



Master Thesis

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

zum Thema

„Geoverarbeitung im Wandel: Vom Desktop in das World Wide Web“

vorgelegt von

Christoph Brennecke B. Sc.
U102849, UNIGIS MSc Jahrgang 2012

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

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

Göttingen, 25. Februar 2015

Erklärung

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

Ort und Datum

Unterschrift

Inhaltsverzeichnis

Erklärung.....	II
Abbildungsverzeichnis.....	V
Listingverzeichnis	VI
Abkürzungsverzeichnis.....	VII
1 Einleitung	1
1.1 Thema	1
1.2 Ziele	2
1.3 Verwandte Arbeiten	3
1.4 Methoden	3
1.5 Aufbau.....	4
2 Grundlagen	5
2.1 Geographische Informationssysteme.....	5
2.2 Onlineinfrastrukturen	8
2.2.1 E-Science und Cyberinfrastructure.....	8
2.2.2 Geospatial Cyberinfrastructure.....	10
2.2.3 Geodateninfrastruktur (GDI).....	11
2.3 Interoperabilität durch Standards	15
2.3.1 Basis Webtechnologien des W3C	17
2.3.2 Web-Services und Service orientierte Architekturen	19
2.3.2.1 SOAP basierte Webservices	21
2.3.2.2 REST basierte Webservices	22
2.3.3 Web GIS Technologien	22
2.3.4 Räumliche Webservices.....	25
2.3.5 Web Processing Service (WPS).....	27
2.3.5.1 Ablauf einer WPS Interaktion	27
2.3.5.2 Open Source Implementierung	29
2.4 Geoverarbeitungsagenda	31
2.4.1 Semantik.....	31
2.4.2 Orchestrierung	32
2.4.3 Performance	33

2.5	Workflow Management Systeme	35
3	Offline Geoverarbeitung	36
3.1	Geoverarbeitung mit ArcGIS.....	36
3.2	Workflow und Anwendungsgebiet.....	37
3.3	Module.....	40
3.4	Ergebnis	46
4	Online Geoverarbeitung	50
4.1	Online Bereitstellung mit ArcGIS for Server	50
4.2	Ausführen des WPS	52
4.3	Orchestrierung.....	56
4.3.1	Orchestrierung mit Taverna.....	56
4.3.2	Alternative Orchestrierung mit dem ArcGIS Model Builder	58
4.4	Performance der Orchestrierung.....	60
5	Fazit	62
6	Literaturverzeichnis.....	63
7	Anhang	68
7.1	Attributtabelle des Ergebnisses der Near Analyse	68
7.2	Tutorial Online Bereitstellung eines Geoverarbeitungsservices mit ArcGIS for Server.....	73
7.3	Tutorial Ausführung eines PyWPS in Taverna	79
7.4	WPS Requests des ‚Population Distance‘ Modells	84
7.4.1	Get Capabilities Request	84
7.4.2	Describe Process Request.....	85
7.4.3	Antwort des Execute Requests	88
7.5	Messprotokolle der ausgeführten Modelle	89

Abbildungsverzeichnis

Abbildung 1: Geoverarbeitungsagenda.....	2
Abbildung 2: Entwicklung von Geoprocessing zum Geoprocessing Web.....	7
Abbildung 3: E-Science Komponenten.....	9
Abbildung 4: Komponenten einer Geospatial Cyberinfrastructure	11
Abbildung 5: Architektur einer Geodateninfrastruktur.....	12
Abbildung 6: Komponenten des Geoprocessing Web	14
Abbildung 7: Charakteristika von Interoperabilität	15
Abbildung 8: Webservice-basierte SOA	20
Abbildung 9: Aufbau einer SOAP Nachricht.....	21
Abbildung 10: Web GIS Architektur	23
Abbildung 11: Web GIS Clients	25
Abbildung 12: Ablauf einer WPS Interaktion	28
Abbildung 13: Funktion von PyWPS innerhalb einer Client-Server Architektur.....	29
Abbildung 14: ‚Population Distance‘ Modell	39
Abbildung 15: Modul 1: Project	40
Abbildung 16: Modul 2: Extract by Location	41
Abbildung 17: Modul 3: Near	42
Abbildung 18: Modul 4: Buffer.....	43
Abbildung 19: Modul 5: Statistics.....	44
Abbildung 20: Modul 6: Join.....	45
Abbildung 21: Ergebnis des ‚Population Distance‘ Modells	47
Abbildung 22: Attributtabelle des erstellten Near Features	48
Abbildung 23: Attributtabelle der erstellten Buffergeometrie.....	49
Abbildung 24: Prozesse des PML WPS in Taverna	56
Abbildung 25: Prozesse des erstellten WPS in Taverna	57
Abbildung 26: Ausführung des v.extract Prozesses des PML WPS	57
Abbildung 27: Prozess des erstellten WPS in Taverna	58
Abbildung 28: Ergebnisse der Performancemessung	61

Listingverzeichnis

Listing 1:	URL der bereitgestellten Modelle auf ArcGIS for Server	50
Listing 2:	REST-URL des Webservices	51
Listing 3:	GetCapabilities Request via HTTP Get	52
Listing 4:	DescribeProcess Request via HTTP Get	52
Listing 5:	URL des Execute Requests via HTTP Post	53
Listing 6:	XML Inhalt des Execute Requests via HTTP Post.....	55
Listing 7:	SOAP-URL des Webservices.....	56

Abkürzungsverzeichnis

BPEL	Business Process Execution Language
CI	Cyberinfrastruktur
CSW	Catalogue Service Web
DBMS	Datenbankmanagementsystemen
DIN	Deutsches Institut für Normung
DLR	Deutsches Zentrum für Luft- und Raumfahrt
ESRI	Environmental Systems Research Institute
GI	Geographische Informationen
GIS	Geographisches Informationssystem
GDI / SDI	Geodateninfrastruktur / Spatial Data Infrastructure
GEOSS	Global Earth Observation System of Systems
GRASS	Geographic Resources Analysis Support System
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
INSPIRE	Infrastructure for Spatial Information in the European Community
OASIS	Organization for the Advancement of Structured Information Standards
OE	Orchestrierungs-Engine
OGC	Open Geospatial Consortium
OSM	Open Street Map
OWS	OGC Webservice
PML	Plymouth Marine Laboratory

REST	Representational State Transfer
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SOS	Sensor Observation Service
UDDI	Universal Description, Discovery and Integration protocol
UML	Unified Modeling Language
URL	Uniform Resource Locator
VGI	Volunteered Geographic Information
W3C	World Wide Web Consortium
WCS	Web Coverage Service
WFS	Web Feature Service
WMS	Web Map Service
WPS	Web Processing Service
WS-BPEL	Web Service Business Process Execution Language
WSDL	Web Services Description Language
WSO	Web Service Orchestration
WWW	World Wide Web
XML	Extensible Markup Language

1 Einleitung

1.1 Thema

Der Verbindung des Internets und des World Wide Webs mit traditionellen Disziplinen hat sowohl neue Disziplinen geschaffen als auch erweitert. Auch Web GIS, als Teil der Geoinformatik und GI Science, kann zu dieser Entwicklung hinzugezählt werden. Das Web hat Geographische Informationssysteme (GIS) einer breiteren Masse zugänglicher und es auch flexibler gemacht. Die Art und Weise wie geographische Informationen erlangt, übertragen, veröffentlicht, geteilt und visualisiert werden hat sich dadurch erheblich geändert. Seit der Erfindung des ersten Geographischen Informationssystems durch Roger Tomlinson 1962, über die Entwicklung des ersten Web GIS 1993, haben sich diese immer mehr weg von der reinen Desktop Anwendung, hin zu bereitgestellten Modellen die auf interoperablen GIS Webservices basieren, entwickelt (Fu, 2011, p.4 ff). Diese interoperablen und verketteten Webservices werden mit zunehmender Wichtigkeit geographischer Informationssysteme in Unternehmen und der Verfügbarkeit räumlicher Daten immer wichtiger (Alameh, 2003, p.22). Auch Staub (2009, p.1) beschreibt diese Entwicklung mit dem Wandel vom Aufbau von Datenbanksystemen und von Geographischen Informationssystemen seit den siebziger Jahren, zu den aktuellen Entwicklungen im Bereich des Datenaustausches, der Zusammenführung von Datenbeständen und Aspekten der Zusammenarbeit von verschiedenen Systemen.

Bei dieser Entwicklung spielt die Interoperabilität, also *„die Fähigkeit von Softwarekomponenten zu einer selbstständigen Zusammenarbeit über Software-, System- und Institutionsgrenzen hinweg“* (Andrae, 2012, p.15), eine Hauptrolle. Interoperabilität tritt auch im Zuge der Entstehung von Geodateninfrastrukturen (GDI) auf nationaler und internationaler Ebene, sowie bei der Problematik der Nutzung verteilter Geodaten auf. In vielen aktuellen Anwendungen und Initiativen steht die gemeinsame Nutzung von verteilten Geodaten über OGC Webservices (OWS) im Vordergrund (Staub, 2009, p.19 f).

Bezugnehmend auf diese aktuellen Themen dreht sich diese Arbeit um den von dem Open Geospatial Consortium (OGC) definierten Web Processing Service (WPS). Mit dieser Spezifikation aus dem Jahr 2007 sind standardisierte Analyse und Verarbeitung von Geodaten im Internet definiert (OGC, 2007c). Für eine solche Verarbeitung werden nach Poore (2011, p.5010) hochleistungsfähige (Online-) Infrastrukturen benötigt, die in der Lage sind diese großen Datenmengen zu speichern, zu analysieren und zu verarbeiten. Auch in der Wissenschaft allgemein wird interdisziplinäre Zusammenarbeit eine immer größere Rolle spielen und somit der interoperable Austausch von Daten, Werkzeugen oder Computermodellen.

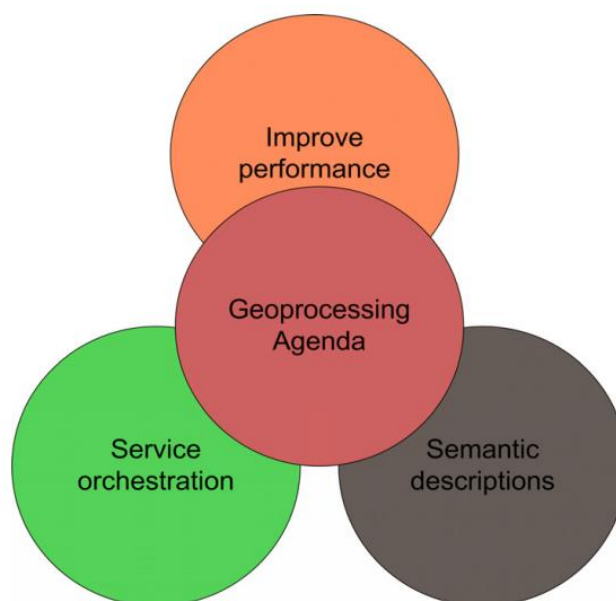
1.2 Ziele

In dieser Masterarbeit „Geoverarbeitung im Wandel: Vom Desktop in das World Wide Web“ soll zunächst ein Offline Workflow entwickelt werden, der anschließend in eine Online Infrastruktur überführt wird. Zusätzlich soll dieser Workflow modularisiert werden und anschließend wieder orchestriert werden, um eine möglichst hohe Wiederverwendbarkeit zu erreichen. Anhand der Orchestrierung des modularisierten Workflows soll die Performance der Orchestrierung gemessen werden und eine Aussage getroffen werden, ob und wie effizient diese ist. Die Forschungsfrage der diese Arbeit zugrunde liegt lautet demnach wie folgt:

Wie wirkt sich die Modularisierung von Geo-Webservices auf deren Effizienz aus?

