt 2.1 TanDEM-X-Mission beschrieben werden bei der TanDEM-X-Mission zwei räumlich versetzte Antennen eingesetzt. Diese Anordnung ist notwendig, um eine perspektivische Sicht auf die Erde zu erhalten. Die empfangenen Radarsignale beider Antennen liefern hochaufgelöste Bilddaten, die interferometrisch ausgewertet werden, wobei je Antenne mindestens ein Radarbild desselben Gebiets erforderlich ist (siehe Abbildung 3.1). Die Radarbilder werden zu einem Interferogramm übereinandergelagert, indem die Phasenwerte - die Amplituden - der zurückgestreuten Radarsignale addiert werden und ein Interferenzmuster ergeben. Die Phasendifferenzen werden durch die Aufnahmegeometrie (die Anordnung und Ausrichtung der Antennen), die Bodenbeschaffenheit sowie durch Störeinflüsse bestimmt. Um Störeinflüsse möglichst vollständig zu beseitigen und aus dem Interferogramm ein digitales Geländemodell (DEM) zu erzeugen, wird eine Referenzhöhe in die Berechnungen einbezogen.

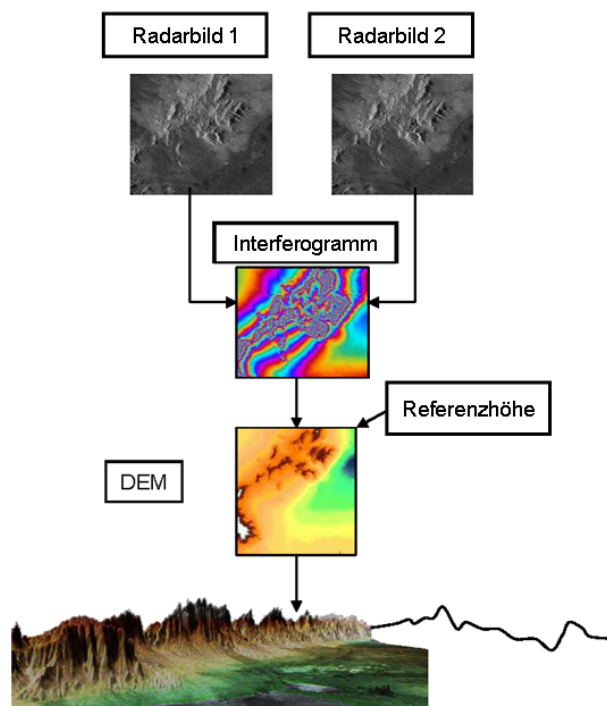


Abbildung 3.1: DEM-Generierung (Quelle: DLR)

Bei der Verarbeitung der Bilddaten ergeben sich Informationen, die bei der Auswertung der Daten benötigt werden. Zum einen sind dies Eingabedaten über den Aufnahmemodus und die Qualität der Aufnahme, zum anderen unterschiedliche Parameter, die bei der Verarbeitung verwendet wurden. Die Gesamtheit dieser Daten soll Wissenschaftlern nun zugänglich gemacht werden.

3.1 Stand der Technik

Der Vorteil der Visualisierung der Daten gleich nach der Verarbeitung ist schon frühzeitig nach dem Start der TanDEM-X-Mission erkannt worden. TanDEM-X ist ein großes Softwaresystem, das von mehreren heterogenen Teams entwickelt wurde und wird. Damit die Mission erfolgreich durchgeführt werden kann, muss fachübergreifendes Wissen zwischen den einzelnen Teams ausgetauscht werden. Eine erste Auswertung der Daten befindet sich am Anfang der gesamten Kette. Die graphische Darstellung dieser Auswertungen ist schon in diesem Stadium äußerst wichtig, um eine schnelle Rückmeldung und eine Beurteilung der Daten zu liefern. Das Projekt TanDEM-X wird unter einem sehr großen Zeitdruck entwickelt. Der TanDEM-X-Satellit sollte ursprünglich zeitgleich mit seinem Zwillingssatelliten TerraSAR-X im All arbeiten, denn die Mission ist auf Grundlage von beiden Satelliten ausgelegt. Sie besitzen eine Lebensdauer von 5 Jahren, haben aber weitere Reserven für bis zu 7 Jahre. 3 Jahre lang sollen sie in Formation fliegen. Mit dem Start der TanDEM-X-Mission im Juni 2010 wird der TerraSAR-X-Satellit bereits ein Jahr über seine garantierte Lebensdauer hinaus für die Mission benötigt. Ein weiterer Zeitdruck entstand durch den Start der Trägerrakete. Beide Satelliten wurden mit je einer Dnepr-Rakete vom kasachischen Weltraumbahnhof Baikonur gestartet. Wie man sich vorstellen kann, ist eine Raketenreservierung nicht einfach über ein Reisebüro zu buchen, sondern muss frühzeitig angekündigt werden. Das Zeitfenster für den Transport des TanDEM-X ins All wurde folglich festgelegt und musste wahrgenommen werden, um nicht mehrere Monate auf den nächsten Start warten zu müssen. Bestimmt durch diese zwei Bedingungen wurden viele Entwicklungen, die auch nach dem Satellitenstart am Boden getätigt werden konnten, im Zeitplan nach hinten verschoben. Die Weiterentwicklung der Software zur Verarbeitung der Daten und Aufrechterhaltung des Betriebs hatte Vorrang, so dass bisher ein sehr rudimentäres System für eine Qualitätsbetrachtung der Daten entstanden ist.

Diese Arbeit konzentriert sich auf die Betrachtung der Daten im System des integrierten TanDEM-X-Prozessors ITP. Die für die Verarbeitung der Daten wichtigen Parameter werden in sogenannte Parameter-Pool-Dateien geschrieben. Dies sind ASCII-Dateien, die einfach lesbar sind. Parameter, die zwischen unterschiedlichen Systemen ausgetauscht werden, sind in XML-Dateien gespeichert. Bei der Generierung des digitalen Geländemodells werden unter anderem auch Quicklooks erzeugt. Dies sind Übersichtsbilder von Satellitendaten, die eine geringere Bildauflösung oder Farbtiefe im Gegensatz zu den Originaldaten besitzen und somit weniger Speicherplatz verbrauchen. Darin lassen sich in einem schnellen Überblick unter anderem die Abdeckung des Gebiets und Störungen bei der Aufnahme erkennen. Die Parameter-Pool-Dateien sowie die XML-Dateien werden auch für das System zur Qualitätsbetrachtung der Daten verwendet, das Wissenschaftlern die relevanten Informationen aus diesen Dateien über das Web im internen Netz zur Analyse bereitstellt. Wie in Kapitel Grundlagen im Abschnitt 2.1 TanDEM-X-Mission erwähnt wird der ITP in die drei Verarbeitungsschritte PRCSER, IQPC und RDEMGEN unterteilt. Für jedes dieser Szenarien sind folgende Shell-Skripte zur Generierung der unterschiedlichen Internetseiten im HTML-Format vorhanden:

- `makeHTMLreport.sh`: Dieses Shell-Skript generiert für eine bestimmte Akquisition eine HTML-Seite, die Reportseite genannt wird. Die Webseite enthält eine Übersicht über die

Aufnahmeparameter, qualitätsrelevante Parameter, protokollierte Ausgabewerte der Programme, Quicklooks, Statistiken und Übersichtskarten von dem aufgenommenen Gebiet.

- `makeDailyReportHTML.sh`: Dieses Skript baut für alle Akquisitionen, die an einem bestimmten Tag verarbeitet wurden, die Übersichtsseite auf. Von dieser Übersichtsseite gelangt man auf die HTML-Seite der einzelnen Akquisitionen.
- `makeAllDay.sh` Dieses Shell-Skript baut zuerst für jede Akquisition die zugehörige Reportseite auf, indem es für alle Akquisitionen an diesem Tag das Skript `makeHTMLreport.sh` ausführt. Danach wird für die Übersichtsseite das Skript `makeDailyReportHTML.sh` ausgeführt.
- `makeAllYear.sh`: Dieses Skript führt für jeden Tag in einem Jahr das Skript `makeAllDay.sh` aus und generiert somit eine Übersichtsseite und Reportseiten für alle Akquisitionen, die in einem Jahr verarbeitet wurden.
- `makeAll.sh`: Dieses Skript führt für die Jahre 2010-2019 das Skript `makeAllYear.sh` aus und generiert somit alle HTML-Seiten der Daten, die innerhalb 2010 und 2019 verarbeitet wurden.

Diese Skripte müssen alle manuell gestartet werden.

3.2 Anforderungen

Das bisherige Verfahren erzeugt zwar dynamisch die HTML-Seiten, diese sind dann aber statisch und nicht mehr veränderbar. Eine der Anforderungen ist eine dynamische Darstellung der Ergebnisse. Der aufzubauende Webservice soll die Analyse von Parametern in der TanDEM-X-Mission ermöglichen. Hierfür werden mit unterschiedlichen Verarbeitungstechniken unterschiedliche charakteristische Eigenschaften der Eingabewerte ausgewertet. Dieses System soll aus einer Datenbank und einer Weboberfläche bestehen, die den Benutzern die Möglichkeit gibt, Anfragen auf die Daten dieser Datenbank zu tätigen und die Ergebnisse in der Weboberfläche anzuzeigen. Die Daten sind Verarbeitungsparameter, Verarbeitungstechniken und die Ergebnisse nach der Verarbeitung mit deren Eingabewerten. Alle diese Daten haben einen geographischen Bezug, es handelt sich also um Geodaten. Es sollen unterschiedliche Abfragemöglichkeiten bereitstehen, wie geographische Abfragen, und Geodaten sollen visualisiert werden können. Die Software soll für die Benutzer transparent sein. Es soll auf jeden Parameterwert abgefragt werden können. Darüber hinaus sind statistische Auswertungen wünschenswert. Die dargestellten Karten sollten überschaubar sein und die für die Auswertung relevanten Informationen bereitstellen. Das Einfügen der TanDEM-X-Daten in die Datenbank soll den bereits existierenden laufenden Betrieb nicht beeinflussen oder stören.

3.3 Kriterien für die Auswahl der Software

Diese Bachelorarbeit wird mit Hilfe der Arbeitsgruppe DFD-IT in dem Bereich "Geoservice" implementiert. Diese Arbeitsgruppe entwickelt im DFD einen Service für den Zugang und der Darstellung von Geodaten. Die Auswahlkriterien der Software sind bereits vorgegeben. Die verwendete Softwarekomponente heißt GDAS und bietet einen standardisierten Zugang zu geokodierten Raster- und Vektordaten. Der Zugang und die Darstellung der Daten erfolgt auf Basis der OGC-Schnittstellen WMS, WCS und WFS.

Kapitel 4

Implementierung

Der herzustellende Webservice besteht aus mehreren Komponenten. In der Datenbank werden die Daten mit räumlichem Bezug gespeichert. Hierfür wird die objektrelationale Datenbank PostgreSQL mit ihrer Erweiterung PostGIS verwendet. Der Mapserver verarbeitet die Anfragen vom Client, ruft die Daten vom DBMS ab und generiert WFS/OGC-konforme Ausgaben. Als Mapserver wird "GeoServer" benutzt. In der Webanwendung werden die WFS-Anfragen generiert und die Geodaten mit "OpenLayers" visualisiert.

4.1 Logische Struktur

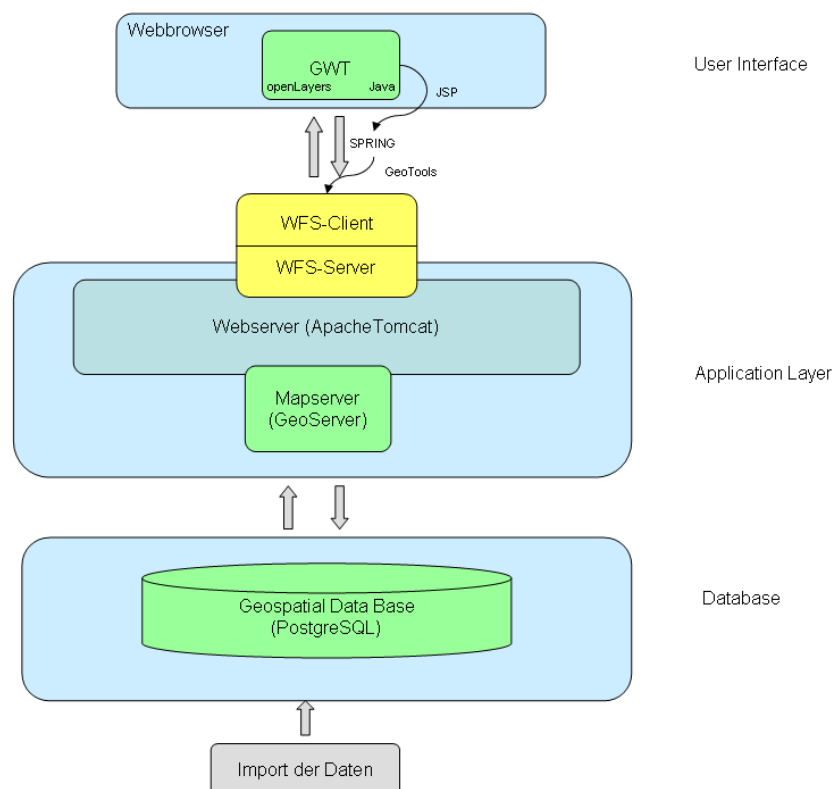


Abbildung 4.1: Logische Struktur

Abbildung 4.1 zeigt die logische Struktur des Webservice. Zunächst werden die aus der TanDEM-X-Mission erhaltenen Daten in die PostgreSQL-Datenbank eingefügt. Auf die Geo-

daten kann nun über den GeoServer zugegriffen werden. Der internetgestützte Zugriff auf Seiten des Nutzers erfolgt über WFS. Der Nutzer startet in einem Browser die Webanwendung, die mit Hilfe von GWT entwickelt wird. Als Framework wird Spring verwendet. Die Darstellung der geographischen Daten erfolgt mit Hilfe von openLayers. GeoTools wird für die Erstellung von WFS-Anfragen auf die Datenbank eingesetzt. Des Weiteren wird für die Erzeugung der dynamischen HTML-Seiten die Programmiersprache JavaServer Pages (JSP) verwendet. Der entwickelte Java-Code wird mit Apache Tomcat auf dem Webserver ausgeführt. Die Anfragen auf die Geodatenbank erfolgt über den Mapserver. Das Ergebnis ist eine interaktive Webanwendung, die auf HTML, Cascading Style Sheets (CSS) und JavaScript basiert. Aus Sicht der Nutzer wird nur ein Standard-Webbrowser ohne zusätzliche Plugins benötigt.