Brauner et al. (2009, p.4 ff) teilt die Forschung zum Thema Geoverarbeitungsservices zunächst in zwei Kategorien ein. Zum einen Forschung, die sich auf generische Probleme dieser Services, wie Orchestrierung und Performance, bezieht. Zum anderen Forschung die sich mit der Umsetzung in speziellen Anwendungen beschäftigt. Über die Anwendungsgrenzen hinaus können so drei Gemeinsamkeiten der aktuellen Forschungsarbeit festgehalten werden: Service Orchestrierung, semantische Beschreibungen und Performance Verbesserungen. Die Autoren definieren diese Gemeinsamkeiten als aktuelle und zukünftige Forschungsagenda um webbasierte Geoinformationen im Zusammenhang mit Geoverarbeitungsservices verfügbar zu machen.

Abbildung 1: Geoverarbeitungsagenda



Quelle (Brauner et al., 2009, p.6)

Brauner et al. (2009) folgend spielen diese drei Hauptforschungsgebiete auch in dieser Arbeit eine Rolle. Es soll die anwendungsspezifische Orchestrierung eines mit ArcGIS for Server veröffentlichten Webservices untersucht und die Performance dieser Orchestrierung gemessen werden.

1.3 Verwandte Arbeiten

Auf dem Gebiet des Online Geoprocessing gibt es viele Arbeiten, die sich mit der Umsetzung und Orchestrierung eines WPS beschäftigen. Im Kapitel 2.4.2 werden einige Arbeiten in diesem Zusammenhang genannt und erläutert was sich daraus für diese Arbeit ergibt. Als Vorbild dieser Arbeit ist die Arbeit von de Jesus et al. (2012) zu nennen. Im Gegensatz zu vielen anderen Arbeiten, die sich mit Orchestrierung beschäftigen, steht hier eine anwenderfreundliche Orchestrierung mit einem Workflow Management System im Vordergrund. Die genauen Inhalte dieser Arbeit sollen ebenfalls im 2.4.2 genauer betrachtet werden und in Bezug mit der verwendeten Methode dieser Arbeit gebracht werden. Als Erweiterung zu den bisherigen Arbeiten soll diese Arbeit zusätzlich Wert auf die Performance der Orchestrierung von Webservices legen.

1.4 Methoden

In dieser Arbeit werden im Zusammenhang mit der Bereitstellung eines WPS bzw. eines RESTful Webservices verschiedene Methoden angewandt. Zunächst wird mittels des ModelBuilders der Software ESRI ArcMap aus dem Anwendungspaket ArcGIS for Desktop (ESRI, 2015a) ein entsprechendes Geoverarbeitungsmodell erstellt. Mittels des ESRI Softwarepakets ArcGIS for Server (ESRI, 2015b) wird dieses Modell anschließend Online bereitgestellt. Zum einen als OGC WPS und zum anderen auch als Geoverarbeitungsmodell das über die REST-Schnittstelle aufgerufen werden kann. Als Client zum Aufrufen der bereitgestellten Webservices dienen in dieser Arbeit zum einen ArcMap, um das Geoverarbeitungsmodell über die REST Schnittstelle aufzurufen, und zum anderen der Webbrowser Google Chrome (Google, 2015), über den der erstellte WPS entweder über einen HTTP-Get Befehl oder über einen HTTP-Post Befehl ausgeführt werden kann. Während der HTTP-Get Befehl über die Befehlszeile im Browser eingegeben wird, wird zum Ausführen eines HTTP-Post Befehls eine spezielle Browser Erweiterung benötigt. Hier kommt der Postman REST Client zum Einsatz (Postdot Technologies, 2015). Im Zuge der Orchestrierung wird zum einen ein Weg aufgezeigt der die Verkettung eines anderen Webservices mit dem Workflow Management System Taverna zeigt (Taverna, 2015). Zum anderen wird der erstellte RESTful Webservice wiederum mittels des ArcGIS ModelBuilders orchestriert. Die Performance wird schließlich über die ArcGIS interne Zeitmessung ausgelesen und verglichen.

1.5 Aufbau

Zu Beginn dieser Masterarbeit wird die Definition und Betrachtung der allgemeinen Geoverarbeitung im Bereich Geographischer Informationssysteme wichtig sein. Weiter werden im Kapitel Grundlagen, Konzepte der E-Science und damit verwandte definiert und voneinander abgegrenzt. Wie schon erwähnt steht auch die Interoperabilität im Vordergrund und in diesem Zusammenhang werden die standardgebenden Organisationen, wie das World Wide Web Consortium (W3C) aus dem Bereich der allgemeinen Web Technologien, und das Open Geospatial Consortium (OGC) aus dem Bereich der geographischen Informationen sowie deren Standards erläutert. Aus dem Bereich der Computer Science müssen vor allem die verschiedenen Kommunikationsprotokolle definiert werden mit denen Client und Server kommunizieren. Weiter sollen im Kapitel der Grundlagen aus dem Zusammenhang der Arbeit mit Webservices in verschiedenen Onlineinfrastrukturen, das dazugehörige Konzept der Service Orientierten Architektur (SOA) und die damit in Verbindung stehenden Webtechnologien definiert werden. Von den allgemeinen Webservices wird anschließend der Übergang zu den speziellen Geo-Webservices geschaffen. Schließlich wird nochmal intensiv auf die Geoverarbeitungsagenda nach Brauner et al. (2009) eingegangen und der bisherige Stand der Forschung und dazu verwandte Arbeiten beschrieben.

Im Kapitel „Offline Geoprocessing“ wird zunächst der Anwendungsfall aus dem Gebiet der Standortanalyse und das daraus entstandene Modell erläutert. Das Modell sowie dessen einzelne Module werden mit der Software ArcGIS der Firma ESRI (Environmental Systems Research Institute) erstellt. Die Arbeitsschritte und die benutzten Werkzeuge werden hier ebenso erläutert wie letztendlich auch das Ergebnis der einfachen Ausführung des erstellten Geoverarbeitungsmodells.

Im Kapitel „Online Geoprocessing“ wird dann die Online Bereitstellung, die Orchestrierung und schließlich die Performance erörtert. In der Schlussbetrachtung wird schließlich das Projekt im Ganzen betrachtet und auf Basis der zugrunde liegenden Forschungsfragen bewertet.

2 Grundlagen

2.1 Geographische Informationssysteme

In dieser Arbeit soll, wie in der Einleitung beschrieben, ein Modell zur Verarbeitung von Geodaten entstehen, welches nicht nur stationär auf einem Computer, sondern auch in einer Online Umgebung zur Verfügung steht. Die Grundlage für die Verarbeitung von geographischen Informationen/Geodaten bildet ein **Geographisches Informationssystem** oder auch **Geoinformationssystem (GIS)**.

„Ein Geoinformationssystem ist ein rechnergestütztes System, das aus Hardware, Software, Daten und den Anwendungen besteht. Mit ihm können raumbezogene Daten digital erfasst, gespeichert, verwaltet, aktualisiert, analysiert und modelliert sowie alphanumerisch und graphisch präsentiert werden“ (Lange, 2006, p.320).

Nach Bartelme (2005, p.64 ff) besteht ein GIS aus vier strukturellen Komponenten: Hardware, Software, Daten und Anwender.

- Die **Hardware** umfasst das Computersystem mit Prozessor, Speichermedien, Peripheriegeräten und Vernetzung.
- Die **Software** beschreibt Programmsysteme mit Werkzeugen zur Erfassung, Verwaltung, Analyse und Präsentation der Informationen. Dabei ist ArcGIS das bekannteste Beispiel und wird auch in dieser Arbeit verwendet. Daneben existieren noch viele weitere, u.a. auch Open Source GI Systeme wie QGIS und GRASS (Geographic Resources Analysis Support System).
- Die **Daten** umfassen quantitative und qualitative Informationen, die zusammen eine Abbildung der realen Welt darstellen. In Geoinformationssystemen werden Objekte aus der realen Welt in digitaler Form dargestellt. Solche Objekte mit einem Lagebezug zur Erde werden als Feature bezeichnet. *“A feature is an abstraction of a real world phenomenon” (OGC, 2009, p.1).* Dabei kann in zwei Datenmodelle unterschieden werden. **Vektordaten** bestehen aus einfachen Geometrien Punkten, Linien oder Polygonen und einer Topologie. Diesen einfachen geometrischen Elementen/Features sind Eigenschaften/Attribute zugeordnet. **Rasterdaten** wie beispielsweise digitale Höhenmodelle oder Satellitenbilder bestehen aus einzelnen Pixeln denen jeweils ein bestimmter Wert zugeordnet ist. Beide Datenmodelle können sowohl zwei als auch dreidimensional sein.
- Der **Anwender** gibt schließlich die Fragestellungen und Anwendungen vor (Lange, 2006, p.320).

Von den genannten vier funktionalen Aspekten aus dem Softwarebereich Erfassung, Verwaltung, Analyse und Präsentation steht die Analyse von Geodaten und damit auch die Geoverarbeitung/Geoprocessing mit einem Geoinformationssystem in dieser Arbeit im Vordergrund.

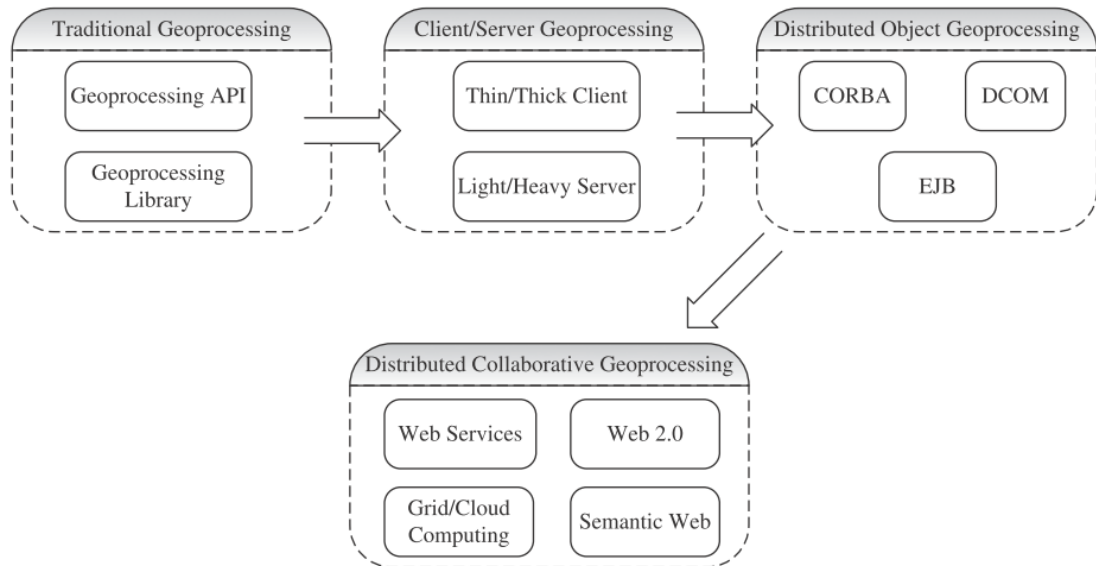
„*Geoprocessing is an operation used to manipulate spatial data*“ (Zhao, Foerster and Yue, 2012, p.4). **Geoprocessing oder Geoverarbeitung** ist nach Zhao et al. (2012) also eine Operation um räumliche Daten zu manipulieren. Lange (2006, p.341 ff) beschreibt drei große Gruppen räumlicher Analysefunktionen. Bei der Generierung von Zonen, sogenannten Buffer-Funktionen, wird um ein Geobjekt eine Fläche in einer festgelegten Entfernung generiert, wobei die alten Daten unverändert bleiben und neue generiert werden. Beispiele finden sich in der Standortanalyse oder in der Bauleitplanung, wo bestimmte Abstände zu Schutzgebieten o.ä. vorgeschrieben sind. Bei der Verarbeitung von Grenzen, sogenannten Boundary-Funktionen, wird die Geometrie der Daten verändert indem z.B. Teilbereiche ausgestanzt werden. Dabei bleiben die Attribute der Daten unverändert. Bei räumlichen Überlagerungen und Verschneidungen, sogenannten Overlay-Funktionen, können sowohl Geometrie als auch die Sachdaten miteinander verknüpft werden und so neue Daten entstehen.

Weiter ist auch eine Kombination verschiedener Werkzeuge möglich, welche in speziellen Modellen umgesetzt wird. In diesem Zusammenhang beschreibt ein Modell eine graphische Repräsentation einer sich wiederholenden Aufgabe eines speziellen Geoverarbeitungswerkzeugs. So ein **Geoverarbeitungsmodell** kann in verschiedenen Formen vorkommen. In graphischen Modellen kann eine Verarbeitungssequenz sehr einfach dargestellt werden. Schritt für Schritt können Parameter, In- und Output sowie die verschiedenen Werkzeuge und deren Verbindungen dargestellt werden. Beispiele dafür sind Flussdiagramme, Unified Modeling Language (UML) Schemata und Modelle des ESRI ModelBuilders, welcher auch in dieser Arbeit zum Einsatz kommt. Modelle in Form von Skripts sind Geoverarbeitungsbefehle in einer bestimmten Programmiersprache wie Python geschrieben. So können Modelle des ESRI ModelBuilders beispielsweise auch in ein Python Skript exportiert werden. Einzelne Geoverarbeitungswerkzeuge oder ganze Geoverarbeitungsmodelle können nun auch als ein sogenannter **Geoverarbeitungsservice** bereitgestellt werden, was im Laufe dieser Arbeit umgesetzt werden soll. Zusätzlich dazu können beispielsweise auch Web Mapping Services (WMS) oder Web Feature Services (WFS) mit eingebunden werden (Gehlot and Verbree, 2006, p.2). Wie Geoverarbeitungswerkzeuge als Service standardisiert in verschiedenen Onlineinfrastrukturen bereitgestellt werden können, definiert der OGC Web Processing Service (WPS).

Diese verschiedenen Formen der Geoverarbeitung auf verschiedenen Ebenen stellen auch verschiedene Entwicklungsschritte dar. Abbildung 2: Entwicklung von Geoprocessing

zum Geoprocessing Web beschreibt die Entwicklung vom traditionellen Desktop Processing zum sogenannten Geoprocessing Web. Dieser Begriff ist einer von vielen aus dem Bereich der Online Infrastrukturen und wird im Folgenden wie auch der allgemeine Kontext, genauer erläutert.

Abbildung 2: Entwicklung von Geoprocessing zum Geoprocessing Web



Quelle (Zhao, Foerster and Yue, 2012, p.4)

2.2 Onlineinfrastrukturen

Durch die steigende Menge räumlicher Daten im Internet steigt auch die Zahl der Einrichtungen die Zugang zu diesen Daten ermöglichen, was wiederum neue Möglichkeiten für die Wissenschaft eröffnet (Hofer, 2013, p.750). Für diese Art der Einrichtungen, welche sich mit räumlichen Daten und Online Processing beschäftigen, gibt es verschiedene Begriffe aus dem Bereich der Online Infrastrukturen. Um den Gesamtzusammenhang besser zu verstehen sollen die vielen Begriffe aus der Welt der Onlineinfrastrukturen in diesem Kapitel zunächst genauer definiert werden.

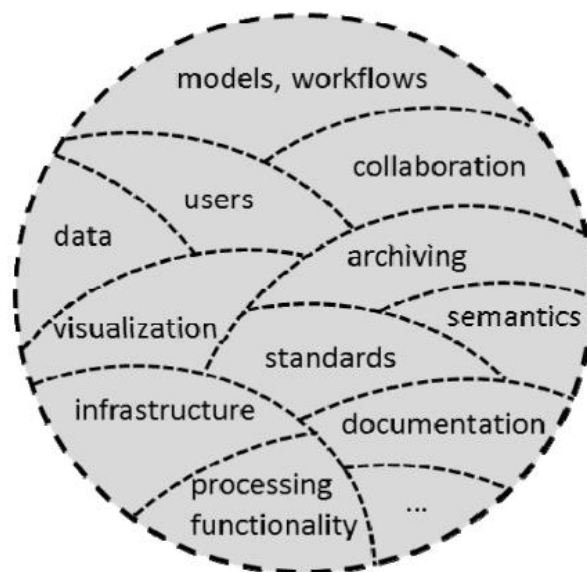
Generell kann es sich bei einer Infrastruktur um vieles handeln. Es ist ein Mix aus technischen und sozialen Themen, die eine Infrastruktur bilden. Immaterielle Dinge wie Organisationen, Institutionen, Standards, Gesetze und Märkte auf der einen und Technologien verschiedenster Art auf der anderen Seite (Poore, 2011, p.5511).

Online Geoprocessing in den verschiedenen Arten von Infrastrukturen hat viele Facetten. Die Konstruktion wissenschaftlicher Analysen durch Experten, das Bereitstellen vordefinierter Funktionen für größere Benutzergruppen, die Entwicklung spezifischer Services zur Automatisierung sich wiederholender Arbeitsabläufe und die Nutzung mächtiger, bereitgestellter Verarbeitungsinfrastrukturen (Hofer, 2013, p.750). Das Hauptziel von Online Geoprocessing Umgebungen bleibt die Unterstützung der Nutzer um Informationen und Wissen aus den bereitgestellten Daten zu erlangen (Hofer, 2013, p.752).

2.2.1 E-Science und Cyberinfrastructure

E-Science beschreibt das generelle Prinzip Wissenschaft interdisziplinär, online und in bereitgestellten, zusammenarbeitenden Umgebungen umzusetzen (Hofer, 2013, p.750). De Jesus (2012, p.75) beschreibt E-Science auch als „umbrella concept“, das alle Internettechnologien abdeckt, es geht also weniger um die zugrunde liegende Infrastruktur (Stewart et al., 2010, p.42). Die Forschung und Weiterentwicklung aller folgenden Arten von Onlineinfrastrukturen sind somit der E-Science zuzuordnen. Abbildung 3 zeigt eine Übersicht der verschiedenen Komponenten, die bei E-Science Umgebungen eine Rolle spielen.

Abbildung 3: E-Science Komponenten



Quelle (Hofer, 2013, p.752)

Die Umgebungen in der E-Science ausgeübt wird werden übergeordnet als Cyberinfrastructures bezeichnet und haben demnach einen stärkeren technischen Bezug. „*Cyberinfrastructure consists of computing systems, data storage systems, advanced instruments and data repositories, visualization environments, and people, all linked together by software and high performance networks to improve research productivity and enable breakthroughs not otherwise possible*“ (Stewart et al., 2010, p.43).

Hofer (2013, p.750 f) beschreibt das Vorkommen unterschiedlicher räumlicher Prägungen der beiden Begriffe E-Science und Cyberinfrastructure. So würden die beiden Begriffe oftmals auch Synonym verwendet, wobei E-Science europäisch und Cyberinfrastructure amerikanisch geprägt ist. Die Autorin stimmt jedoch mit den Definitionen von Stewart et al. (2010) überein. So wird auch in dieser Arbeit den Verständnissen der beiden Autoren gefolgt und die beschriebenen Definitionen genutzt.

Poore (2011, p.5010 f) beschreibt, dass der Fokus bei der Entwicklung von (räumlichen) Cyberinfrastrukturen auf dem Nutzer liegen sollte. Cyberinfrastrukturen sollten über den Bereich professioneller Nutzer hinausschauen und Laien als potentielle Nutzer und deren möglichen Beitrag miteinbeziehen. So können solche Cyberinfrastrukturen von den Möglichkeiten des Webs 2.0, also der Möglichkeit der Nutzer aktiv zu den Inhalten beizutragen, profitieren.

Wie schon zu Beginn der Arbeit beschrieben hat das Web Geographische Informationssysteme (GIS) einer breiteren Masse zugänglicher gemacht. Hofer (2013, p.753) zitiert in diesem Zusammenhang Schade et al. (2012) die sagen, dass die Entwicklungen aus dem Bereich der E-Science jedoch nicht in dem Umfang genutzt werden wie zu erwarten wäre. Geographische Analysen mit einem Desktop GIS sind immer noch weit verbreitet. Durch die neuen Möglichkeiten wie geographische Informationen erlangt, übertragen, veröffentlicht, geteilt und visualisiert werden (Fu, 2011, p.4 ff) haben sich auch spezielle Online Infrastrukturen gebildet, die einen räumlichen Bezug haben. Im Folgenden werden die Begriffe Geospatial Cyberinfrastructure, Geodateninfrastruktur und Geoprocessing Web erörtert.

2.2.2 Geospatial Cyberinfrastructure

„Spatial cyberinfrastructure (spatial CI) [...] combines the tools and computing technologies of CI with the power of spatial analysis to address complex environmental and social issues such as climate change, disaster response, transportation planning, and national security“ (Poore, 2011, p.5510). In Ihrer Definition beschreibt Poore die Verbindung der Eigenschaften einer traditionellen Cyberinfrastruktur mit den Möglichkeiten räumlicher Analysen um komplexe Themen aus Umwelt und Sozialem zu lösen.

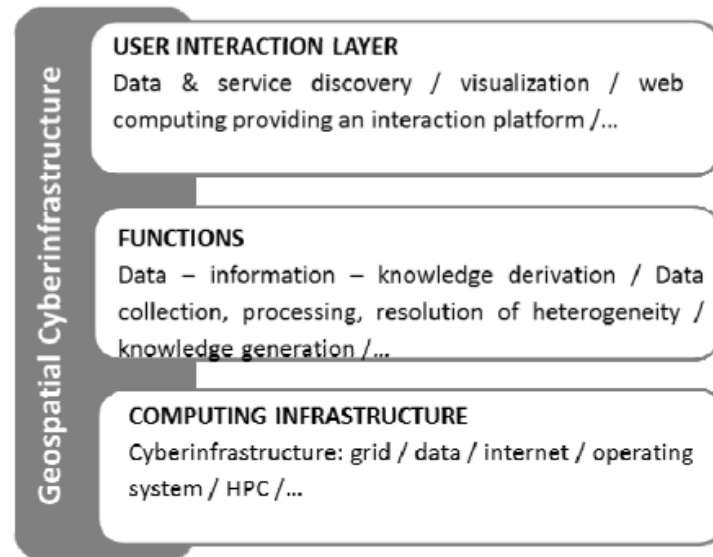
Yang et al. (2010, p.266 ff) nennen drei Rahmenbedingungen bzw. Ressourcen, aus denen sich eine GCI zusammensetzt: (1) Funktionen; (2) Communities; (3) Basistechnologien.