4.2 Physikalische Struktur

4.2.1 Hardware

Integrated TanDEM-X-Prozessor Das System für den ITP besteht aus 20 Sun X4600 Verarbeitungsrechnern und 2 Sun T5240 Kontrollrechnern.

Rechner für GDAS

- 2 x Sun Blade Center 6000
- 4 x X6240 Blades à 2 x 4 Core AMD, 3 Ghz, 64GB

Die Architektur der Hardware im GDAS ist in Abbildung 4.2 dargestellt.

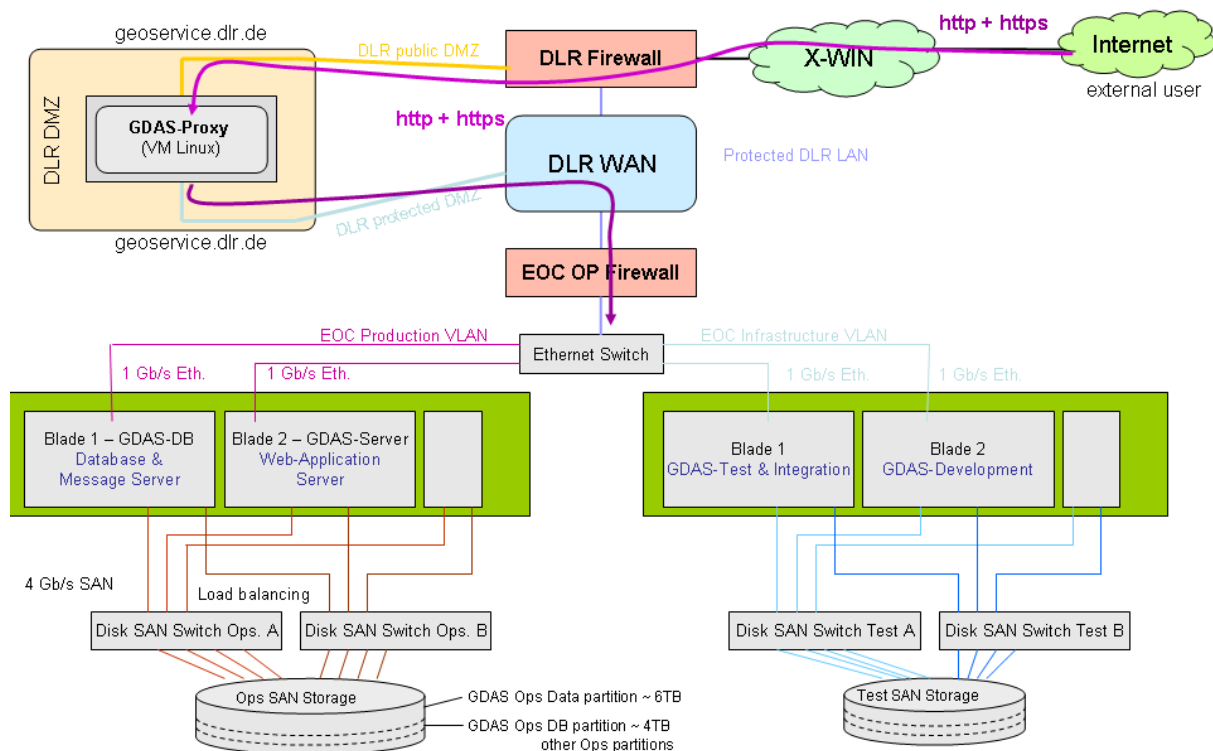


Abbildung 4.2: GDAS Hardware (Quelle: DLR)

4.2.2 Software

Für die Implementierung des Webservice kommt eine breite Palette von Software und Entwicklungstools zum Einsatz. Um den Rahmen der Bachelorarbeit nicht zu sprengen, werden hier die einzelnen Softwarekomponenten aufgelistet und kurz erläutert. Für eine tiefere Betrachtung wird der Leser gebeten, dies unabhängig von dieser Arbeit zu tun.

GeoServer Wie im Kapitel 2.3.4 beschrieben ist GeoServer ein Software-Server, der in Java geschrieben ist. Mit seiner Hilfe können Nutzer Geodaten teilen und bearbeiten. Veröffentlicht werden die Daten mit räumlichem Bezug über offene Standards.

Google Web Toolkit (GWT) ist eine Sammlung von Klassen, Bibliotheken und Schnittstellen für die Erstellung von Webanwendungen. Die Entwicklung basiert auf Java. Das Tool besitzt einen eigenen Compiler, mit dem Java-Code in HTML und JavaScript übersetzt werden kann. Die asynchrone Kommunikation mit dem Server kann per Remote Procedure Calls und XML oder JSON vollzogen werden. Die GW-Web-UI-Klassenbibliothek ist der Kern des GWT. Sie bietet konfigurierbare Schnittstellen und stellt Klassen zur Verfügung, mit denen beispielsweise Schaltflächen und Eingabefelder generiert werden können.

Smart GWT ist eine Erweiterung von GWT, mit der Rich-Client-Webanwendungen entwickelt werden. Der Ausdruck Rich Client wird bei Server-Client-Architekturen verwendet, bei der die eigentliche Verarbeitung der Daten auf dem Client durchgeführt wird. Hierbei wird meistens auch eine grafische Benutzeroberfläche von dem Client bereitgestellt.

GIT ist eine Open Source Software zur verteilten Versionsverwaltung von Dateien.

Java ist eine objektorientierte Programmiersprache.

Java Platform, Enterprise Edition (Java EE) bezeichnet eine Softwarearchitektur, mit der plattformunabhängige, webbasierte Anwendungen entwickelt werden können. Diese Anwendungen werden in der Programmiersprache Java erstellt.

JavaScript ist eine sehr umfangreiche Script-Sprache. Zum Übersetzen des Programmes ist kein Compiler erforderlich, sondern es wird zur Laufzeit interpretiert.

JUnit Dies ist ein Framework, das zum Testen von Java-Programmen verwendet wird. Hier werden einzelne Programmteile (Module) der Software getestet. Es werden automatische Testprogramme geschrieben, die von jedem aufgerufen und durchgeführt werden können. Sie sollten nach jeder Programmänderung durchgeführt werden, um somit einen entstandenen Fehler schnell zu identifizieren. Der JUnit-Test läuft entweder korrekt durch oder bricht ab. Ist der Test abgebrochen, kann dies zwei Ursachen haben, ein Fehler (Error) oder ein falsches Ergebnis (Failure). Falsche Ergebnisse werden erwartet, wohin gegen Fehler unerwartet sind und sofort behoben werden sollten.

OpenLayers ist eine JavaScript-Bibliothek zur Darstellung von Karten in einer Webanwendung (siehe Abschnitt "OpenLayers" im Kapitel "Grundlagen").

Spring ist ein Framework, um mit Java komplexe Webanwendungen zu entwickeln.

GeoTools ist eine frei verfügbare Bibliothek von Java. Sie dient zur Bearbeitung und Darstellung geographischer Daten. Die GeoTools-Bibliothek implementiert die Spezifikation Simple Features Access des Open Geospatial Consortium (OGC).

Eclipse ist ein Programmierwerkzeug zur Entwicklung von Software. Als umfangreiche Entwicklungsumgebung (IDE = Integrated Development Environment) verfügt es über Werkzeuge, die dem Entwickler immer wiederkehrende Aufgaben bei der Softwareentwicklung abnehmen. Sowohl Eclipse also auch die verwendeten Plugins sind in Java programmiert. Die unterschiedlichen Aufgaben können in verschiedenen Sichten dargestellt und mittels „Drag and Drop“ in der Oberfläche verschoben werden. Der Quellcode wird in Editorfenstern bereitgestellt, und die Syntax ist zur besseren Sichtbarkeit farblich hervorgehoben. Zahlreiche Funktionalitäten unterstützen eine komfortable Softwareentwicklung.

Maven ist ein Build-Management-Tool, das auf Java basiert und von Apache Software Foundation entwickelt wurde, um vor allem Java-Programme standardisiert zu erstellen und zu verwalten. Bei großen Softwareprojekten müssen die benötigten Informationen unterschiedlicher Art und Herkunftsquellen zu einem gemeinsamen Produkt, dem sogenannten Build zusammengebaut werden. All diese Informationen werden bei Maven in einer XML-Datei mit dem Dateinamen pom.xml (für Project Object Model) gespeichert.

PostgreSQL/PostGIS ist ein objektrelationales Datenbankmanagementsystem, das zusammen mit der Installation des PostGIS-Pakets GIS-fähig wird.

Tomcat ist ein HTTP-Server, der dynamische Webseiten mit JSP und Java Servlets generiert, die zur Erweiterung der Funktionalität im Server dienen. Zur Ausführung der Servlets wird eine Java Virtual Machine (JVM) benötigt, die durch Tomcat bereitgestellt wird, so dass keine JVM im Browser des Clients notwendig ist.

GDAS-Entwicklungsumgebung steht für Geospatial Data Access Service im Deutschen Datenfernerkundungszentrum (DFD). GDAS bietet standardisierte Webdienstleistungen (OGC) für den Zugriff auf Geodatenätze und Fernerkundungsdatenprodukte. Weiterhin ermöglicht es den interoperablen Datenzugriff mit verschiedenen Software-Tools und stellt Geodatenätze, zum Beispiel Land-/Wassermasken für wertschöpfende Geoinformationsketten bereit. Da es eine erweiterbare Plattform ist, dient es für den hier entwickelten Webservice als Basissoftware, in der die notwendigen Erweiterungen und Entwicklungen eingebaut werden.

4.3 Vorgehensweise

4.3.1 Verwaltung der Software mit Maven

Die Entwickler der GDAS-Software teilen sich einen Server, auf dem sich sowohl die Installation von Eclipse als auch die Arbeitsbereiche der Entwickler befinden. Sobald die Entwickler ihre Software fertiggestellt haben, werden ihre Module zu einem Artefakt zusammengebaut. Dieses Zusammenbauen wird von einem Rechner übernommen, auf dem sich eine Installation von Maven, das zentrale Repository von Maven und das Repository von GIT befindet. Als Repository Manager von Maven wird NEXUS verwendet. Jeder Entwickler hat sein eigenes Maven Repository. Der Pfadname muss in der Datei `/.m2/settings.xml`, die Benutzereinstellungen von Maven enthält, mit dem Tag `<localRepository>` definiert werden. Wird ein mvn-Projekt mit "mvn install" gebaut, so werden die resultierenden Artefakte in das lokale Repository kopiert. Sobald der Entwickler seine Artefakte den anderen Entwicklern zur Verfügung stellen möchte, kopiert er diese mit "mvn deploy" in das zentrale Repository. Der Link zu dem zentralen Repository wird in einer XML-Datei mit dem Dateinamen pom.xml (für Project Object Model) mit dem Tag `<distributionManagement>` definiert. Diese Datei enthält alle Informationen zum Softwareprojekt und folgt einem standardisierten Format. Wird Maven ausgeführt, prüft es zunächst,

ob diese Datei alle nötigen Angaben enthält und ob alle Angaben syntaktisch gültig sind, bevor es weiterarbeitet.

4.3.2 GDAS-Installation

Die GDAS-Software wird vom DFD in einem zentralen Repository für wissenschaftliche Daten-Management-Systeme (Scientific Data Management Software = SDMS) zur Verfügung gestellt und soll auf dem Rechner tonga.af.op.dlr.de unter dem Verzeichnis /gdas/node/usr installiert werden.

Zum Installieren der GDAS-Software von dem zentralen SDMS-Repository werden folgende Kommandos ausgeführt. Zuerst wird die Software, die mit den Archivierungsformaten tar und zip in dem Repository archiviert wurde, in das lokale Verzeichnis kopiert:

```
/home/tiopbuild/usr/bin/sdmsdeploy de.dlr.gdas.node:
gdas-node-tomcat-postGIS:1.0-SNAPSHOT:tar.gz:delivery copy
```

Mit dem nächsten Kommando wird die Software ausgepackt:

```
/home/tiopbuild/usr/bin/sdmsdeploy de.dlr.gdas.node:
gdas-node-tomcat-postGIS:1.0-SNAPSHOT:tar.gz:delivery unpack
```

Der Abzug der postGIS-Software wird umbenannt, mit dem aktuellen Datum versehen, und ein Link wird angelegt.