- (1) Die Funktionen beinhalten die grundlegenden Eigenschaften einer CI, also Computing, Networking und Hardware. Hinzukommen die räumlichen Eigenschaften. Diese Basiskomponenten stellen dem Nutzer Zugang zu räumlichen Daten, Informationen, Wissen und Verarbeitungswerkzeugen zur Verfügung.
- (2) Die Community besteht aus beteiligten Wissenschaftlern der unterschiedlichen, (geo-) wissenschaftlichen Disziplinen, aber auch aus anderen Endnutzern. Diese bringen wichtige Komponenten, wie zur Verfügung gestelltes Wissen und auch die Sammlung von Daten mit ein.
- (3) .Die Basistechnologien stellen technische Unterstützung zur Verfügung. Das Ziel dabei ist die Funktionen aufrechtzuerhalten und weiterzuentwickeln, aber auch neue zu erfinden. Dabei liegt der Blickwinkel auch auf der Sammlung der Daten und des zur Verfügung stehenden Wissens durch das semantische Web. Als Beispiel sind erdbeobachtende Sensor Netzwerke wie der OGC Sensor Observation Service (SOS) zu nennen. Yang et al. (2010, p.267) beschreiben diese als treibende Kraft für kommende Entwicklungen durch den Mehrwert der Echtzeitinformationen.

Wie bereits beschrieben betont auch Poore (2011, p.5010 f) die wichtige Rolle der Mitarbeit und Beiträgen der Nutzer und hält es für unabdingbar den Fokus bei der Entwicklung auf

diese zu legen. Nach Hofer (2013, p.751) liegen die Stärken einer räumlichen CI in den erforderlichen Technologien für Datenspeicherung und –prozessierung. Abbildung 4 fasst die Eigenschaften und Funktionen einer Geospatial Cyberinfrastruktur nochmals zusammen.

Abbildung 4: Komponenten einer Geospatial Cyberinfrastruktur



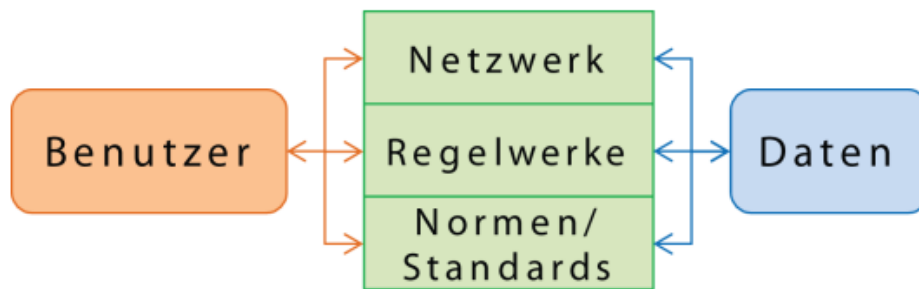
Quelle (Hofer, 2013, p.756) nach (Yang et al., 2010)

2.2.3 Geodateninfrastruktur (GDI)

Neben einer GCI ist der Begriff Geodateninfrastruktur (GDI), bzw. Spatial Data Infrastructure (SDI) sehr verbreitet. „An SDI has been defined as the technology, policies, standards, human resources, and related activities necessary to acquire, process, distribute, use, maintain, and preserve spatial data“ (Poore, 2011, p.5012).

In dieser Definition werden die Eigenschaften einer GDI gut beschrieben. Im Gegensatz zu einer GCI ist zu sehen, dass eine GDI neben den Technologien vor allem zusätzlich die Komponenten der Richtlinien und Standards aufweist. Staub (2009, p.65 f) verweist in seiner Arbeit darauf, dass mit der Schaffung von GDIs neue Möglichkeiten der Datennutzung entstehen. Die Verfügbarmachung von Daten für potenzielle Nutzer löst ein Grundproblem aktueller GIS-Anwendungen. Die Komponenten einer GDI aus Abbildung 5 sind nicht nur Bestandteile einer GDI, sondern stellen gleichzeitig auch die notwendige Voraussetzung für die Realisierung einer GDI dar.

Abbildung 5: Architektur einer Geodateninfrastruktur



Quelle (Staub, 2009, p.66)

Poore (2011, p.5012) greift Craglia und Johnston (2004, p.17) auf und verweist darauf, dass GDIs oft nicht interoperabel sind. So werden fehlende Daten, schlechte Dokumentation, Inkompatibilität, schlechter Zugang und schlechte Benutzbarkeit beschrieben. Seit der Studie von Crompvoets und Bregt (2003) ist jedoch einige Zeit vergangen, weshalb vor allem durch die INSPIRE Initiative große Fortschritte gab. Nach Hofer (2013, p.750) liegt der Fokus einer GDI immer noch hauptsächlich auf der Bereitstellung von Geodaten. In den letzten Jahren sind jedoch auch mehr und mehr Analysefunktionen dazugekommen. Vor allem aber der technische Fortschritt in den Bereichen Geodaten Akquise, Rechenleistung, Internetprotokollen und der Verarbeitung von Geodaten haben die Entwicklung von GDIs voran gebracht. So ist die Nutzung solcher Anwendungen heutzutage nicht nur am Desktop möglich sondern auch durch Apps auf Mobilgeräten (Bernard et al., 2014, p.613).

Weiter zieht Poore (2011, p.5012) einen Vergleich zu Volunteered Geographic Information (VGI), also freiwillig erhobenen geographischen Informationen, die klassischen GDIs in Sachen Abdeckung, Vollständigkeit und Genauigkeit Konkurrenz machen. Dies belegen auch anderen Arbeiten von Coleman et al. (2009) und Neis et al. (2010). Das bekannteste Beispiel aus diesem Bereich ist die Open Street Map (OSM).

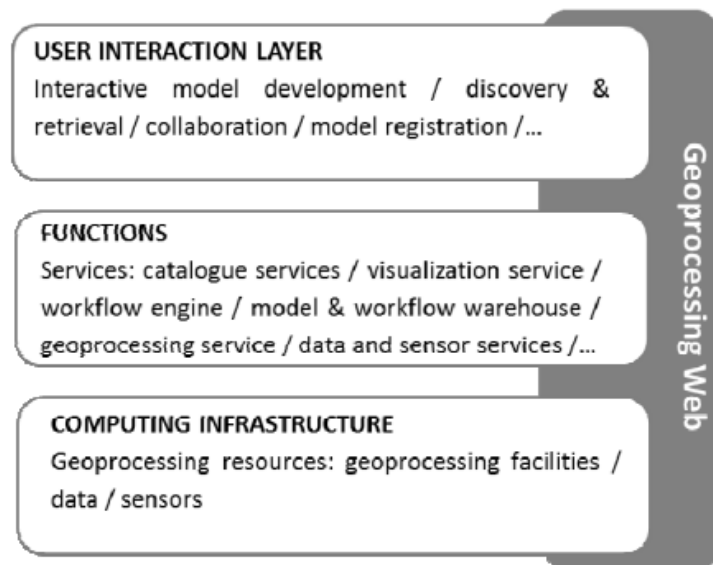
Ein Beispiel für ein Geodateninfrastruktur ist die GDI-DE. Diese ist in Projekt von Bund, Ländern und Kommunen bei dem raumbezogene Daten vernetzt über das Internet bereitgestellt werden. Sie ist ebenfalls Teil einer europäischen Geodateninfrastruktur, die über die bereits erwähnte EU-Richtlinie INSPIRE umgesetzt wird (Bundesamt für Kartographie und Geodäsie, 2014a). INSPIRE trat am 15. Mai 2007 in Kraft und steht für „Infrastructure for Spatial Information in the European Community“. Es soll die grenzübergreifende Nutzung von Geodaten in Europa erleichtern und ist ein Vorhaben für eine gemeinsame Geo-

dateninfrastruktur in Europa. INSPIRE fordert eine einheitliche Beschreibung und ein einheitliches Format der Geodaten sowie deren Bereitstellung im Internet, mit Diensten für Suche, Visualisierung und Download (Bundesamt für Kartographie und Geodäsie, 2014b).

Bernard et al. (2014, p.616) führen den Begriff "Scientific Geodata Infrastructure" ein. Neben der traditionellen Auswertung wissenschaftlicher Arbeit, die auf Zitationen veröffentlichter Artikel und deren Einfluss beruht, werden produzierte Daten und entwickelte Methoden in Form von Software Tools als Hauptteil der wissenschaftlichen Aktivität den Autoren nach oft vernachlässigt. Dabei gehen diese oft über den eigentlichen Zweck hinaus und können, bei ausreichender Beschreibung und Zugänglichkeit, wertvoller Input für andere Wissenschaftler sein. Dieser Prozess und Austausch würde die Effizienz und Nachhaltigkeit wissenschaftlicher Arbeit deutlich erhöhen und auch interdisziplinäre Wissenschaft fördern. Im Bereich der Geodaten muss dabei vor allem auf die Pflege der Metadaten geachtet werden um Genauigkeit und Qualität zu gewährleisten. Dabei spielt auch die Dokumentation der Herkunft und die Abstammungslinie eine wichtige Rolle (Lineage/Provenance). Diese Informationen über den Ursprung der Daten könnten eine ähnliche Rolle einnehmen wie Zitationen um den Einfluss wissenschaftlicher Arbeit zu messen.

Im Gegensatz zu einer räumlichen CI bezieht sich der Begriff **Geoprocessing Web** vor allem auf die Bereitstellung verschiedener Komponenten für Modellentwicklung und -austausch. Alles in allem liegen die Unterschiede der beiden Begriffe nur in verschiedenen Akzenten und überlappen in ihren Zielen, Konzepten und Technologien (Hofer, 2013, p.749). Ein wichtiger Unterschied ist, dass das „Geoprocessing Web“ ausschließlich auf Webtechnologien basiert und ihren Nutzen aus offenen Webstandards zieht (Hofer, 2013, p.751). Beide Arten der Infrastrukturen haben gemeinsam, dass sie die Zusammenarbeit von Wissenschaftlern in diesem Gebiet verbessern wollen (Hofer, 2013, p.751). Abbildung 6 fasst die Eigenschaften und Funktionen des Geoprocessing Web nochmals zusammen.

Abbildung 6: Komponenten des Geoprocessing Web



Quelle (Hofer, 2013, p.757) nach (Zhao, Foerster and Yue, 2012)

Abschließend zur Betrachtung der verschiedenen Onlineinfrastrukturen ist zu sagen, dass Geoverarbeitung grundsätzlich in allen genannten Infrastrukturen stattfinden kann. Bei dem im Zuge dieser Arbeit entstandenen, online bereitgestellten Workflow handelt es sich um eine Umsetzung des Geoprocessing Web.

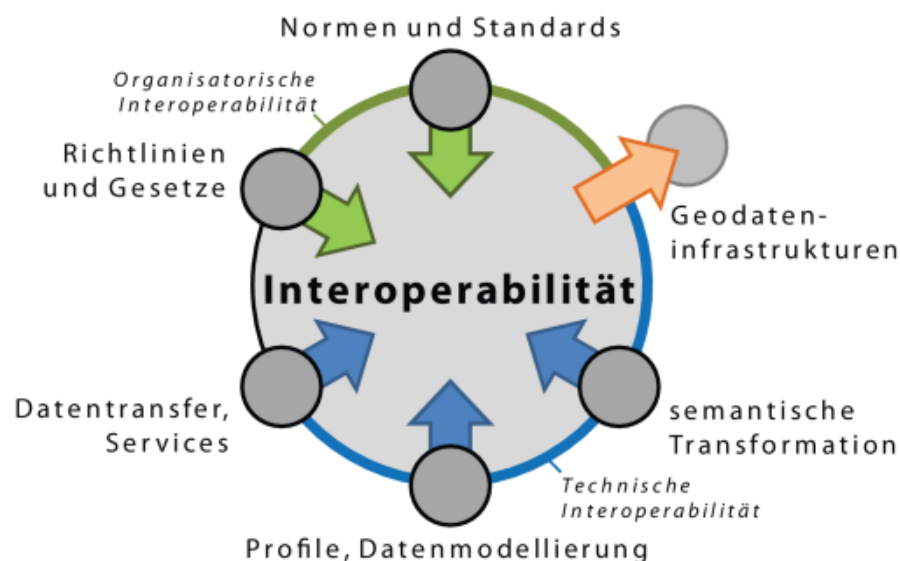
2.3 Interoperabilität durch Standards

Um Geoverarbeitung in einer Online Infrastruktur durchführen zu können spielt Interoperabilität eine Hauptrolle. „*Interoperabilität ist die Fähigkeit von Softwarekomponenten zu einer selbstständigen Zusammenarbeit über Software-, System- und Institutionsgrenzen hinweg. Softwarekomponenten sind interoperabel, wenn sie ohne Mitwirkung des Nutzers miteinander kommunizieren können. Sie tun dies über standardisierte Schnittstellen und genormte Übergabeformate. Schnittstellen wiederum sind Vereinbarungen über den Informationsaustausch zwischen Softwarekomponenten*“ (Andrae, 2012, p.15).

Weiter kann noch zwischen syntaktischer und semantischer Interoperabilität unterschieden werden. Staub (2009, p.24 f) definiert syntaktische Interoperabilität auch als Interoperabilität der Systeme wie sie bei OWS vorkommt. Sie erlaubt die interoperable Nutzung verfügbarer Daten über eine Standardschnittstelle, welche durch ein standardisiertes Protokoll angesprochen wird. Anschließend werden die Informationen in einem Standardformat zurückgegeben. Dabei erfolgt Abfrage und Auslieferung der Daten in der Struktur des Anbietermodells, wobei die Datenstruktur der verfügbaren Daten vom Nutzer nicht beeinflussbar ist. Dies könnte ein Nachteil sein, wenn die Struktur nicht zu jener des Nutzersystems passt.

Semantische Interoperabilität hingegen definiert Staub (2009, p.25 f) auch als Interoperabilität der Konzepte. Diese wird durch ein modellbasiertes Transferverfahren realisiert, wobei mittels semantischen Modelltransformationen die Datenstrukturen derart umgebaut werden, dass der Nutzer die passende Datenstruktur erhält.

Abbildung 7: Charakteristika von Interoperabilität



Quelle (Staub, 2009, p.20)

In der OpenGIS Abstract Specification „OpenGIS Service Architecture“ (OGC, 2002, p.7) ist auch von geographischer Interoperabilität die Rede: *“Geographic interoperability” is the ability of information systems to 1) freely exchange all kinds of spatial information about the Earth and about the objects and phenomena on, above, and below the Earth’s surface; and 2) cooperatively, over networks, run software capable of manipulating such information“.*

Interoperabilität gehört seit der Entwicklung von GIS zu den größten Herausforderungen. Nach Fu (2011, p.68f) haben sich dabei über die Jahre sechs Phasen der Implementierung und Umsetzung herausgebildet:

- 1) Konvertierung von Daten
- 2) Standardisierter Austausch von Geodaten
- 3) Offene Datenformate
- 4) Direkter Zugriff auf APIs
- 5) Einheitliche Objekte in Datenbankmanagementsystemen (DBMS)
- 6) Standardisierte GIS Webservices

Um Interoperabilität zu gewährleisten sind einheitlichen Standards und Normen erforderlich. Hierzu sollen die im Bereich des World Wide Web und der raumbezogenen Informationen, standardgebenden Organisationen vorgestellt werden, die für diese Arbeit eine entscheidende Rolle spielen.

Standardisierung im Bereich der Geoinformatik ist ein Zusammenspiel verschiedener Organisationen. Eine allgemeine, übergeordnete Rolle spielt die **International Standardisation Organisation (ISO)**. Die ISO ist eine unabhängige Vereinigung von im Moment 165 nationalen Normierungsorganisationen. Für Deutschland ist das Deutsche Institut für Normung (DIN) Mitglied in der ISO (ISO, 2014). Die Normen der Serie 19100 bilden die Grundlage für den Bereich der Geoinformation und werden vom Technischen Komitee 211 der ISO herausgegeben. Sie beschreiben Methoden, Werkzeuge und Dienste zur Erfassung, Verwaltung, Analyse, und der Übertragung von Daten in elektronischer Form zwischen unterschiedlichen Benutzern und Systemen. In der ISO Norm 19109 ist z.B. das allgemeine Anwendungsschema definiert, das für die Definition von Datenmodellen, wie der verschiedenen webbasierten Geodienste, auch von der OGC herangezogen wird. Weiter sind mittlerweile z.B. auch die OGC Normen Web Map Service (WMS), ISO Norm 19128, und Geography Markup Language (GML), ISO Norm 19136, übernommen worden (Geodaten Infrastruktur Brandenburg, 2005).

Bei dem **World Wide Web Consortium (W3C)** handelt es sich um eine internationale Gemeinschaft bestehend aus den Mitgliedsorganisationen, vor allem große Unternehmen und verschiedene wissenschaftliche Einrichtungen, Verwaltung aber auch der Öffentlichkeit, die Zusammenarbeiten und Web Standards entwickeln. Direktor des W3C ist Tim Berners Lee, der Erfinder des World Wide Web (WWW) (W3C, 2014). Die wichtigsten Webstandards des W3C in Zusammenhang mit dieser Arbeit sind HTTP, XML, SOAP und WSDL.

Das **Open Geospatial Consortium (OGC)**, gegründet 1994 unter dem Namen OpenGIS Consortium, ist die standardgebende Institution im Geo-Bereich und kümmert sich um die Interoperabilität der Softwarekomponenten. Dabei spielt auch die Integration geospezifischer Software in allgemeine IT-Technologien, also das Zusammenspiel mit dem W3C eine wichtige Rolle (Andrae, 2012, p.15). Staub (2009, p.42) Grundsätzlich sind zwei Typen von OGC-Standards zu unterscheiden, zum einen die Abstract Specification als Referenzmodell oder konzeptionelles Fundament für die Entwicklung von OGC-Standards und zum anderen die Implementation Specifications, die Schnittstellen von Systemen und Sprachen definieren (OGC, 2015). Die wichtigsten Standards im Zusammenhang mit dieser Arbeit sind dabei Webbasierte Geodienste wie WMS, WFS und WPS.

2.3.1 Basis Webtechnologien des W3C

Die nachfolgenden Definitionen sollen vor allem dem Verständnis dieser Arbeit dienen. Die Bedeutung und Funktion einiger dieser Grundlagen kann jedoch auch über das beschriebene hinausgehen und wird teilweise verkürzt dargestellt. Dazu wird vor allem den Erklärungen und Ausführungen von Fu (2011, p.26 ff) gefolgt, der die Zusammenhänge vor allem in Verbindung mit der Anwendung von Geoinformationssystemen im Web erläutert.

Einfache Anwendungen im World Wide Web basieren auf einer **Client-Server-Architektur**. Im einfachsten Fall kann eine solche Architektur eine Zwei-Schichten-Architektur sein. Im Normalfall handelt es sich jedoch um eine Drei-Schichten-Architektur. Diese besteht aus Präsentationsschicht, Logikschicht, und der Datenhaltungsschicht. Die Präsentationsschicht beinhaltet den Client, der typischerweise ein Webbrowser ist. Die Logikschicht besteht aus dem Webserver und die Datenhaltungsschicht aus einer Datenbank. Bei einer Zwei-Schichten-Architektur wird die Logik von dem Client übernommen. Es fehlt also ein Webserver.

Der typische Verlauf einer Webanwendung ist, dass ein Benutzer über einen Webbrowser eine Anfrage an einen Webserver schickt indem er eine URL in die Adresszeile eingibt. Der Webserver empfängt diese Anfrage, macht auf Basis der URL das dazugehörige Dokument oder Skript in der Datenbank ausfindig und gibt das Ergebnis an den Client zurück. Die Antwort ist typischerweise im HTML-Format. Der Webbrowser empfängt die Antwort vom

Webserver und präsentiert diese dem Benutzer. Diese Architektur beruht auf den Herzstücken des World Wide Web (WWW), den W3C Standards **HTTP, URL und HTML**.

Das **Hypertext Transfer Protocol (HTTP)** definiert Regeln und Prozeduren nach denen die unterschiedlichen Akteure im Web miteinander kommunizieren. Es sind insgesamt acht Anfragemethoden definiert: GET, POST, HEAD, PUT, DELETE, TRACE OPTIONS und CONNECT. Die bekanntesten und am meisten genutzten Methoden sind HTTP-GET und HTTP-POST. Alle Anfragen und Antworten bestehen aus einem Header und einem Body. In dem Header wird der Inhalt der Nachricht definiert. Neben HTML/Text können verschiedene Datenformate wie Bild- oder Audiodateien übertragen werden, was HTTP flexibel macht. Weiter ist das Protokoll auch als zustandslos beschrieben, was bedeutet, dass die Verbindung zwischen Client und Server nach der Übertragung verloren geht und vergessen wird.

Uniform Resource Locator (URL) ist eine Unterart eines Uniform Resource Identifiers (URI) und spezifiziert wo eine Ressource im Web zu erreichen ist und wie darauf zugegriffen wird. Umgangssprachlich wird URL auch als Webadresse beschrieben. Das am häufigsten benutzte Übertragungsprotokoll ist das beschriebene HTTP.

Hypertext Markup Language (HTML) ist eine Auszeichnungssprache zur Erstellung von Webseiten. HTML Dokumente bestehen aus dem Inhalt, Layout- und Formatierungsinformationen. Wenn eine Webseite im Webbrowser geladen wird, interpretiert dieser den HTML-Code stellt die Webseite dementsprechend dar.