Nach dem Wechsel in das Arbeitsverzeichnis werden die Konfigurationen für die Datenbank und den Server in der Datei template.properties, die im Unterverzeichnis etc liegt, angepasst:

```
#
# GDAS / SDMS Template File
# gdas-node-tomcat-postGIS
#

# General Configuration Properties
gdas.instance.name=default

# GDAS-Node Configuration Properties
gdas.node.home=/home/tiopdelivery/gdas/node/gdas-node-tomcat-postGIS

# GDAS-DB Configuration Properties
gdas.db.home=${GDAS_NODE_HOME}
gdas.db.user=geoadmin
gdas.db.host=127.0.0.1
gdas.db.port=7432
gdas.db.data= $GDAS_DB_BASE/data
gdas.db.logs=${GDAS_DB_BASE}/logs

# GDAS-Server Configuration Properties
gdas.server.home=${GDAS_NODE_HOME}
gdas.server.data=${GDAS_NODE_HOME}/data
gdas.server.logs=${GDAS_NODE_HOME}/logs
```

```
gdas.server.opts=-XX:MaxPermSize=256m -Xmx1024M -Djava.awt.headless=
true -server
gdas.server.java.home=/usr/java1.6
gdas.server.host=localhost
gdas.server.http.port=7070
gdas.server.https.port=7443
gdas.server.shutdown.port=7005
gdas.server.ajp.port=7009
gdas.server.ssl.file=${GDAS_NODE_HOME}/etc/ssl/gdas-server-cert.pem
gdas.server.ssl.keyfile=${GDAS_NODE_HOMES}/etc/ssl/gdas-server-key.pem
gdas.server.admin.name=geoadmin
gdas.server.admin.pass=XXXXXX
```

Als Nächstes muss eine Instanz für den Knotenpunkt der Geodaten eingerichtet werden, der Ablauf ist ähnlich wie oben beschrieben. Auch hier gibt es unter `gdas-node-tomcat-postGIS-geonode/etc` eine Datei `template.properties`, die angepasst werden muss. Hier wird nur der relevante Teil der Datei gezeigt:

```
# Geonetwork Configuration Properties
geonetwork.name=DLR Catalogue
geonetwork.log.dir=${GDAS_SERVER_LOGS}/geonetwork
geonetwork.db.driver=org.postgresql.Driver
geonetwork.db.host=127.0.0.1
geonetwork.db.name=geonetwork
geonetwork.db.port=7432
geonetwork.db.user=geoadmin
geonetwork.db.pass=XXXXXX
geonetwork.db.activator=
```

4.3.3 Datenbank

In dem neuen webbasierten System sollen die Qualitätsparameter und die unterschiedlichen Einstellungen des ITP bei der Verarbeitung der Daten angezeigt werden. Diese Daten sollen in einer räumlichen Beziehung stehen, somit erfolgt die Speicherung der Daten in einer Geodatenbank. Die Entscheidung fiel auf die Datenbank PostgreSQL mit der Erweiterung PostGIS. Auf dem Datenbank-Server wurde bereits eine Vorlagen-Datenbank als Muster für das Anlegen einer Datenbank eingerichtet, die Daten mit räumlichem Bezug speichert. Diese Vorlage nennt sich in unserem Fall `template_spatialdb` und besteht aus den zwei Tabellen `geometry_columns` und `spatial_ref_sys`. Befindet man sich in der Datenbank, werden die Tabellen mit dem Befehl `\dt` aufgelistet:

```
template_spatialdb=# \dt
```

```
List of relations
 Schema | Name | Type | Owner
-----|-----|-----|-----
 public | geometry_columns | table | geoadmin
 public | spatial_ref_sys | table | geoadmin
(2 rows)
```

Die Tabelle `spatial_ref_sys` enthält die bekanntesten räumlichen Bezugssysteme. In ihr stehen die Spatial Reference IDs, die Schlüssel, die sich auf das Koordinatenreferenzsystem beziehen. Der Befehl `\d` listet die Tabellendefinition:

```
template_spatialdb=# \d spatial_ref_sys
```

```
Table "public.spatial_ref_sys"
Column          | Type                               | Modifiers
-----|-----|-----
srid             | integer                            | not null
auth_name       | character varying(256)
auth_srid       | integer
rtext           | character varying(2048)
proj4text       | character varying(2048)
```

Indexes:

```
"spatial_ref_sys_pkey" PRIMARY KEY, btree (srid)
```

Die Spalten der Tabelle `spatial_ref_sys` haben folgende Bedeutung:

- `srid` ist ein Integer-Wert, der das Bezugssystem innerhalb der Datenbank eindeutig identifiziert.
- `auth_name` ist der Standardname für dieses Bezugssystem.
- `auth_srid` ist ein Integer-Wert, der das Bezugssystem gemäß dem Standard definiert.
- `srtxt` ist eine WKT-Beschreibung des Bezugssystems.
- `proj4text` ist eine Beschreibung des Bezugssystems im proj4-Ausgabeformat.

Die Tabelle `geometry_columns` stellt die Verbindung zwischen den Spalten mit Geometriedaten in Tabellen und der `spatial_ref_sys` dar.

```
template_spatialdb=# \d geometry_columns
```

```
Table "public.geometry_columns"
Column          | Type                               | Modifiers
-----|-----|-----
f_table_catalog | character varying(256)           | not null
f_table_schema  | character varying(256)           | not null
f_table_name    | character varying(256)           | not null
f_geometry_column | character varying(256)          | not null
coord_dimension | integer                           | not null
srid            | integer                           | not null
type            | character varying(30)            | not null
```

Indexes:

```
"geometry_columns_pk" PRIMARY KEY, btree (f_table_catalog,
f_table_schema, f_table_name, f_geometry_column)
```

Die ersten vier Spalten (`f_table_catalog`, `f_table_schema`, `f_table_name`, `f_geometry_column`) dienen zur eindeutigen Suche der Geometriespalte, d. h. in welcher Tabelle sie sich befindet und wie sie heißt. Die nächsten drei Spalten beschreiben die räumlichen Metadaten:

- `coord_dimension` definiert die Dimension der Spalte

- srid bietet das räumliche Bezugssystem und muss sich auf eine gültige Zeile in der Tabelle `spatial_ref_sys` beziehen.
- type stellt die Geometrie dar (Punkt, Linie, Polygon, etc.).

Aus der Vorlagen-Datenbank lässt sich eine neue Datenbank mit dem Namen `tdm_production` folgendermaßen anlegen:

```
createdb -O tdm_production -T template_spatialdb
```

In der Datenbank `tdm_production` werden nun drei Tabellen `interferometric_quality_pre_checks`, `rawdem_generations` und `precise_screenings` angelegt, sie enthalten Geometrie- und Sachdaten. Dies geschieht in zwei Schritten. Die Spalten für die Sachdaten - hier am Beispiel der Tabelle `rawdem_generations` - werden einfach mit der Create-Table-Anweisung erzeugt:

```
CREATE TABLE rawdem_generations (
  id SERIAL NOT NULL ,
  acquisition_item_id BIGINT NULL DEFAULT NULL,
  .....
);
```

Die Spalten mit Geometriedaten müssen zur Tabelle mit `AddGeometryColumn` hinzugefügt werden. In der Tabelle `rawdem_generations` wird die Spalte mit dem Namen "center", dem Geometriotyp Punkt, dem Schlüssel 4326 für das Bezugssystem WGS84 (Abkürzung für World Geodetic System 1984) und der Dimension 2 wie folgt generiert:

```
SELECT AddGeometryColumn('rawdem_generations','center', '4326', 'POINT',
'2');
```

Ein Punkt (33 11) wird folgendermaßen in die Tabelle "rawdem_generations" in die Spalte mit dem Namen "center" eingefügt:

```
INSERT INTO rawdem_generations (... , 'center', ...) VALUES
( ... , GeometryFromText('POINT(33 11)',4326), ...);
```

Die drei Punkte sollen anzeigen, dass noch mehr Spalten in der Tabelle vorhanden sind, hier aber nicht aufgeführt werden.

Eine Geometriespalte in der Tabelle `rawdem_generations` vom Typ Polygon mit dem Namen "spatial_coverage_polygon", demselben Bezugssystem und der Dimension 2, wird folgendermaßen hinzugefügt:

```
SELECT AddGeometryColumn('rawdem_generations','spatial_coverage_polygon',
'4326', 'POLYGON', '2');
```

Ein Polygon wird folgendermaßen in die Tabelle "rawdem_generations" in die Spalte mit dem Namen "spatial_coverage_polygon" eingefügt:

```
INSERT INTO rawdem_generations (... , 'spatial_coverage_polygon',...)
VALUES (... , GeometryFromText
('POLYGON(( 34 11, 34 12, 33 12, 33 11, 34 11))',4326),...);
```

Das Bezugssystem WGS84 ist ein weltweit genutztes geodätisches Datum, das die Lage und den Maßstab eines zwei- oder dreidimensionalen Koordinatensystems beschreibt.

Eine Anforderung aus dem TanDEM-X-Projekt ist, dass der neu zu entwickelnde Service den Ablauf der Generierung der Rohdaten-DEM nicht stören darf. Dies erschwert die gesamte Handhabung des neuen Systems gewaltig, da über die bereits in die Datenbank eingefügten Daten außerhalb der Datenbank Buch geführt werden muss. Aus diesem Grund wird auf einem Rechner, genannt tdxpgs, ein Skript gestartet, das die relevanten Daten für die Datenbank von dem Verarbeitungsrechner, der für die DEM-Generierung zuständig ist, auf diesen zusätzlichen Rechner überträgt. Hier befindet sich für jeden Teil der Verarbeitung (IQPC, PRCSER, RDEMGEN) des ITP jeweils ein Verzeichnis, in dem die Daten einsortiert werden. Das Skript funktioniert folgendermaßen:

1. Auf dem Verarbeitungsrechner wird eine Liste erstellt, die alle Verarbeitungsläufe enthält, die noch nicht transferiert wurden.
2. Die Liste wird auf den zusätzlichen Rechner transferiert.
3. Die Liste wird abgearbeitet.
 - 3a. Auf dem tdxpgs-Rechner werden Pfade und Interface-Files neu angelegt.
 - 3b. Für jeden Listeneintrag wird in dem Verarbeitungsrechner das entsprechende Verzeichnis gezippt, dann übertragen und auf dem tdxpgs einsortiert.
4. Wenn der letzte Eintrag abgearbeitet ist, beendet sich das Skript.

Als Nächstes werden die übertragenen Daten mit einem Import-Skript in die Datenbank importiert.

4.4 GeoServer

Beim Einrichten von GDAS und dem GeoServer wird in der Datei `template.properties` der port und der host für den GeoServer eingetragen.

```
gdas.server.host=localhost
gdas.server.port=7070
```

Der localhost ist der Rechner, auf dem GDAS und der GeoServer installiert wurden und hat den Namen `tonga.af.op.dlr.de`.

Über die URL <http://tonga.af.op.dlr.de:7070/repository/web/> kann der Benutzer, bei Eingabe seines Benutzernamens und Passwortes, Zugang zum GeoServer erhalten und gelangt auf die Einstiegsseite (siehe Abbildung 4.3). Benutzername und Passwort sind ebenfalls in der Datei `template.properties` eingetragen.

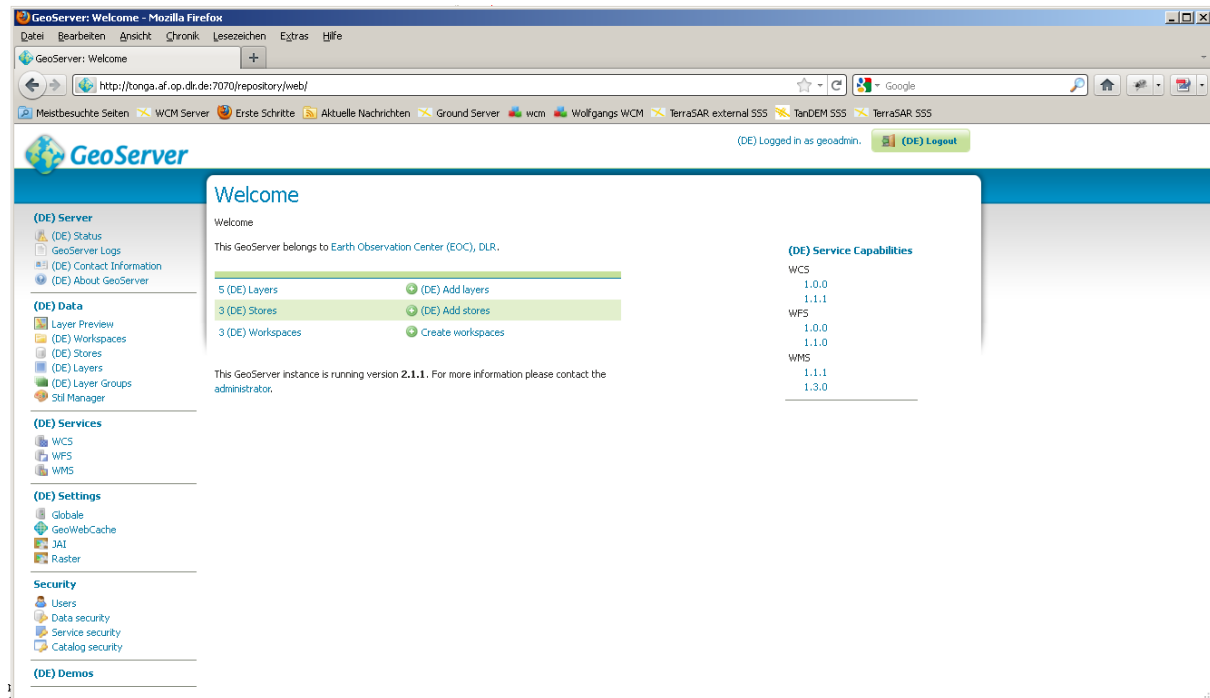


Abbildung 4.3: Einstiegsseite GeoServer

Der erste Schritt im GeoServer ist, einen Arbeitsbereich anzulegen. Dies geschieht über die links angebrachte Menüleiste unter Data - Workspaces. Es erscheint eine neue Seite, die eine Übersicht über die bereits angelegten Arbeitsbereiche auf dem GeoServer liefert. Der Nutzer hat die Möglichkeit, einen neuen Arbeitsbereich anzulegen oder einen bereits bestehenden zu bearbeiten oder zu löschen. Für den zu erstellenden Webservice wird ein neuer Arbeitsbereich mit dem Namen "itp" erstellt (Abbildung 4.4).