Damit Client und Server miteinander kommunizieren können sind zusätzlich sowohl clientseitige, als auch die serverseitige Anwendungen vonnöten. Serverseitige Anwendungen sind Computerprogramme wie Apache oder Tomcat. Neben der Software gehören auch entsprechende Programmiersprachen zu den serverseitigen Anwendungen. Clientseitige Anwendungen beschreiben zumeist Browsertechnologien, beinhalten aber auch Desktop und mobile Anwendungen, sowie ebenfalls entsprechende Programmiersprachen wie z.B. JavaScript.

Im Zuge der Beschreibung von Basiswebtechnologien ist es zusätzlich noch wichtig Datenaustauschformate, und damit die **Extensible Markup Language (XML)**, zu erwähnen. XML ist eine Auszeichnungssprache die es erlaubt eigene Tags und Attribute zu definieren. Weil es ein reines Text Dokument ist, ist es plattformunabhängig und selbstbeschreibend. Aufgrund dieser Vorteile ist es das am meisten benutzte Datenaustauschformat. Auf der anderen Seite gibt es jedoch auch Nachteile. Im Gegensatz zu dem Datenaustauschformat JSON (JavaScript Object Notation) ist es aufgrund der Größe des Textdokuments weniger effizient.

2.3.2 Web-Services und Service orientierte Architekturen

Neben den beschriebenen Basiswebtechnologien spielen heutzutage sogenannte **Web-services** eine immer größere Rolle. *„A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format [...]. Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards“* (W3C, 2004a, p.7).

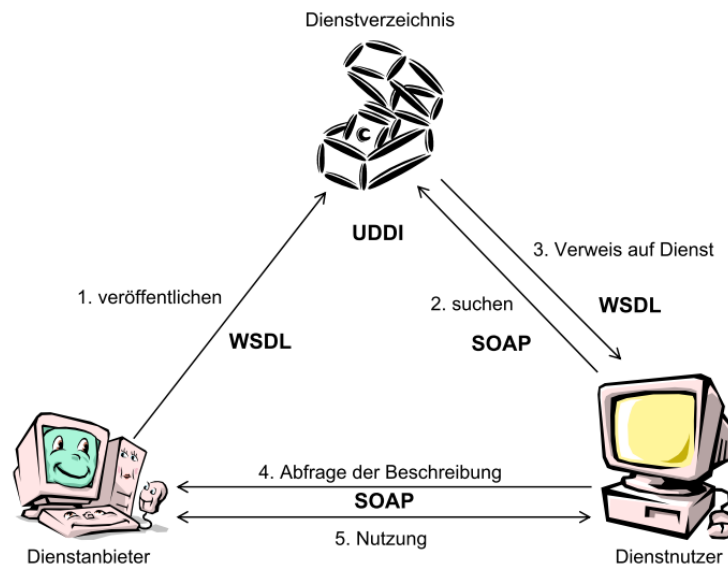
Webservices stellen die Basis **serviceorientierter Architekturen (SOA)** dar. *„Unter einer SOA versteht man eine Systemarchitektur, die vielfältige, verschiedene und eventuell inkompatible Methoden oder Applikationen als wiederverwendbare und offen zugreifbare Dienste repräsentiert und dadurch eine plattform- und sprachenunabhängige Nutzung und Wiederverwendung ermöglicht“* (Melzer, 2010, p.13). Der Vorteil liegt darin, dass einzelne Webservices voneinander unabhängige Funktionen zur Verfügung stellen können. Diese basieren auf verschiedenen Eingabe Parametern und den entsprechenden Ausgabe Parametern. Webservices können über das Internet verteilt zur Verfügung gestellt werden und lassen sich zu komplexen Prozessen kombinieren. Demnach können Anwendungen auf der Komposition verschiedener Webservices beruhen. Melzer (2010, p.63) beschreibt die unterschiedlichen Ansätze verschiedener Definitionen. Dabei stellt er, dass alle Definitionen die Aussage gemeinsam haben, dass es sich bei Webservices um eine Technik zur Maschine-Maschine Kommunikation handelt.

Alameh (2003, p.29) zitiert in diesem Zusammenhang Abel et al. (1998) die bereits zur damaligen Zeit beschrieben, dass das Modell der Webservices auch im GIS Bereich dem Benutzer freie Kombinationen ermöglicht um individuelle Lösungen mit wenig Programmier-, Integrations- und Instandhaltungsaufwand zu schaffen. Das und eine effektive Verkettung verschiedener Services sind die Schlüsselfaktoren um GIS über die traditionellen Grenzen hinaus zu erweitern. Dies beschreiben auch Stollberg und Zipf (2007, p.1): *„The aggregation of web services in order to achieve a common goal is a basic concept in Service Oriented Architectures (SOA).“*

Webservices bestehen grundsätzlich aus drei Komponenten. Der **Provider/Dienstanbieter** stellt den Service bereit, der **Consumer/Dienstnutzer** nutzt den Service und die **Registry/Dienstverzeichnis** fungiert als Vermittler der anderen beiden Komponenten. Dabei müssen Dienstanbieter und Dienstnutzer nicht zwangsläufig voneinander wissen. Der Dienstanbieter kann seine Verfügbarkeit im Dienstverzeichnis registrieren und der Dienst-

nutzer kann das Dienstverzeichnis nach den benötigten Services durchsuchen (siehe Abbildung 8: Webservice-basierte SOA) (Fu, 2011, p.52). Diese drei Komponenten werden mithilfe der Spezifikationen SOAP, WSDL und UDDI beschrieben. **SOAP** (Simple Object Access Protocol) beschreibt das XML-basierte Nachrichtenformat der Kommunikation und dessen Einbettung in ein Transportprotokoll. **WSDL** (Web Services Description Language) ist die XML-basierte Beschreibungssprache, um Webservices zu beschreiben und **UDDI** (Universal Description, Discovery and Integration protocol) beschreibt einen Verzeichnisdienst für Webservices. Es spezifiziert eine standardisierte Verzeichnisstruktur für die Verwaltung von Webservice Metadaten, dazu gehören wie beschrieben die benötigten Informationen zum Auffinden der Webservices (Melzer, 2010, p.63).

Abbildung 8: Webservice-basierte SOA



Quelle (Melzer, 2010, p.64)

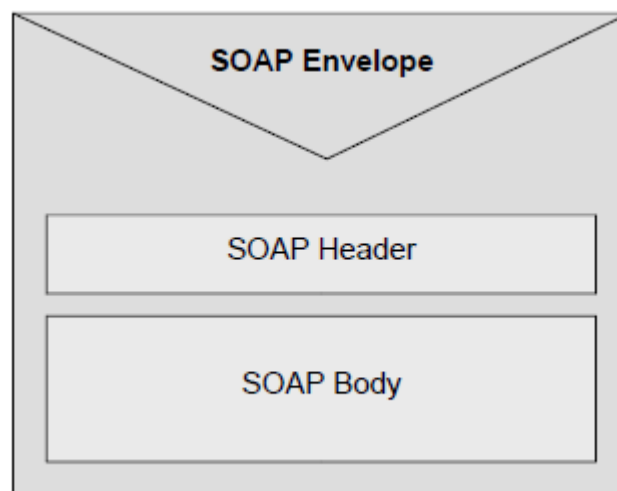
Auf diese Webservice Architekturen aufbauend haben sich auch viele Web GIS Technologien entwickelt. In der Rolle als Dienstanbieter haben sich Server Produkte wie vor allem ArcGIS Server entwickelt, mit denen raumbezogene Webservices bereitgestellt werden können. Dienstverzeichnisse können verschieden Geoportale wie z.B. das ArcGIS Services Directory sein. Die Metadaten dort registrierter Services können wiederum über allgemeine Webportale wie Google, Bing oder Yahoo indexiert sein und so von vielen Nutzern gefunden werden. Produkte für Dienstanutzer können spezielle Geobrowser, wie Google Earth oder der ArcGIS Explorer, sein. Eine Sonderrolle können Desktop GIS Softwareprodukte wie ArcGIS Desktop einnehmen. Zum einen können GIS Ressourcen vorbereitet werden um diese als Webservice auf einem ArcGIS Server zu veröffentlichen, zum anderen können

auch viele Arten von Webservices konsumiert werden (Fu, 2011, p.53 f). Funktionen und Beispiele von bestimmten räumlichen Webservices werden im Kapitel 2.3.4 in Bezug auf die verschiedenen OGC Standards genannt und erläutert. Bezogen auf diese Arbeit und den Bereich der geographischen Informationen können Webservices in zwei Typen eingeteilt werden: SOAP basierte Webservices und REST basierte Webservices.

2.3.2.1 SOAP basierte Webservices

Wie bereits beschrieben steht der Begriff SOAP für Simple Object Access Protocol und es handelt sich um ein XML-basiertes Nachrichtenformat zur Kommunikation mit Webservices. Heutzutage wird jedoch nur noch die Abkürzung SOAP verwendet, da Simple Object Access Protocol eigentlich ein irreführender Begriff ist. Es ist weder ein einfaches Protokoll und auch nicht zum Zugriff auf Objekte geeignet. Da der Begriff sich jedoch zur damaligen Zeit bereits verbreitet hatte blieb man bei der Abkürzung SOAP (Melzer, 2010, p.84). SOAP basierte Webservices nutzen HTTP-Post Befehle um Anfragen zu versenden. Diese Anfragen sind in SOAP gekapselte XML Dokumente (Fu, 2011, p.61 f). Abbildung 9 zeigt den schematischen Aufbau einer SOAP Nachricht.

Abbildung 9: Aufbau einer SOAP Nachricht



Quelle (Melzer, 2010, p.87)

SOAP ist also eine Anwendung von XML und definiert ein plattform- und programmiersprachenunabhängiges Konzept zur Übertragung von Daten, mit der prinzipiell beliebige XML Dokumente übertragen werden können. Eine vollständige SOAP-Nachricht muss aus den Hauptelementen Envelope und Body sowie einem optionalen Header-Element bestehen. Der Envelope ist das Wurzelement jeder SOAP-Nachricht sowie der Rahmen in dem die Nachricht beschrieben und als SOAP Nachricht identifiziert wird. Header und Body sind die

Kind Elemente (Melzer, 2010, p.114). In dem optionalen Header sind Sicherheitsinformationen zur Authentifizierung und Autorisierung enthalten. Der Body enthält die eigentlichen Inhalte. Dies können Informationen zum Datenaustausch, oder Anweisungen für einen entfernten Prozeduraufruf sein (Baun, Kunze, Nimis and Tai, 2011, p.25).

Die Schnittstelle eines Webservices wird im Fall von SOAP durch WSDL (Web Service Description Language) beschrieben. WSDL ist eine Erweiterung der XML Schema Spezifikation und ermöglicht sowohl die abstrakte Beschreibung von Schnittstellen mit ihren Operationen, als auch die konkrete Beschreibung technischer Informationen zum Aufruf des Dienstes (Melzer, 2010, p.115).

2.3.2.2 REST basierte Webservices

Ursprünglich wurde REST (Representational State Transfer) von Roy Fielding im Rahmen seiner Dissertation definiert (Fielding, 2000). Diese Definitionen gehen jedoch über den Rahmen dieser Arbeit hinaus. Wichtig ist im Zusammenhang mit dieser Arbeit, dass es sich bei REST nicht um ein Protokoll sondern um einen Architekturstil handelt, der definiert wie existierende Web-Protokolle verwendet werden können, um unter anderem möglichst einfach Webservices zu realisieren. Während SOAP, durch seine Unabhängigkeit vom Transportmedium und verschiedene Erweiterungsmöglichkeiten, viele verteilte Anwendungen unterstützt, ist REST auf Anwendungen im World Wide Web begrenzt. REST basierte Webservices werden auch als RESTful bezeichnet. Eine REST konforme Ressource muss mittels einer URL zugreifbar sein d.h., dass eine URL genau einen Seiteninhalt als Ergebnis einer serverseitigen Aktion darstellt, was für dynamisch erzeugte Seiten mitunter zusätzlichen Aufwand erfordert (Melzer, 2010, p.111). Fu (2011, p.63) definiert REST basierte Webservices wie folgt: *„RESTful Web services are Web services that transmit data over HTTP without an additional messaging layer such as SOAP. In the most common RESTful Webservice architecture, the client sends all parameters in the request URL.“*

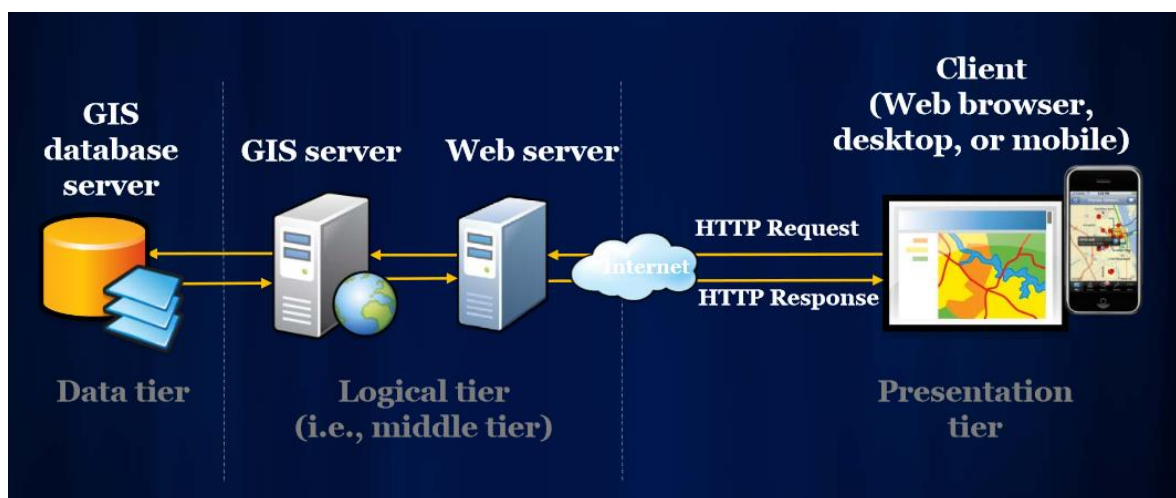
2.3.3 Web GIS Technologien

Die beschriebene Architektur der Basiswebtechnologien lässt sich nun auch auf Web GIS übertragen. *„[...] Web GIS is any GIS that uses Web technology to communicate between components“* (Fu, 2011, p.13). Die Ausführungen in diesem Kapitel beruhen ebenfalls auf dem Verständnis und den Erläuterungen von Fu (2011).

Auch Web GIS beruht auf dem Prinzip der Client-Server Architektur. Wie beschrieben sendet ein Client einen Request zum Server der diesen Request übersetzt, auf die Datenbank zugreift, eine Response erzeugt und zum Client zurück sendet. Web GIS erweitert einfache

Webanwendungen nun durch GIS Fähigkeiten. Der Benutzer nutzt eine Web GIS Anwendung durch einen Client, welcher hier ein Webbrowser, eine Desktop- oder eine mobile Anwendung sein kann. Er sendet eine Anfrage über HTTP an den Webserver, der diese an einen GIS Server weiterleitet. Der GIS Server erhält die gewünschten Daten, wie eine Karte, Geodaten oder auch eine Analyse, aus der Datenbank. Dieses Ergebnis wird über HTTP an den Client zurückgegeben, der es wiederum dem Benutzer präsentiert. Auch hier lässt sich die Drei-Schichten-Architektur erkennen. Die Datenhaltungsschicht ist die GIS Datenbank und die Präsentationsschicht der Client. Die Logikschicht besteht in diesem Fall aus Webserver und GIS Server.

Abbildung 10: Web GIS Architektur



Quelle (Fu, 2011, p.33)

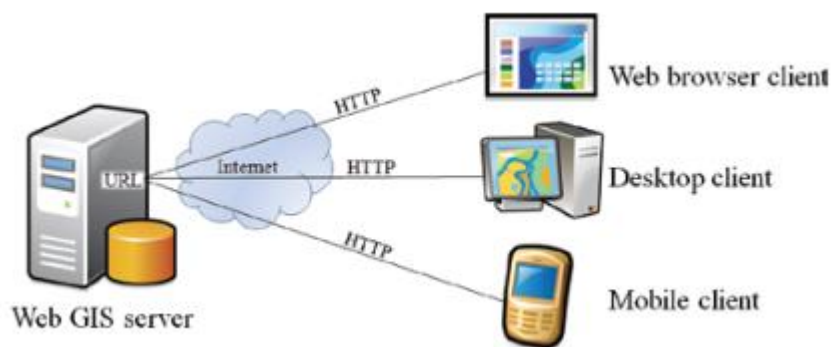
Der **Web GIS Server** ist die wichtigste Komponente eines Web GIS. Die Qualität und die Möglichkeiten eines Web GIS werden maßgeblich von diesem beeinflusst. Das bekannteste Web GIS Server Produkt ist ArcGIS Server von ESRI. Neben dieser kommerziellen Anwendung existieren auch Open Source Anwendungen wie Geoserver und Mapserver. In dieser Arbeit kommt aufgrund der weitreichenden Möglichkeiten ArcGIS Server zum Einsatz. ArcGIS Server ermöglicht u.a. die Bereitstellung von 2D Karten und 3D Globen, Such- und Bearbeitungsfunktionen sowie Routing. Neben diesen Fähigkeiten ist für diese Arbeit besonders die Möglichkeit der Bereitstellung von Geoverarbeitungsfunktionen unter Beachtung der OGC Webservice (OWS) Standards von Bedeutung. An dieser Stelle ist jedoch auch darauf hinzuweisen, dass auch die genannten Open Source Produkte die OGC Standards beachten.

In der **GIS Datenbank** werden die nötigen Daten eines Web GIS gespeichert und abgerufen. Es können sowohl Vektor- als auch Rasterdaten gespeichert werden. Möglich ist auch die Unterstützung von CAD- oder 3D-Daten sowie z.B. GPS Koordinaten. Eine gute GIS

Datenbank umfasst eine Vielzahl von Fähigkeiten, wie die Speicherung einer großen Sammlung räumlicher Daten. Es sollte räumlich, relationale Modelle, wie z.B. umfassende Regeln zur Topologie, beinhalten. Weiter sollte auch Mehrbenutzerbetrieb möglich sein, sowie die Fähigkeit zur Datensicherung und Wiederherstellung. Schließlich sollte die Performance auch bei ansteigender Datenmenge und vielen gleichzeitigen Zugriffen konstant hoch bleiben.

Web GIS Clients können verschiedene Rollen einnehmen. Zuerst stellen sie die Benutzerschnittstelle des Systems dar. Diese interagiert mit dem Benutzer, nimmt dessen Input auf, sendet die Anfrage an den Server und präsentiert das Ergebnis. Für die Anwendung eines Web GIS muss der Benutzer demnach nicht wissen, was im Back-End des Servers abläuft. Für den Benutzer stehen Performance und Benutzerfreundlichkeit im Vordergrund. Weiter ist es möglich, dass der Client auch eigentliche Aufgaben des Servers übernimmt. So können z.B. Geoverarbeitungsaufgaben statt vom Server auch vom Client übernommen werden. Übernimmt der Client viele dieser Aufgaben spricht man auch von einem **Thick Client**. Die Vorteile dieser Architektur sind die schnelle Interaktion mit dem Benutzer, da Logik oder Programm sowie auch die Daten lokal auf dem Rechner liegen. Auch ergibt sich durch diese Architektur weniger Druck auf den Server. Als Nachteile ergeben sich mögliche Schwierigkeiten durch Installation der nötigen Anwendungen oder Plugins. Weiter kann es zu Limitierungen durch Bandbreite und Rechenleistung kommen, da die übertragenen Datenmengen sehr hoch sein können. Das Gegenteil eines Thick Clients ist ein **Thin Client**. Bei dieser Architektur leistet der Server den Großteil der Arbeit. Der Client sendet lediglich die Anfrage an den Server der diese verarbeitet und an den Client zurückgibt. Die Vorteile eines Thin Clients sind, dass weder ein spezielles Programm oder Plug In benötigt wird. Der Webbrowser übernimmt alle nötigen Aufgaben. Auch wird keine hohe Rechenleistung auf Seiten des Clients benötigt. Dagegen liegt bei dieser Architektur der gesamte Druck auf dem Server, sodass dieser auch mit vielen Anfragen und großen Datenmengen klarkommen muss. Schließlich ist die Interaktion des Benutzers durch den Browser eher limitiert als bei einem ausgefeilten Programm.

Abbildung 11: Web GIS Clients



Quelle (Fu, 2011, p.13)

Wie beschrieben ist der **Client** in den meisten Fällen ein Webbrowser, kann jedoch auch eine Desktop oder eine mobile Anwendung sein. In speziellen Fällen kann auch eine Server Anwendung der Client sein, wenn diese als Client eines anderen Servers fungiert. Bei einem **Webbrowser** als Client können heutzutage auch, durch moderne Webtechnologien wie AJAX, Flex oder Silverlight, umfassende, dynamische und benutzerfreundliche Schnittstellen erzeugt werden. Ein Beispiel dafür wäre der ArcGIS Explorer Online. Dies ist eine Weiterentwicklung zu der Funktion als einfacher Client, der auf statischem HTML und einfachem JavaScript basiert. Ein Web GIS Client als **Desktop Anwendung** ist ein umfangreiches Programm, das lokal installiert ist und auf die Ressourcen des Computers zugreift. ArcGIS Desktop oder auch die Open Source Anwendung QGIS haben sich in diesem Zusammenhang von einem reinem Desktop GIS Produkt weiterentwickelt und können heutzutage auch als Web GIS Client fungieren. Die beiden Anwendungen bilden auch ein Beispiel für einen Thick Client, der viele Operationen übernehmen kann, die in einem Webbrowser nicht möglich sind. Ein weiteres Beispiel in diesem Zusammenhang ist Google Earth. Bei einer **mobilen Anwendung** als Client kann zum einen wieder der Browser als Client fungieren oder aber eine Anwendung wie z.B. ArcGIS Mobile oder Google Earth Mobile (Fu, 2011, p.33 ff).

2.3.4 Räumliche Webservices

Nach Fu (2011, p.56 ff) können Webservices aus dem Bereich der geographischen Informationen allgemein anhand ihrer Funktionen in verschiedene Kategorien eingeteilt werden. Es gibt **Kartenservices, Datenservices, Analyseservices und Metadaten Katalogservices**.