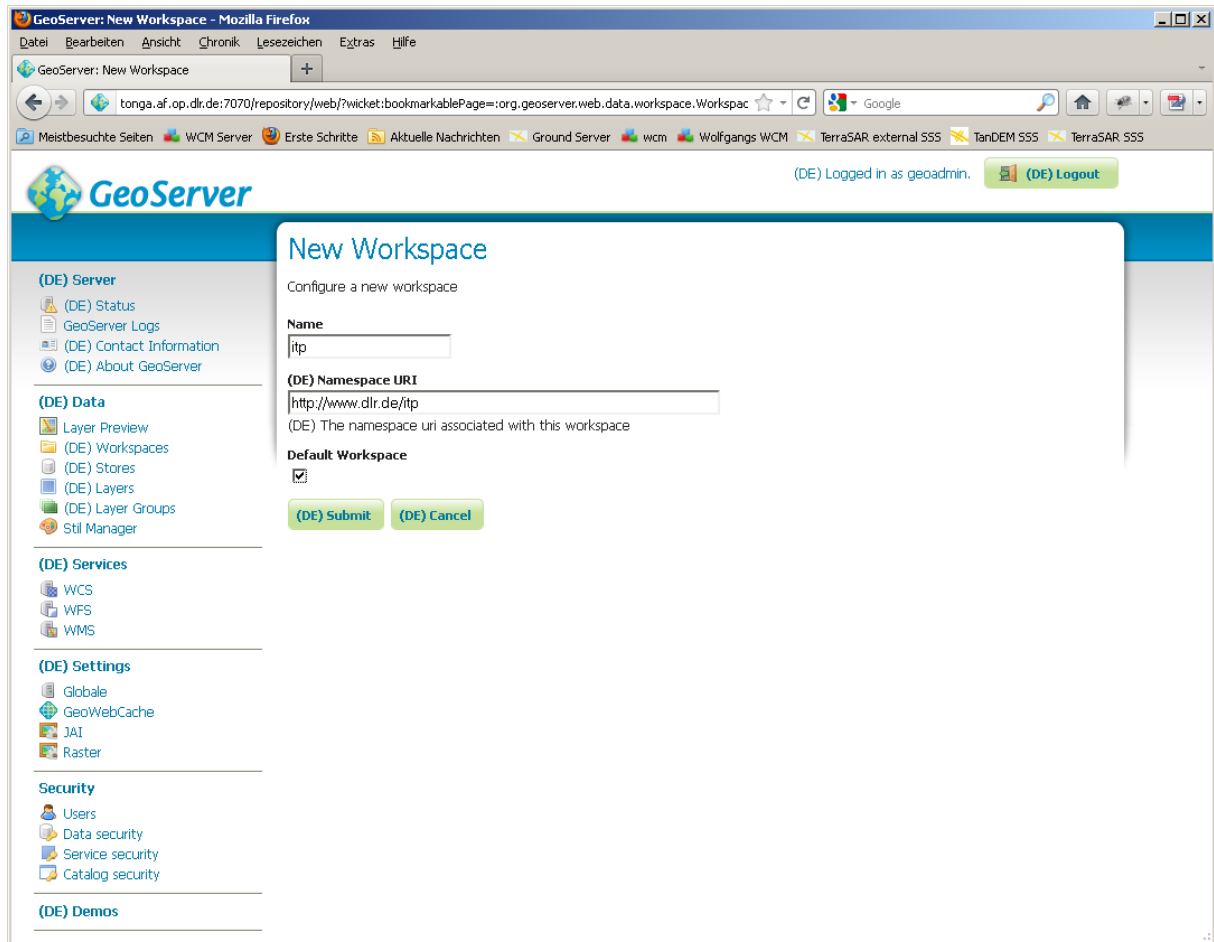


Abbildung 4.4: Neuen Arbeitsbereich anlegen

In dem Menü auf der linken Seite werden über Data - Stores die Daten angegeben, die über GeoServer zugänglich gemacht werden sollen. Zuerst muss man den Typ der Datenherkunft angeben. Unterschieden wird zwischen Vektor- und Rasterdaten und den unterschiedlichen Datenquellen innerhalb dieser Datentypen. Bei den Vektordaten können die Daten unter anderem aus einer PostGIS-Datenbank zur Verfügung gestellt werden. Da die Tabellen einer PostGIS-Datenbank zugänglich gemacht werden sollen, wird diese Option ausgewählt.

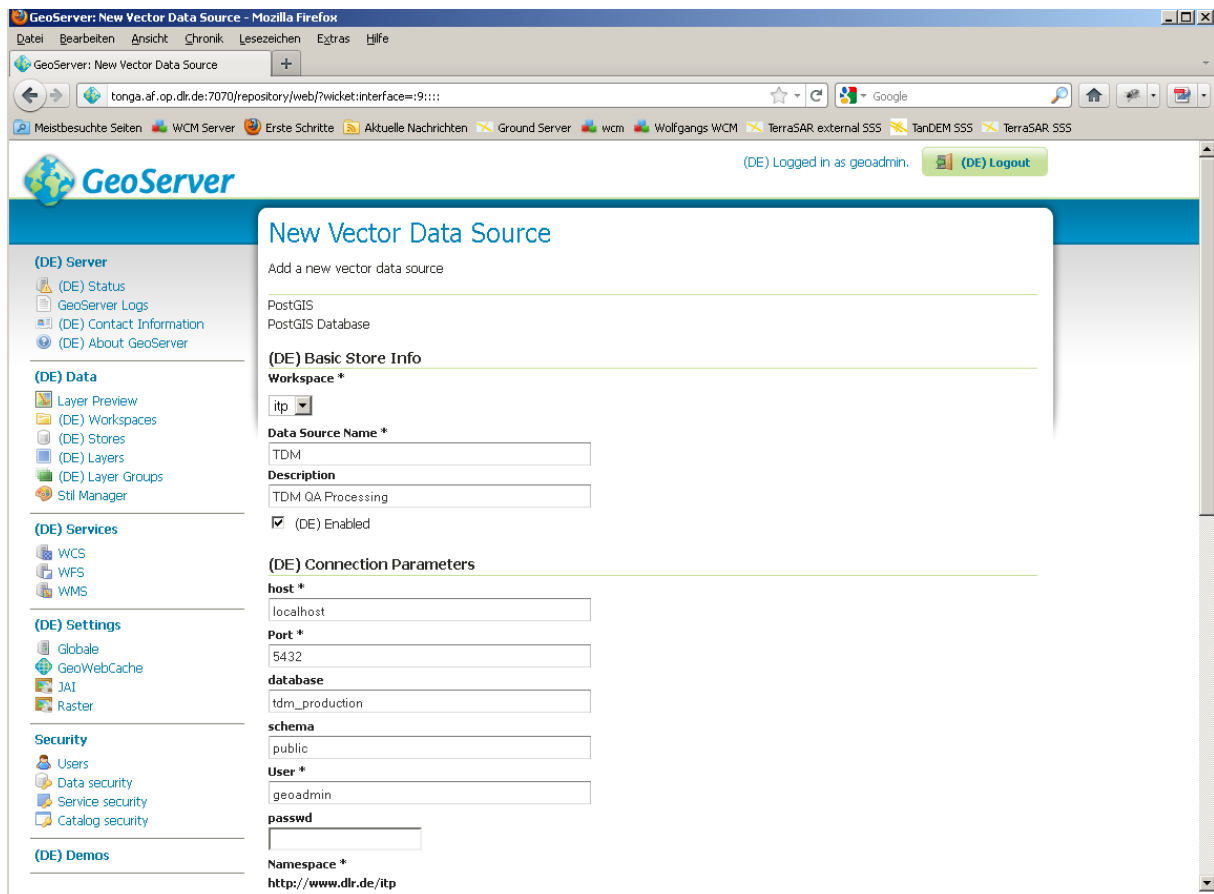


Abbildung 4.5: Neue Datenquelle anlegen

Im weiteren Vorgehen müssen der Arbeitsbereich, der Name der Datenquelle und die Verbindungsparameter eingegeben werden (Abbildung 4.5). Auch hier werden wieder die Parameter host und port aus der Datei template.properties benötigt. Der Name der Datenbank, auf die zugegriffen werden soll, heißt "tdm_production" und muss in dem Textfeld database eingetragen werden. Außerdem müssen die Zugriffsdaten des Nutzers eingegeben werden. Diese Parameter können später auch wieder verändert werden, in dem man einfach auf den Namen des "Stores" klickt, der verändert werden soll.

The screenshot shows the GeoServer web interface in a Mozilla Firefox browser. The main content area is titled "(DE) Layers" and contains a table of layers. The table has the following data:

(DE) Type	(DE) Workspace	(DE) Store	(DE) Layer	(DE) Enabled?	(DE) Native SRS
<input type="checkbox"/>	dir	countries	countries	✓	EPSG:4326
<input type="checkbox"/>	itp	dem_tif	TDM1_IDEM_04_E043_N34_DEM	✓	EPSG:4326
<input type="checkbox"/>	itp	TDM	rawdem_generations	✓	EPSG:4326
<input type="checkbox"/>	itp	TDM	tdm_production_view	✓	EPSG:4326
<input type="checkbox"/>	itp	TDM	centertype	✓	EPSG:4326

The left sidebar menu includes the following items:

- (DE) Server
 - (DE) Status
 - GeoServer Logs
 - (DE) Contact Information
 - (DE) About GeoServer
- (DE) Data
 - Layer Preview
 - (DE) Workspaces
 - (DE) Stores
 - (DE) Layers
 - (DE) Layer Groups
 - Stil Manager
- (DE) Services
 - WCS
 - WFS
 - WMS
- (DE) Settings
 - Globale
 - GeoWebCache
 - JAI
 - Raster
- Security
 - Users
 - Data security
 - Service security
 - Catalog security
- (DE) Demos

Abbildung 4.6: Layers

Links in dem Menü können über Data - Layers die bestehenden Datenquellen, die über GeoServer veröffentlicht werden sollen, konfiguriert werden (Abbildung 4.6).

Edit Layer
itp:rawdem_generations

(DE) Basic Resource Info

(DE) Name: rawdem_generations

(DE) Title: rawdem_generations

(DE) Abstract:

(DE) Keywords

(DE) Current Keywords:

(DE) New Keyword: (DE) Add

(DE) Metadata links

(DE) Coordinate Reference Systems

(DE) Native SRS: EPSG:4326

(DE) Declared SRS: EPSG:4326

(DE) SRS handling: (DE) Force declared

(DE) Bounding Boxes

(DE) Native Bounding Box

(DE) Min X	(DE) Min Y	(DE) Max X	(DE) Max Y
-99,541	-27,49	136,503	42,81

(DE) Lat/Lon Bounding Box

(DE) Min X	(DE) Min Y	(DE) Max X	(DE) Max Y
-99,541	-27,49	136,503	42,81

(DE) Feature Type Details

(DE) Property	(DE) Type	(DE) Nullable	(DE) Min/Max Occurrences
acquisition_item_id	Long	true	0/1
time_tag	Timestamp	true	0/1
operator_id	Integer	true	0/1
partition_id	Integer	true	0/1
rawdem_quality	Integer	true	0/1

Abbildung 4.7: Editieren eines Layers

Zuerst wählt man den Layer aus, der verändert werden soll. Danach erscheint eine Maske (Abbildung 4.7), um die Parameter dieser Schicht zu verändern. Sind all diese Schritte getätigt worden, kann über den GeoServer auf die Geodaten innerhalb der Datenbank zugegriffen werden.

4.5 Visualisierung

Verschiedene Module sind für die Anwendung erforderlich, um die Daten im Web zu visualisieren. Eine solche Anwendung setzt sich aus mehreren Dienstleistungen zusammen und bietet verschiedene Funktionalitäten über das Application Programming Interface (API) an. Eine spezifische Funktionalität einer Anwendung nennt sich Service und definiert die Schnittstelle für diese spezifische Funktionalität. Die Menge der öffentlichen Schnittstellen und der abstrakten Klassen, die eine Dienstleistung definieren, heißt Service Provider Interface (SPI). Ein Service Provider implementiert das Service Provider Interface und enthält eine oder mehrere konkrete Klassen, die die Implementierung einer spezifischen Anwendung realisieren. Die in dieser Arbeit entwickelte Anwendung verwendet das Konzept des Service Provider Interface von Java. Die modulare Struktur, die bei der Entwicklung verwendet wird, ermöglicht eine einfache Erweiterbarkeit. Auf Seiten des Servers werden die Module "Provider" und "EGP" verwendet, auf Seiten des Clients "API" und "GWT". In der Abbildung 4.8 ist die generelle Architektur des Webservice mit den einzelnen Modulen und deren Zusammenhang mit dem GeoServer und der Datenbank dargestellt. Die Services WMS und WCS werden bei dieser Webanwendung nicht benötigt, können aber verwendet werden und sind deswegen zur Vollständigkeit in die Abbildung mit aufgenommen worden.

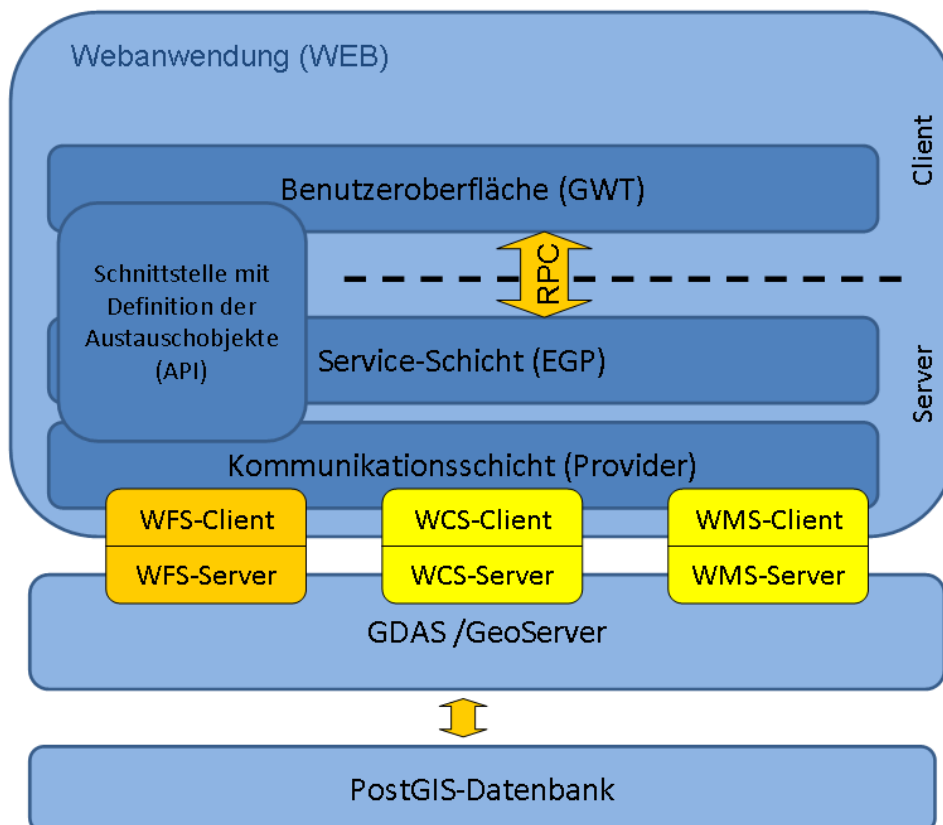


Abbildung 4.8: Generelle Architektur

4.5.1 Provider

Das Modul Provider implementiert die Kommunikationsschicht, bietet die Schnittstelle zu Internetdiensten und somit den Zugriff über den GeoServer auf die Geodaten in einer Datenbank an. Bei der Schnittstelle zwischen dem GeoServer und der Anwendung handelt es sich um einen Web Feature Service, der kompatibel zu OGC WFS 1.0.0 und WFS 1.1.0 ist. Der WFS-Server wird von dem GeoServer übernommen, wohingegen der WFS-Client in dem Modul Provider implementiert ist. Der einfache Datenzugriff ist über den Austausch von Anfrage- und Antwortnachrichten, wie in Abbildung 4.9 dargestellt, definiert.

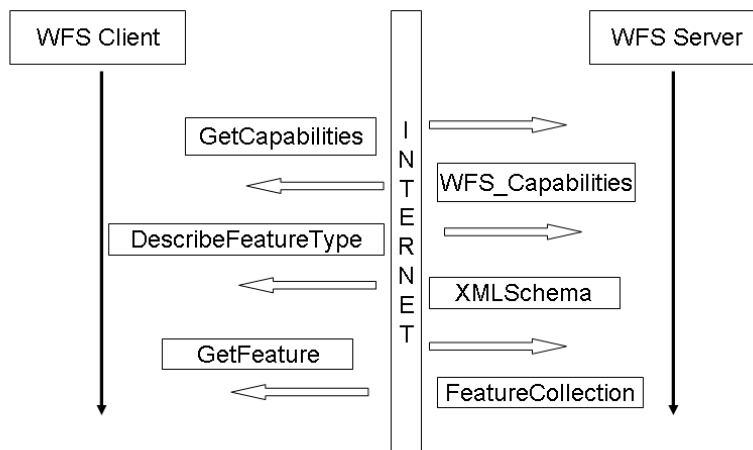


Abbildung 4.9: WFS-Nachrichtenaustausch

Der WFS-Client schickt einen `GetCapabilities`-Aufruf, also eine Anfragenachricht an den WFS-Server. Auf diese Anfrage schickt der WFS-Server eine Antwortnachricht (`WFS_Capabilities`) in Form eines XML-Dokuments zurück. Diese XML-Datei enthält alle Informationen über die räumlichen Daten und Funktionen, die für die Verarbeitung der raumbezogenen Daten derzeit auf dem WFS-Server zur Verfügung stehen. Der Zweck von `DescribeFeatureType` ist es, Informationen über den `FeatureType`, also zum Beispiel eine Liste der vorhandenen Tabellen oder der Attribute (Spaltenname) einer Tabelle, vor der eigentlichen Anfrage der Daten zu bekommen. Mit dem Aufruf `GetFeature` lassen sich Informationen zu den einzelnen Geo-Objekten (Features) erhalten, und es können räumliche Filter oder Filter auf Attribute gesetzt werden.

Bei der Implementierung des Webservice wird eine Konfigurationsdatei erstellt, in der die Konfiguration für die Verbindung zum GeoServer (WFS-Server) und dem Arbeitsbereich itp (siehe Abschnitt 4.4 und Abbildung 4.4) eingetragen wird. Die Verbindungsparameter werden somit an einer Stelle konfiguriert und können im Programmverlauf einheitlich verwendet werden. Diese Konfigurationsdatei ist folgendermaßen aufgebaut:

```
#
# DLR TDXQA Service Provider Configuration
#
#
# URL to WFS Capabilities
egp.provider.dlr.tdxqa.wfs.capabilitiesURL=
```

```

http://tonga.af.op.dlr.de:7070/
repository/itp/wfs?request=GetCapabilities
# WFS type
egp.provider.dlr.tdxqa.wfs.typeName1=
itp:precise_screenings
egp.provider.dlr.tdxqa.wfs.typeName2=
itp:interferometric_quality_pre_checks
egp.provider.dlr.tdxqa.wfs.typeName3=itp:rawdem_generations
#
# URL extension for WFS versions
egp.provider.dlr.tdxqa.wfs.version.1.0.0=VERSION=1.0.0
egp.provider.dlr.tdxqa.wfs.version.1.1.0=VERSION=1.1.0
egp.provider.dlr.tdxqa.wfs.separator=&
#

```

Unter "# URL to WFS Capabilities" befindet sich die URL des WFS-Servers. In unserem Fall ist dies der Rechner, auf dem der GeoServer installiert ist: "tonga.af.op.dlr.de" mit der Portnummer 7070. Hier können unter dem Verzeichnis "repository" die vorhandenen Arbeitsbereiche ausgewählt werden. Der gewünschte Arbeitsbereich für unser System heißt "itp" (siehe Abschnitt 4.4 und Abbildung 4.4). Und der Typ der Anfrage (engl. Request) wird als "GetCapabilities" konfiguriert. Unter WFS Type werden drei Parameter für die drei Tabellen der unterschiedlichen Szenarien (PRCSCR, IQPC, RDEMGEN) angegeben.

Neben dieser Konfigurationsdatei wird eine Java-Klasse benötigt, um den tatsächlichen Zugriff auf den GeoServer zu realisieren. Dieser Zugriff erfolgt mit der Unterstützung der Java-Bibliothek GeoTools, die standardkonforme Methoden für die Bearbeitung von Geodaten bereitstellt. Sofern eine Klasse auf Bestandteile eines Paketes innerhalb dieser Bibliothek zugreifen möchte, müssen diese Bestandteile durch eine import-Anweisung zugänglich gemacht werden. In unserem Fall wird auf das Paket Data mit den Unterpaketen DataStore, DataStoreFinder, FeatureSource und Query zugegriffen.

Damit der Zugriff an mehreren Stellen im Programmverlauf möglich ist, wird die Anfrage in einer Funktion getDataStore definiert. Die Funktion benötigt die URL des WFS-Servers für den korrekten Verbindungsaufbau. Diese URL wird aus der Konfigurationsdatei extrahiert, in der außerdem noch weitere Verbindungsparameter festgelegt sind.

Es folgt ein Programmauszug, der den Zugriff auf die Datenbank mit der Unterstützung der GeoTools-Bibliothek, insbesondere mit dem Unterpaket DataStore darstellt.

```

String url = this.capabilitiesURL + getProperty(KEY_SEPARATOR)
+getProperty(KEY_SERVICE_1_1);

private DataStore getDataStore(String url) throws IOException
LOG.debug("Getting Data Store for " + url + " ...");
Map<String, Comparable> connectionParameters =
new HashMap<String, Comparable>();
connectionParameters.put("WFSDataStoreFactory:GET_CAPABILITIES_URL",
url);
connectionParameters.put("WFSDataStoreFactory:BUFFER_SIZE",
Integer.parseInt("15"));
connectionParameters.put("WFSDataStoreFactory:TIMEOUT",
Integer.parseInt("100000"));

DataStore data = DataStoreFinder.getDataStore(connectionParameters);

```

```
LOG.debug("Getting Data Store for " + url + " done");
return data;
```

DataStore wird verwendet, um den Zugriff und die Speicherung von Geodaten in Vektor-Formaten zu ermöglichen. Mit dieser Methode kann das Schema, also die Typinformationen aufgerufen werden, wobei festgelegt werden muss, welcher Typ, also welche Tabelle verwendet werden soll. Wir bekommen die Namen der Typen wiederum aus unserer Konfigurationsdatei mit der Funktion:

```
typeName = getProperty(KEY_TYPE_NAME);
```

Der Parameter *KEY_TYPE_NAME* entspricht dem WFS Type aus der Konfigurationsdatei und lautet zum Beispiel *itp:rawdem_generations*.

Mit der Funktion *getSchema* erhält man Informationen über die Tabelle, die bei der Funktion *getProperty* eingegeben wurde:

```
SimpleFeatureType schema = data.getSchema(typeName);
```

Durch die Funktion *getAttributeDescriptors()* können folgendermaßen Informationen über die Tabelle erhalten werden:

```
List<AttributeDescriptor> attributeDescriptors =
schema.getAttributeDescriptors();
```

Für unseren Service sind die Namen der Attribute und deren Datentyp von Interesse. Damit diese Informationen innerhalb des aufzubauenden Systems an die weiteren Module weitergereicht werden können, wird eine neue Klasse erzeugt. Diese Klasse heißt "TDXQAAttributeDescriptor" und besteht aus den Variablen *name*, *type*, *defaultValue*, *value1*, *value2* und *operator*. Die Variablen *Name* (*name*), *Typ* (*type*) und der Standardwert (*defaultValue*) werden in der grafischen Benutzeroberfläche für die Anzeige der vorhandenen Parameter benötigt. Die Variablen *value1*, *value2* und *operator* sind für den Aufbau einer WFS-Anfrage gemäß den von dem Benutzer in der grafischen Oberfläche ausgewählten Parametern und den dazugehörigen eingegebenen Werten notwendig. Für jedes Attribut, das mit der *GetCapabilities*-Anfrage geliefert wird, wird eine Instanz der Klasse *TDXQAAttributeDescriptor* erzeugt und an eine Liste angehängt. Dadurch entsteht eine Liste, bei der jedes Element aus einer Instanz der Klasse

TDXQAAttributeDescriptor besteht. Diese Liste umfasst nun alle Informationen, die für das Anzeigen der Daten auf der Benutzeroberfläche benötigt werden. Des Weiteren wird eine Suchanfrage auf bestimmte Werte der Elemente der Tabelle benötigt. Die unterschiedlichen Werte werden von dem Benutzer über die Benutzeroberfläche (siehe Modul GWT) eingegeben. Gemäß diesen Eingaben wird eine gefilterte Anfrage an den WFS-Server aufgebaut, die nur die Elemente der Tabelle zurückgibt, die diese Auswahlkriterien erfüllen. Dafür muss die Datenquelle der Tabelle über die Funktion *data.getFeatureSource(typeName)* ermittelt werden, um danach eine Suchanfrage mit den entsprechenden Werten abzusetzen:

```
FeatureSource<SimpleFeatureType, SimpleFeature> source =
data.getFeatureSource(typeName);
FeatureCollection<SimpleFeatureType, SimpleFeature> features =
source.getFeatures(andQuery);
```

Die Ergebniselemente aus der gefilterten Anfrage werden in der Klasse *TDXQAResult.java* abgelegt und können somit auf der Benutzeroberfläche angezeigt werden. Hierfür wird im Provider die Schnittstellenklasse *TDXQASPI.java* (siehe Modul API) implementiert. Um den Service nach

außen bekanntzugeben, wird in einer Datei unter META-INF/services der Service eingetragen. Diese Datei besitzt denselben Namen wie die Schnittstellenklasse (TDXQASPI).

4.5.2 EGP

Die Serviceschicht wird durch das Modul EGP realisiert. Hier werden die Services, die mit dem Provider möglich sind, definiert. In dem System werden zwei Services benötigt. Der erste Service stellt eine WFS-Anfrage über den GeoServer an die Datenbank und gibt die Liste mit den Attributen und deren Parametern zurück. Der zweite Service liefert gemäß Auswahlkriterien bestimmte Elemente der Tabelle zurück. Der erste Service wird in der Klasse TDXQAServiceImpl.java implementiert und sieht folgendermaßen aus:

```
public List<TDXQAAtributeDescriptor> getTDXQAAtributes()
return tdxqaService.getTDXQAAtributes();
```

In der Klasse TDXQAServiceSPI.java werden alle Services über SPI (Service Provider Interface) eingebunden. TDXQAServiceSPI.java liefert über einen Service Loader alle registrierten Provider, die über TDXQASPI.java (siehe Modul API) definiert sind. Für den Service getTDXQAAtributes ist die Methode wie folgt in der Klasse TDXQAServiceSPI.java implementiert:

```
public List<TDXQAAtributeDescriptor> getTDXQAAtributes() {
for ( TDXQASPI TDXQAService : TDXQAServices)
{
List<TDXQAAtributeDescriptor> tdxqaAttributes =
TDXQAService.getTDXQAAtributes();
if (!tdxqaAttributes.isEmpty())
return tdxqaAttributes;
}
return null;
}
```

Der zweite Service (getWfsFeatureWithQuery) für die Abfrage auf die Tabelle mit Auswahlkriterien wird ebenfalls in diese beiden Java-Klassen eingetragen. Nun stehen die Services, die im Provider realisiert sind, für den weiteren Gebrauch zur Verfügung.

4.5.3 API

Das Modul API dient als Programmierschnittstelle und wird benutzt, um einfache Komponenten in der grafischen Benutzeroberfläche bereitzustellen. Im API werden die Schnittstellen definiert, indem auf dem Client abstrakte Klassen generiert werden. Auf der Serverseite werden diese abstrakten Klassen durch konkrete Klassen ersetzt. Der Client ruft eine Methode (abstrakte Klasse) auf, diese wird dann auf dem Server mit einer konkreten Methode implementiert. Der Client weiß nicht, wie die Implementierung auf dem Server stattfindet. Beim SPI können diese konkreten Klassen auf unterschiedlichen Servern mit unterschiedlichen Klassen/Methoden ausgeführt werden. Im Modul API werden die Informationen, die in dem Modul Provider aus der Datenbank extrahiert werden, der grafischen Benutzeroberfläche zur Verfügung gestellt. Dieses Modul dient als Schnittstelle zwischen den Modulen Provider, EGP und GWT und tauscht zwischen ihnen Datenobjekte (DAO = Data Access Objects) aus.

Die Datenaustauschobjekte werden über die Klassen TDXQAAtributeDescriptor.java und TDXQAQueryResult.java definiert. Für den Austausch selbst werden zwei weitere Klassen benötigt, die als Austauschpartner für die Klassen TDXQAServiceImpl.java und

TDXQAServiceSPI.java aus dem Modul EGP agieren. Dies sind die Klasse TDXQASPI.java und die Klasse TDXQAService.java.

TDXQASPI.java:

```
public interface TDXQASPI

/**
 * returns the List of the attributes with attributeDescriptor
 **/

List<TDXQAAAttributeDescriptor> getTDXQAAAttributes();

/**
 * returns the Query Results
 * @throws IOException
 **/

List<TDXQAQueryResult> getWfsFeatureWithQuery
(List<TDXQAAAttributeDescriptor> listTDXQAAAttribute) throws IOException;
```

TDXQAService.java:

```
public interface TDXQAService

**
 * Returns a List of TDXQAAAttributeDescriptor
 * @return
 *
List<TDXQAAAttributeDescriptor> getTDXQAAAttributes();

**
 * Returns a List of Query Results for the specified elements listed in
listTDXQAAAttribute
 * @param listTDXQAAAttribute
 * @return
 *
List<TDXQAQueryResult> getWfsFeatureWithQuery
(List<TDXQAAAttributeDescriptor> listTDXQAAAttribute) throws IOException;
```

Auf Seiten des Clients stehen nun die Services für den weiteren Gebrauch zur Verfügung.

4.5.4 GWT

Für die Darstellung der grafischen Benutzeroberfläche dient das Modul GWT. In diesem Modul werden die grafischen Elemente aufgebaut. Die Gestaltung der grafischen Benutzeroberfläche wird mit der Unterstützung des Google Web Toolkits (GWT) entwickelt. Die Entwicklung geschieht gemäß dem Entwurfsmuster Model-View-Presenter (MVP), bei dem das Modell und die Ansicht komplett getrennt gehalten werden (siehe Abbildung 4.10). Das Modell enthält die Logik der Ansicht, und die Ansicht enthält die grafischen Elemente, sozusagen die Benutzeroberfläche.

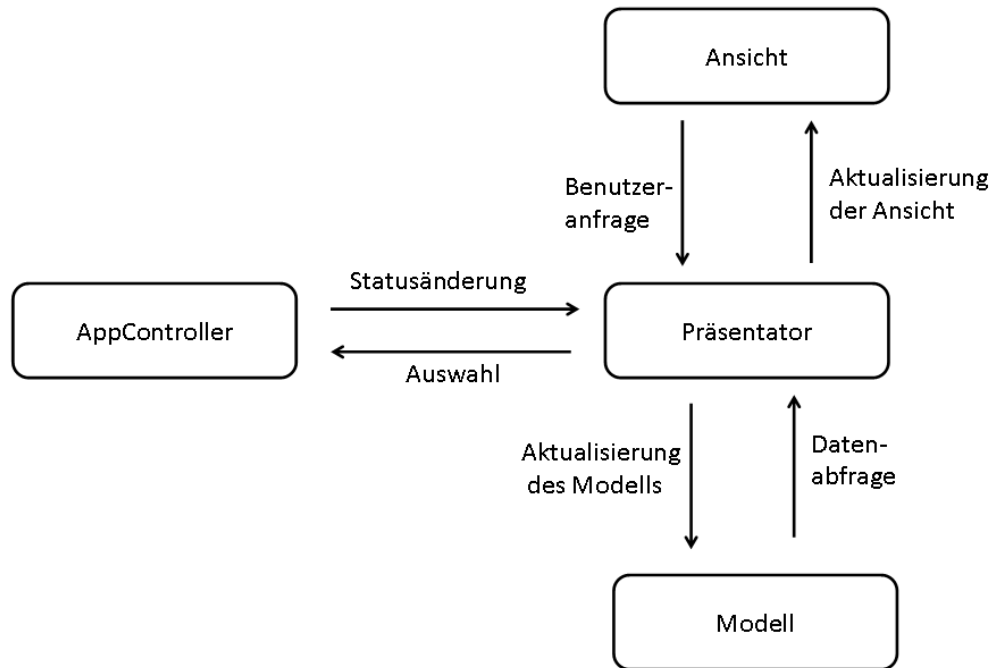


Abbildung 4.10: Model View Presenter

Der Präsentator verbindet die beiden Elemente. Die Aufgabe des Präsentators in der GWT-Anwendung ist es, benötigte Daten vom Server zu holen und Ereignisse (Events), die in der Benutzeroberfläche auftreten, zu behandeln. Die Verwendung der Klasse `AppController.java` wird im Abschnitt WEB erklärt.

Die Klassen, die in der Webanwendung das Modell realisieren, heißen `TDXQAModel.java` und `TDXQAURLModel.java`. In der Klasse `TDXQAModel.java` werden die benötigten Services aufgerufen, die von dem Modul Provider über die Module EGP und API zur Verfügung gestellt werden, um zum Beispiel die gewünschten Parameter aus der Datenbank zu extrahieren. Die Klasse `TDXQAURLModel.java` enthält die Logik, die für das korrekte Navigieren innerhalb der Webseite notwendig ist. Für die Ansicht sind folgende Klassen entwickelt worden:

- `SearchTDXQAPaneView.java` baut die Oberfläche auf, die dem Benutzer die vorhandenen Attribute der Tabelle in der Datenbank anzeigt und eine Suche der Attribute mit bestimmten Werten ermöglicht. Diese Klasse verwendet die Klasse `TDXQAQueryFormWidget.java`, die innerhalb der Benutzeroberfläche die Komponente für die Auswahl implementiert.
- `TDXQAResultView.java` zeigt die Ergebnisse der Suche in Tabellenform an.
- `TDXQAScenarioView.java` ist für die Anzeige der Auswahl der unterschiedlichen Szenarien zuständig.
- `TDXQAVectorView.java` ist für das Überlagern der Polygone auf einer Karte zuständig. Für die Generierung der Polygone werden aus der Datenbank die Elemente des Attributs "spatial_coverage_polygon" extrahiert. Dieses Attribut enthält die Eckkoordinaten der gewünschten Polygone.

Als Pendant zu diesen Klassen sind innerhalb des Präsentators folgende Klassen entwickelt worden: SearchTDXQAPanePresenter.java, TDXQAResultPresenter.java, TDXQAScenarioPresenter.java, TDXQAVectorPresenter.java.

Der Zugriff der GWT-Anwendung aus dem Browser über das Netzwerk auf den Server erfolgt über Remote Procedure Calls (RPC). Dieser Mechanismus tauscht via HTTP Java-Objekte aus. Hierfür muss eine Serviceschnittstelle auf dem Client eingerichtet werden. In der Klasse TDXQAServiceGWTWrapper.java wird die synchrone Schnittstelle definiert. Zu jedem Service muss diese Schnittstelle auch auf der Serverseite implementiert werden, dies wird in der Klasse TDXQAServiceGWTWrapperImpl.java realisiert. Nun können die Methoden, die innerhalb von TDXQAService.java (siehe API) definiert sind, aufgerufen werden. Für einen RPC muss noch eine weitere Schnittstelle erstellt werden, die die asynchrone Kommunikation übernimmt und die auf die Originalserviceschnittstelle aufbaut. Dies wird in der Klasse TDXQAServiceGWTWrapperAsync.java getätigt. Die einzelnen Bestandteile der Benutzeroberfläche sind nun generiert und müssen in dem Modul WEB zu einer einzigen Webanwendung zusammengebaut werden.

4.5.5 WEB

Für Webanwendungen wurde von Sun eine allgemeingültige Verzeichnisstruktur festgelegt, die auch in dieser Arbeit verwendet wird. Auch das von Sun spezifizierte Format der WAR-Dateien wird verwendet (WAR = Web Archive). Diese Web-Archiv-Dateien sind fertig gepackte Webanwendungen nach der Java-Servlet-Spezifikation. Hiermit kann eine gesamte Webanwendung auf den Server aufgespielt werden, die auf dem Client mit Apache Tomcat verwendet werden kann. Der Vorteil von WAR-Dateien ist, dass die Webanwendung zunächst in einer Test-Umgebung ausprobiert werden kann, bevor die WAR-Datei, sofern die Tests erfolgreich waren, im Produktionssystem installiert wird.

Der Verzeichnisbaum einer Java-Webanwendung ist aus einem "Document Root"-Verzeichnis und den Verzeichnissen "WEB-INF" und "META-INF" aufgebaut. Das Verzeichnis "Document Root" ist die Wurzel einer Webanwendung. Hier liegen die HTML- und JSP-Dateien, Bilder sowie CSS-Dateien. Unterhalb des Dokument-Root-Verzeichnisses liegt auch das "WEB-INF"-Verzeichnis, das Konfigurationsdateien und die benötigten Bibliotheken und Java-Klassen der Anwendung umfasst. Das Verzeichnis "META-INF" enthält lediglich Informationen, die für den Umgang mit WAR-Dateien auf der Seite der Java-Tools notwendig sind. Im Modul WEB werden die einzelnen Module in eine lauffähige Webanwendung integriert. Es werden alle Dateien unterhalb des "Document Root"-Verzeichnisses in eine WAR-Datei gepackt. [jsp]

Innerhalb des Moduls WEB ist die Klasse ApplicationController.java für die Kontrolle der gesamten Webanwendung zuständig. Hier wird die Anwendung unter Verwendung des Konstruktors initialisiert und der Kontrollfluss der Bildschirmanzeige gesteuert. Als Entwurfsmuster, um die Abhängigkeiten zwischen Komponenten oder Objekten zu minimieren, wird GIN (GWT INjection) verwendet. Alles was bei GIN gebraucht wird, ist ein Modul, das bestimmt, welche Schnittstellen in welcher Weise implementiert werden, damit die Services verwendet werden können. Dies geschieht durch das Einbinden der Services in die Klasse EGPMoJul.java. Die Klasse EGPGinjector.java ist die Schnittstelle der Services zu GIN. Hier wird auch der ApplicationController eingebunden.

Weiterhin gibt es auch hier dem MVP-Muster folgend die Präsentator- und die Ansicht-Klassen, die für den Aufbau der HTML-Seite der Webanwendung zuständig sind. Zum Beispiel wird die Hauptseite in den Klassen MainWindowPresenter.java und MainWindowView.java realisiert. In den Klassen TabHeaderPresenter.java und TabHeaderView.java werden die Funktionalitäten der innerhalb der Seite verwendeten Karteireiter verwaltet. Der gesamte Aufbau der Seite unterteilt sich in weitere Klassen, die hier nicht alle erwähnt werden.

Die entwickelte Webanwendung ist in mehrere Abschnitte unterteilt. Zuerst muss links in der Maske das gewünschte Szenario ausgewählt werden. Es kann zwischen IQPC, PRCSCR oder RDEMGEN gewählt werden. Danach werden die für das gewählte Szenario aus der Datenbank extrahierten Attribute in einer Liste angezeigt. Der Benutzer kann nun für eine Anfrage ein oder mehrere Attribute auswählen, die gemäß einem Operator und einem Wert nur die Elemente aus der Tabelle in der Datenbank extrahiert, die den Auswahlkriterien genügen. Die Abbildung

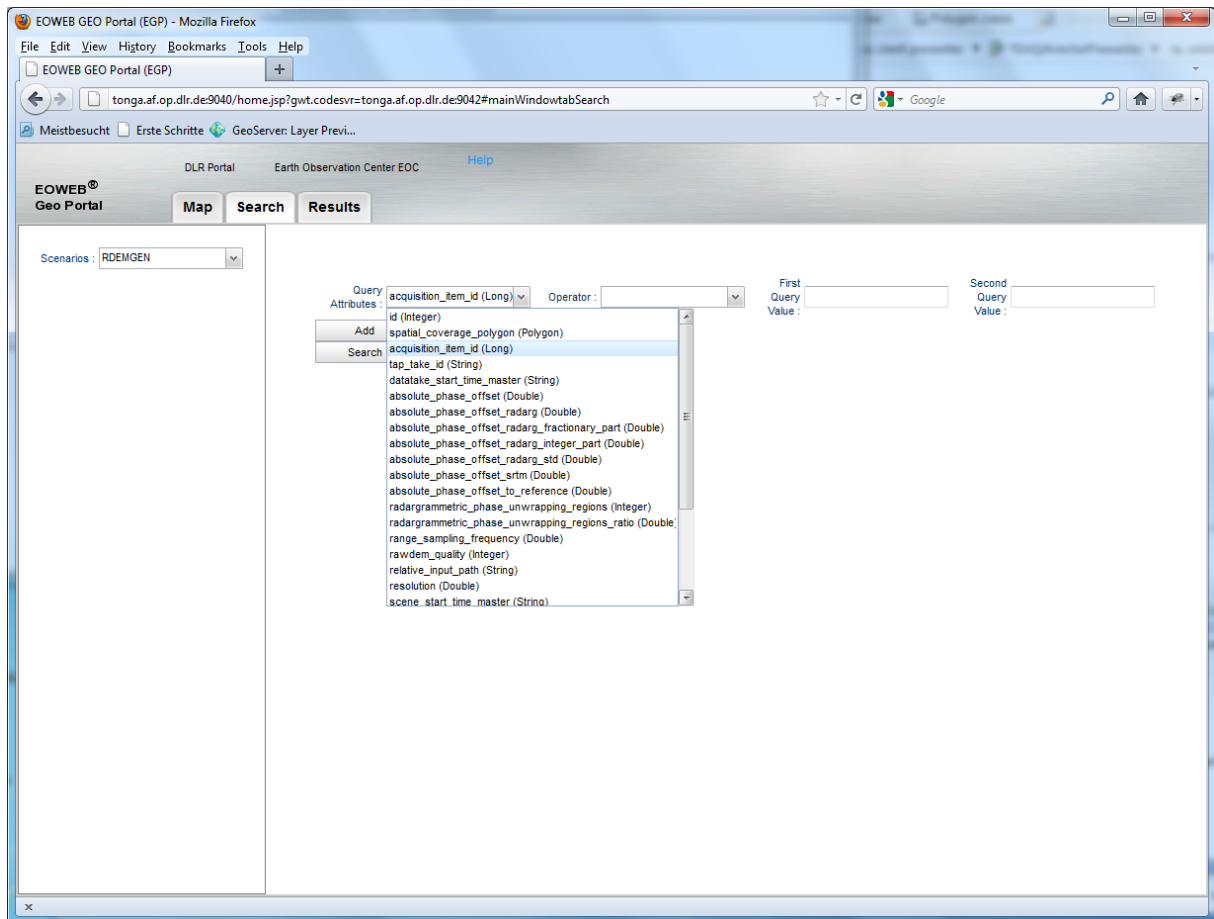


Abbildung 4.11: Suchauswahl

4.11 zeigt in der Suchmaske die vorhandenen Attribute für das Szenario RDEMGEN. Durch die Auswahl des Szenarios RDEMGEN werden die verfügbaren Attribute aus der Tabelle "rawdem_generations" extrahiert und in die Suchmaske eingefügt.

Die Abbildung 4.12 zeigt eine ausgefüllte Anfrage. Als Szenario ist wiederum "RDEMGEN" gewählt worden, dadurch wird die Tabelle "rawdem_generations" verwendet. In dieser Tabelle wird nach Elementen gesucht, die für das Attribut "acquisition_item_id" den Wert "1009226" haben. Durch die Schaltfläche "Add" können weitere Attribute in der Suche aufgenommen werden.

Wird die Schaltfläche "Search" betätigt, wird die Suchanfrage ausgeführt. Die Abbildung 4.13 zeigt die Ergebnisse der Suche in Tabellenform an.

Abbildung 4.14 zeigt die Karte mit den überlagerten Polygonen, die gemäß der Suchanfrage in der Datenbank extrahiert worden sind.

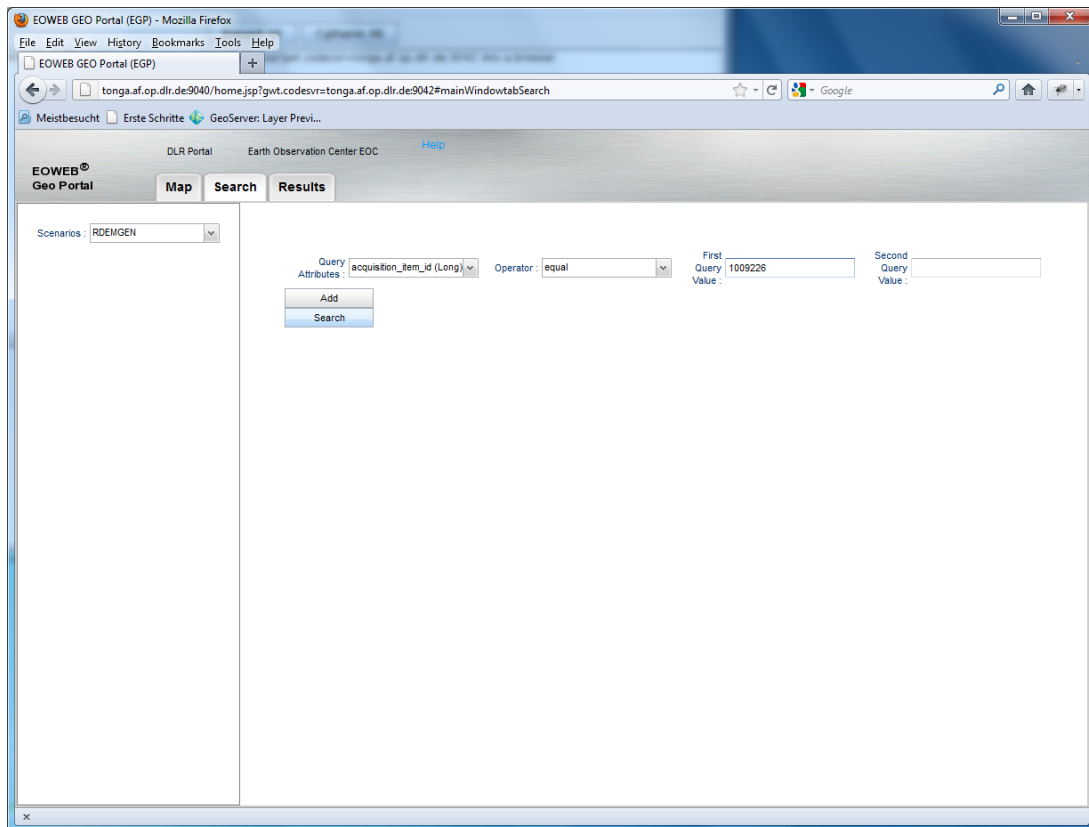


Abbildung 4.12: Suche

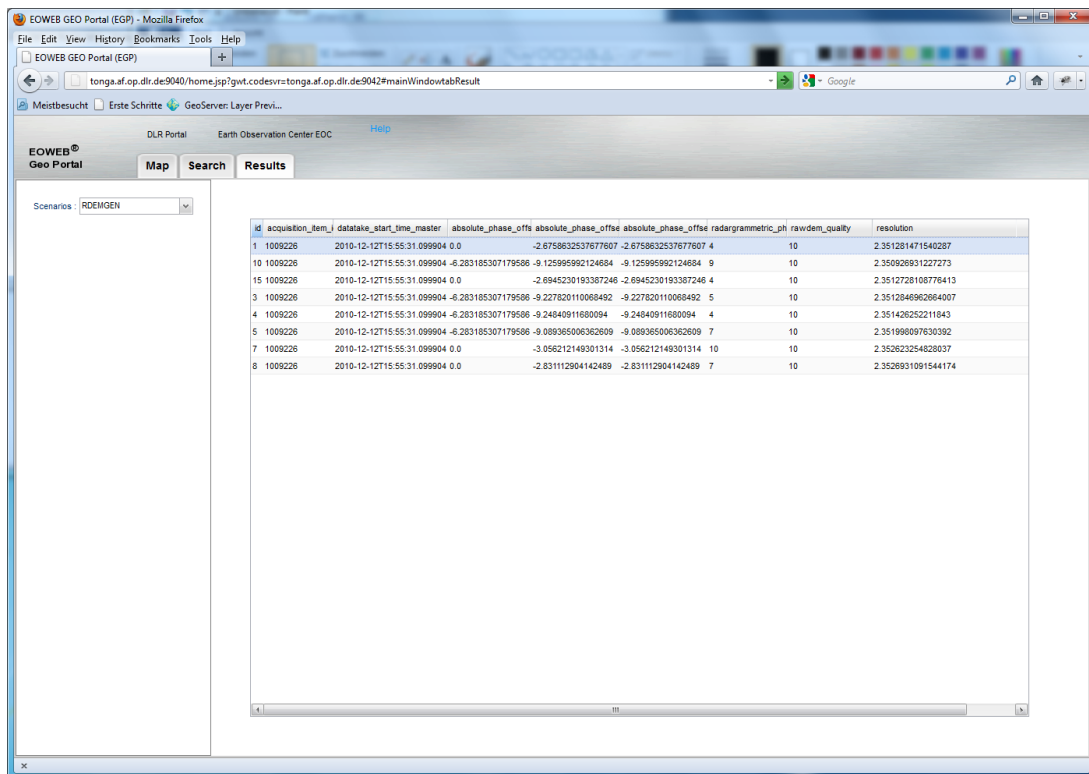


Abbildung 4.13: Ergebnisse

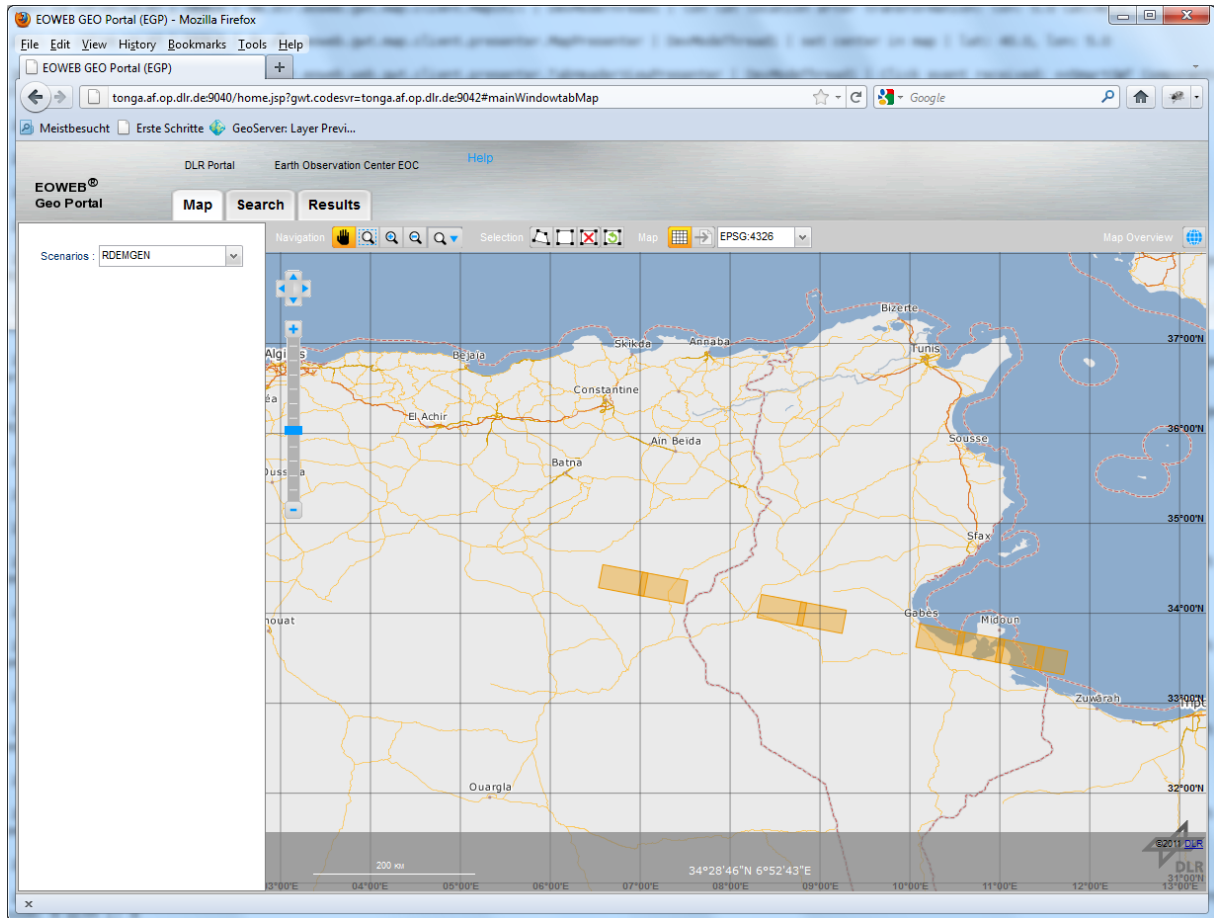


Abbildung 4.14: Karte mit überlagerten Polygonen

Kapitel 5

Schlussfolgerung

Mit der vorliegenden Arbeit wurde die Basis für einen Webservice gelegt, mit dem die grundlegenden Aufgaben zur Analyse und Visualisierung von Qualitätsparametern bewältigt werden können. Die weitere Umsetzung und Vervollständigung des Systems wird in den nächsten Monaten vollzogen.

Der Webservice besteht aus einer Vielzahl von Softwarekomponenten und Schnittstellen, worin sich schon die Komplexität dieses Systems zeigt. Die einheitliche Verwendung von standardisierten Geodatenkomponenten erleichtert die Wiederverwendbarkeit des entwickelten Codes. Zusätzlich unterstützt der Gebrauch dieser Standards den Datenaustausch mit anderen Systemen. Im Bereich von WebGIS schreiten die technischen Entwicklungen sehr schnell voran, so dass es notwendig ist, den Code so flexibel wie möglich zu gestalten, damit er leicht an neue oder geänderte Anforderungen angepasst werden kann. Der Einarbeitungsaufwand in das Google Web Toolkit (GWT), der bestehenden Projektstruktur innerhalb der Arbeitsgruppe DFD-IT und die Benutzung der Java-Bibliotheken, war trotz bestehender Vorkenntnisse verhältnismäßig hoch. Wichtig ist hier aber auch zu erwähnen, dass GWT, via Plugins, gut in Eclipse eingebunden wird. Dadurch wird das Debugging in verschiedenen Browsern ermöglicht, ohne GWT neu zu kompilieren. Dies spart wiederum Zeit bei der Entwicklung.

Die Technologie, die in dem Webservice verwendet wird, ermöglicht es, Geodaten dynamisch zu visualisieren, und vergrößert dadurch erheblich das Potenzial für deren Analyse. Wissenschaftlern wurde damit ein funktionsreiches und nützliches Werkzeug zur Verfügung gestellt, um ihre Analysetätigkeiten zu unterstützen und in manchen Bereichen sogar erst möglich zu machen. Der Webservice wurde insbesondere auf potenzielle Nutzer ausgelegt, die wenig Kenntnis über GIS besitzen. Es wird immer wichtiger, Dienstbündel anzubieten, die genau auf diese Zielgruppen zugeschnitten sind.

Der erstellte Webservice wird nur von Nutzern innerhalb des TanDEM-X-Projekts verwendet. Aus diesem Grund sind Sicherheitsvorkehrungen gegen den Zugriff von Unbefugten in dieser Arbeit weder berücksichtigt noch untersucht worden.

Anhang A

Glossar

Akquisition:

Kunden können Aufnahmen über bestimmte Gebiete bestellen. Diese Bestellungen nennt man Akquisition. Bei der Planung der Aufnahmen von SAR-Daten kann ein großer Data Take, also ein langer zusammenhängender Zeitraum einer Aufnahme, in mehrere Akquisitionen unterteilt werden. Meistens umfasst ein Data Take aber nur eine Akquisition.

Apache Tomcat:

dient zur Ausführung von Java-Code auf Webservern.

Data Take:

Dieser Begriff hat im TanDEM-X-Projekt mehrere Verwendungen. Zum einen ist ein Data Take sowohl der Prozess, als auch das Ergebnis der Aufnahme von SAR-Daten. Eine Aufnahme erfolgt von einem Satelliten innerhalb eines bestimmten zusammenhängenden Zeitraums. Zusätzlich wird der Begriff Data Take für die Datei, die die an Bord aufgenommenen Daten enthält, verwendet. Auch am Boden bezeichnet ein Data Take die Datei mit den Aufnahmedaten.

Datenserver:

Dies ist ein Rechner, der anderen Rechnern innerhalb eines Netzwerks Daten zur Verfügung stellt.

Mapserver:

ist ein Geodienst, der aus Geodaten, die im Vektor- oder Rasterformat vorliegen, Karten generiert, die via HTTP bereitgestellt werden. Weiterhin stellt er bestimmte Funktionalitäten für die Navigation und der Abfrage innerhalb der Kartenanwendung zur Verfügung.

Layer:

Thematische Ebene. Bei GeoServer bezieht sich der Begriff auf eine Schicht mit Vektor- oder Rasterdaten. Bei Vektordaten ist diese Schicht analog zu den FeatureTypes, bei Rasterdaten zu Coverages.

Features:

Dies sind die Abbildungen der realen Welt auf Objekte, die durch Eigenschaften beschrieben werden. Diese Eigenschaften können fachlicher, räumlicher und zeitlicher Natur sein.

Feature Collection:

Dies ist innerhalb eines GML-Dokuments, die Sammlung von Features. Bei einer GetFeature-Anfrage wird dies von dem WFS üblicherweise zurückgegeben.

Feature Type:

Der Feature Type bezeichnet gleichartige Features, sozusagen die Objektklasse. Er gibt die Eigenschaften für eine Menge von Features und ihre Namen an.

Gatewayserver:

Dies ist ein Rechner, der anderen Rechnern oder Geräten innerhalb eines Netzwerks Kommunikationsverbindungen zu anderen Netzen bereitstellt.

Rohdaten:

sind Daten in digitaler Form, in der Reihenfolge, in der sie aufgenommen wurden.

Quicklook:

Dies sind Übersichtsbilder von Satellitendaten, die eine geringere Bildauflösung oder Farbtiefe im Gegensatz zu den Originaldaten besitzen. Sie dienen zu einem schnellen Überblick. Da sie weniger Speicherplatz benötigen, können sie schneller zugänglich gemacht bzw. übertragen werden.

Anhang B

Abkürzungen

Ajax = Asynchronous JavaScript and XML
API = Application Programming Interface
ASP = Active Server Pages
BLOB = Binary Large Object
CGI = Common Gateway Interface
CSW = Catalog Service Web
DBMS = Datenbankmanagementsystem
DEM = Digital Elevation Model
DFD = Deutsches Fernerkundungsdatenzentrum
DLR = Deutsches Zentrum für Luft- und Raumfahrt
EOWEB = Earth Observation on the WEB
EPSG = European Petroleum Survey Group
ESRI = Environmental System Research Institute Inc.
GDAS = Geospatial Data Access Service
GIF = Graphics Interchange Format
GIN = GWT INjection
GIS = Geographisches Informationssystem
GML = Geography Markup Language
GWT = Google Web Toolkit
HTML = Hypertext Markup Language
HTTP = Hypertext Transfer Protocol
ITP = Integrated TanDEM-X Processor
IQPC = Interferometric Quality Pre-Check
IOCS = Instrument Operations and Calibration Segment
JSP = JavaServer Pages
JVM = Java Virtual Machine
KML = Keyhole Markup Language
MCP = DEM Mosaicking and Calibration Processor
MOS = Mission Operation Segment
MVP = Model View Presenter
OGC = Open Geospatial Consortium
OWS = OpenGIS Web Services
PGS = Payload Ground Segment
POM = Project Object Model
PNG = Portable Network Graphics
PRCSCR = Precise Screening
RDEMGEN = Raw DEM Generation

RPC = Remote Procedure Call
SAR = Synthetisches Apertur Radar
SDMS = Scientific Data Management Software
SPI = Service Provider Interface
SQL = Structured Query Language
SQL/MM = SQL/Multimedia
SRID = Spatial Reference Identifier
SSMM = Solid State Mass Memory
TanDEM-X = TerraSAR-X add-on for Digital Elevation Measurement
TDX = TanDEM-X
TSX = TerraSAR-X
URL = Uniform Resource Locator
UMN = Universität von Minnesota
WAR = Web Archive
WCS = Web Coverage Service
WFS = Web Feature Service
WGS84 = World Geodetic System 1984
WKB = Well Known Binary
WKT = Well Known Text
WMS = Web Map Service
XML = Extensible Markup Language

Anhang C

Literaturverzeichnis

- [AleHelBeh 2002] AA. Alesheikh, H. Helali, HA. Behroz: "Web GIS: Technologies an Its Applications", Symposium on Geospatial Theory, Processing and Applications, Ottawa 2002
- [Bri 2008] T. Brinkhoff "Geodatenbanksysteme in Theorie und Praxis Einführung in objektrelationale Geodatenbanken unter besonderer Berücksichtigung von Oracle Spatial" 2., überarbeitete und erweiterte Auflage, Wichmann 2008
- [Bri 2000]T. Brinkhoff: "The Impacts of Map-Oriented Internet Applications on Internet Clients, Map Servers and Spatial Database Systems", Proceedings 9 th International Symposium on Spatial Data Handling 10-12 August 2000, Beijing, China
- [CCte]CCGIS, terrestris: Praxishandbuch WebGIS mit Freier Software UMN MapServer PostgreSQL/PostGIS AveiN! Mapbender
- [EGHKRTT 2002] Russell East, Roop Goyal, Art Haddad, Alexander Konovalov, Andrea Rosso, Mike Tait, Jay Theodore: "The Architecture of ArcIMS, a Distributed Internet MapServer", C.S. Jensen et al. (Eds.): SSTD 2001, LNCS 2121, S. 387-403, 2001. Springer-Verlag Berlin Heidelberg 2001
- [Güt 1994] R.H. Güting, "An Introduction to Spatial Databases Systems ", The very Large Data Base J., Vol3, No4, October 1994)
- [HobFaiJam 2007] Hobona G, Fairbairn D, James P.: "Workflow Enactment of Grid-Enabled Geospatial Web Services" Proceedings of the UK e-Science All Hands Meeting 10-13 September 2007
- [Höf 2009] Stefan Höffken: Google Earth in der Stadtplanung - Die Anwendungsmöglichkeiten von Virtual Globes in der Stadtplanung am Beispiel von Google Earth, Berlin 2009 - 96 S. ISBN 978-3-7983-2116-8
- [JanAdm 2010] Marc Jansen, Till Admas "OpenLayers. Webentwicklung mit dynamischen Karten und Geodaten" Open Source Press 2010
- [KiKi 2002] Do-Hyun Kim, Min-Soo Kim: "Web GIS Service Component Based On Open Environment", Geoscience and Remote Sensing Symposium, 2002. IGARSS '02. 2002 IEEE International
- [KorZeh 2008] Korduan/Zehner "Geoinformation im Internet, Technologie zur Nutzung raumbezogener Informatioen im WWW" Wichmann Verlag Heidelberg 2008
- [LeuRei 2000] Kristin Leukert, Wolfgang Reinhardt: "GIS-Internet Architectures", International Archives of Photogrammetry and Remote Sensing (IAPRS) Vol. XXXIII (2000), Part B4, S. 572-578.
- [Lu 2005] Xiaolin Lu: "An Investigation on Service-Oriented Architecture for Constructing Distributed Web GIS Application", Proceedings of the 2005 IEEE International Conference on Services Computing (SCC'05)
- [Lu 2006] Xiaolin Lu: "Develop Web GIS Based Intelligent Transportation Application Systems with Web Service Technology", 2006 6th International Conference on ITS Telecommunications

Proceedings

- [NeuMatKup] Neumann, Mathiak, Kupfer: "Modellierung und kartographische Visualisierung von Geodaten mit XML-basierten Sprachen", Informatik Forsch. Entw. (2005) 20: 24–32 Springer Verlag
- [MilStoDjo 2005] Aleksandar Milosavljević, Leonid Stoimenov, Slobodanka Djordjević-Kajan: "An architecture for open and scalable WebGIS", 8th AGILE Conference on GIScience, Estoril, Portugal, 2005
- [OTT, SWIACZNY 2001] Ott, Th.; Swiaczny, F.: Time-Integrative Geographic Information Systems. Management and Analysis of Spatio-Temporal Data. Berlin, Heidelberg (Springer) 2001
- [Pen 1998] Z-R Peng: "An assessment framework for the development of Internet GIS" Environment and Planning B: Planning and Design 1999, volume 26, S. 117-132
- [RigSchVoi 2001] P. Rigeaux, M. Scholl, and A. Voisard, "Spatial Databases-with Application to GIS" Morgan Kaufmann Publishers 2001
- [TAL 2001] Gilles Taladoire: "Geospatial data integration and visualisation using open standard, 7th EC-GI & GIS Workshop Potsdam, Germany 13 - 15 June 2001
- [TsoBut 2002] Ming-Hsiang Tsou, Barbara P. Buttenfield: "A Dynamic Architecture for Distributing Geographic Information Services", Transactions in GIS, Volume 6
- [TSO 2004] Ming-Hsiang Tsou, "Integrating web-based GIS and image processing tools for environmental monitoring and natural resource management", J Geograph Syst (2004) 6:1–20
- [TUF 1190] Envisioning Information, Edward R. Tufte Verlag: Graphics Press (Mai 1990) ISBN-10: 9780961392116
- [VoiDav 2002] A. Voisard, B. David "A Database Perspective on Geospatial Data Modelling" IEEE Transactions in Knowledge and Data Engineering, Vol14, No.2 March/April 2002
- [VOS 1994] Vossen, G.: Datenmodelle, Datenbanksprachen und Datenbank-Management-Systeme, Verlag Oldenburg R., 1994, 686 S., ISBN 3-486-24355-1
- [Yan 1999] Peisheng Zhao Chongjun Yang: "The Studies on the Architecture of Web-GIS", Towards Digital Earth— Proceedings of the International Symposium on Digital Earth Science Press ,1999

Web-Referenzen

- [DLR] <http://www.dlr.de>
- [earthuserman] <http://earth.google.com/support/>
- [earthsup] <http://support.google.com/earth/bin/answer.py>
- [ESRI] <http://www.esri.com/software/arcGIS/arcimps/index.html>
- [esriHelp] http://webhelp.esri.com/arcimps/9.2/general/topics/arch_middlewares.htm
- [esriSup] <http://support.esri.com/en/downloads/samples-and-utilities/view/productid/16/metaid/177>
- [geoport] <http://www.geoportal.rlp.de>
- [GeoServ] <http://geoserver.org/display/GEOS/Welcome>
- [GeoservHis] <http://docs.geoserver.org/stable/en/user/introduction/history.html>
- [git] <http://book.git-scm.com>
- [googEarth] <http://www.google.de/intl/de/earth/explore/products>, 01.08.2011
- [Gros 2006] http://www.ucGIS.org/summer2006/studentpapers/grossner_final.pdf
- [gwtMVP] <http://code.google.com/intl/de-DE/webtoolkit/articles/mvp-architecture.html>
- [jsp] <http://www.jsptutorial.org/content/directoryStructure>
- [MapServ] <http://www.mapserver.org>
- [MySQL] <http://dev.mysql.com/doc/refman/5.1/de/GIS-wkb-format.html>
- [ogc] <http://www.opengeospatial.org/ogc>
- [ogckml] <http://www.opengeospatial.org/standards/kml>
- [ogcsfa1] <http://www.opengeospatial.org/standards/sfa> Open Geospatial Consortium, 2011 Standard for Geographic information - Simple feature access - Part 1: Common architecture, Version

1.2.1, OpenGIS Project

[ogcsfs] <http://www.opengeospatial.org/standards/sfs> Open Geospatial Consortium, 2010 Standard for Geographic information - Simple feature access - Part 2: Common architecture, Version 1.2.1, OpenGIS Project

[ogsfs2] http://portal.opengeospatial.org/files/?artifact_id=829 Open GIS Consortium, Inc. OpenGIS Simple Features Specification For SQL Revision 1.1

[ogcwf] <http://www.opengeospatial.org/standards/wfs> Open Geospatial Consortium, 2010 Web Feature Server Interface Implementation Specification, Version 2.0.0, OpenGIS Project

[ogcwms] <http://www.opengeospatial.org/standards/wms> Open Geospatial Consortium, 2006 Web Map Service Implementation Specification, Version 1.3.0, OpenGIS Project

[OpenLa] <http://openlayers.org>

[osgeo] <http://www.osgeo.org>

[osOpen] <http://www.osgeo.org/openlayers>

[postGIS] <http://www.postgis.org/docs/>

[p2pfoun] http://p2pfoundation.net/Open_Planning_Project

[smartt] <http://forums.smartclient.com/showthread.php?t=8159>

[spr01] <http://static.springsource.org/spring/docs/3.0.x/spring-framework-reference/pdf/spring-framework-reference.pdf>

[spr02] <http://www.springbyexample.org/examples/index.html>

[wikiOpen] <http://www.GISwiki.org/wiki/OpenLayers>

[wiki] <http://www.wikipedia.org>