Bei **Kartenservices** handelt es sich um Funktionen die dem Client ermöglichen Karten in der gewünschten geographischen Ausdehnung anzufragen. Diese werden in einem gewünschten Bildformat, z.B. JPEG oder PNG, zurückgegeben. Es handelt sich hierbei um den am häufigsten vorkommenden räumlichen Webservice. Diese können auch Funktionen wie die Abfrage von Attributen, räumliche Identifizierung und dynamische Projizierungsfunktionen umfassen. Weiter können Kartenservices entweder dynamisch oder gecacht, also zwischengespeichert sein. Dynamische Services werden bei sich häufig ändernden Daten, oder verschiedenen thematischen Layern verwendet. Gecachte Webservices basieren auf vordefinierten Kacheln oder auch Tiles und werden demnach auch tiled map service genannt. Diese Methode kann die Leistung eines Services signifikant erhöhen und wird bei statischen Daten bei denen sich der Inhalt selten ändert. **Datenservices** erlauben es Daten über das Web abzufragen, zu editieren und zu synchronisieren. Auch eine Verbindung mit einem Kartenservice ist möglich, der auch die Darstellung der Daten ermöglicht. Es lassen sich sowohl Vektor- als auch Rasterdaten suchen. Bei **Analyseservices** handelt es sich um die Bereitstellung von GIS Analysefunktionen im Internet, was in der Form eines OGC WPS auch in dieser Arbeit zum Einsatz kommt. Bei einem **Metadaten Katalogservice** können Metadaten, d.h. Daten über GIS Daten, gesucht und bereitgestellt werden.

Die interoperable Bereitstellung dieser Webservices gehört, wie schon beschrieben, zu den Hauptaufgaben der OGC. Demnach sollen nun die wichtigsten Standards und Umsetzungen durch die OGC beschrieben werden.

Zu den webbasierten Geodienste des OGC gehört der **Web Map Service (WMS)**. Dieser beschreibt die Visualisierung und stellt die standardisierte Umsetzung eines Kartenservices dar (OGC, 2006). Der **Web Feature Service (WFS)** bezieht sich auf die Bereitstellung von Vektordaten (OGC, 2005), während sich der **Web Coverage Service (WCS)** sich auf die Bereitstellung von Rasterdaten bezieht (OGC, 2012). Diese beiden Services stellen demnach Umsetzungen eines Datenservices dar. Auch der **Catalogue Service Web (CSW)**, der die standardisierte Suche nach bestimmten Geodaten beschreibt (OGC, 2007a) fällt in diese Kategorie. Der **Web Processing Service (WPS)** geht mit der Spezifizierung der Verarbeitung von Geodaten im Web (OGC, 2007c) einen Schritt weiter als die beschriebenen Services zur Bereitstellung und Visualisierung und soll aufgrund seiner Relevanz für diese Arbeit noch etwas genauer betrachtet werden (siehe Kapitel 2.3.5).

Wichtig ist in diesem Zusammenhang noch die **Geography Markup Language (GML)** zu erwähnen. GML ist eine auf XML basierende Auszeichnungssprache, um geographische Informationen zu speichern und zu übertragen. GML ist Inhalt der ISO Norm 19136 und wurde in Zusammenarbeit mit dem OGC festgelegt (OGC, 2007b). In Zusammenhang mit dieser Arbeit spielt GML eine wichtige Rolle als mögliches Daten- und Austauschformat.

Ein GML Feature kann als Input für einen WPS benutzt werden. Möglich wäre auch die Verwendung eines WFS als Teil eines WPS, wobei Daten von einem WFS häufig als GML zurückgegeben werden.

2.3.5 Web Processing Service (WPS)

Am 08. Juni 2007 wurde die Version 1.0.0 der Web Processing Service (WPS) Spezifikation herausgegeben (OGC, 2007c, p.xi). Die Definition der OGC (2007c, p.4) lautet wie folgt: *„The specified Web Processing Service (WPS) provides client access to pre-programmed calculations and/or computation models that operate on spatially referenced data. The data required by the service can be delivered across a network, or available at the server.“*

Ein WPS stellt also Regeln zur Verfügung, wie ein Client die Ausführung eines Prozesses initiieren kann und wie mit dem Output umzugehen ist. Die benötigten Daten können über ein Netzwerk oder einen Server zur Verfügung gestellt werden (OGC, 2014).

Die Definition von Michael und Ames (2007, p.1) stellt zusätzlich noch den Datenaustausch mittels XML in den Mittelpunkt: *„This proposed specification describes a mechanism by which geoprocessing may be performed on remote servers, using principally extensible markup language (XML) for communication through the Internet. The specification is authored in such a way that it should be fully language and platform independent.“* De Jesus et al (2012, p.76) definiert einen WPS mit dem Satz: *„The WPS standard specifies how atomic geospatial processes can be run in a SOA environment using HTTP-GET, HTTP-POST and SOAP interfaces“*, und stellt damit die atomaren Eigenschaften in den Vordergrund. Auch das OGC WPS Standard Dokument OGC 05-007r7 (OGC, 2014, p.1) geht auf diese Eigenschaft ein. Es wird ausgeführt, dass kein spezifischer Prozess definiert ist der von einem WPS implementiert werden kann, sondern nur der generische Mechanismus. Jede Implementation eines WPS definiert die zur Verfügung stehenden Prozesse, sowie die Möglichen Inputs und Outputs selbst.

2.3.5.1 Ablauf einer WPS Interaktion

Im speziellen definiert der OpenGIS WPS Standard (2007c, p.4 f) drei Operationen, die von einem Client angefragt und von dem WPS Server beantwortet werden können.

a) GetCapabilities

Diese Operation erlaubt dem Client die Metadaten (Capabilities) des Services Anzufragen und zu erhalten. Hier sind die Fähigkeiten des Dienstes sowie die Namen und Beschreibungen der einzelnen Prozesse aufgeführt.

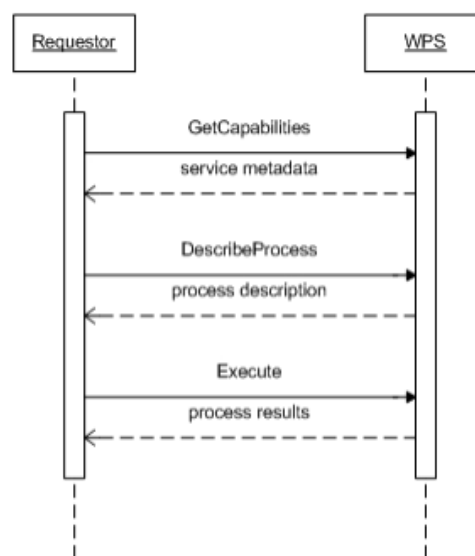
b) DescribeProcess

Diese Operation erlaubt es dem Client spezifische Informationen zu den einzelnen Prozessen des Dienstes anzufragen und zu erhalten. Dazu gehört der benötigte Input und dessen Format sowie der Output der erzeugt werden kann.

c) Execute

Die Execute Operation erlaubt dem Client einen spezifischen Prozess des WPS auszuführen. Dazu werden die bereitgestellten Inputparameter genutzt und schließlich der produzierte Output zurückgegeben.

Abbildung 12: Ablauf einer WPS Interaktion



Quelle (Schaeffer, 2008, p.4)

Ein Prozess könnte z.B. eine Operation zur Verschneidung zweier Polygone sein. Die Antwort auf den GetCapabilities Request beschreibt eine Operation die Intersect heißt. Die Antwort auf den DescribeProcess Request beschreibt, dass als Input für die Intersect Operation zwei Polygone im GML 2.2 Format benötigt werden. Der produzierte Output kann entweder im GML 2.2 oder im GML 3.1 Format sein und als Ressource im Web bereitgestellt werden kann. Der Client führt die Execute Operation aus, implementiert den Input direkt in die Anfrage und legt fest, dass der Output im Web bereitgestellt werden soll. Die Antwort auf den Execute Request ist ein XML Dokument, das den Input und den Output identifiziert und sagt ob der Prozess erfolgreich abgelaufen ist. War er erfolgreich so enthält er in diesem Fall auch eine Referenz zu dem erstellten Output im Web (OGC, 2007c, p.5).

Es sind drei verschiedene Input und Output Typen für einen WPS definiert: Literal Data, Complex Data und Bounding Box Data. **Literal Data** können alle Datentypen wie String, Float, Date, etc. sein, die als Primitive Datentypen im W3C XML Schema (2004b) definiert

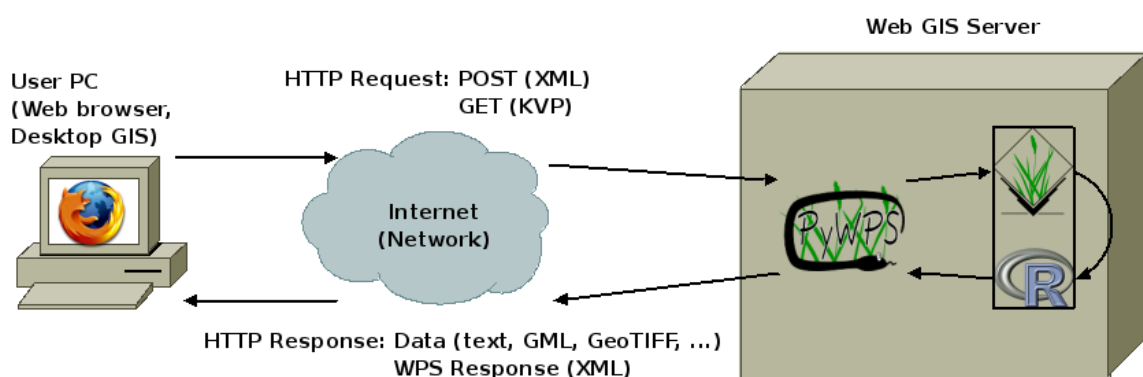
sind. **Complex Data** ist der Datentyp mit dem komplexe Vektor- oder Rasterdaten dargestellt werden. Diese können auf zwei Arten in den WPS Request implementiert werden. Zum einen können die Daten Teil der Anfrage zum Server, in Form von z.B. GML für Vektor- und z.B. als base64 Kodierung für Rasterdaten sein. Zum anderen kann eine Referenz zu den Daten, also eine URL die auf die Daten verweist, Teil der Anfrage sein. Schließlich sind **Bounding Box Data** noch als In- bzw. Output für einen WPS definiert. Dieser beinhaltet die Koordinaten für ein Begrenzungsrechteck, wobei auch das jeweilige Referenzsystem angegeben werden muss (OGC, 2007c).

Bis zur Etablierung des WPS kam es in vielen Fällen vor, dass der Code oder das bestimmte Programm z.B. aufgrund von Fortschritten in der Programmierung nur eine kurze Lebensdauer hatte (Gehlot and Verbree, 2006, p.1). Auch Schaeffer (2008, p.2) greift Gehlot und Verbree (2006) auf und beschreibt die leidende Interoperabilität.

2.3.5.2 Open Source Implementierung

Bei **PyWPS** handelt es sich um den Rahmen für eine Implementierung der WPS 1.0.0 Spezifikation der OGC. PyWPS basiert auf der Programmiersprache Python und übersetzt die Anfragen zwischen dem Client und den auf dem Server installierten Werkzeugen. Hierbei sind vor allem Werkzeuge aus dem Open Source GI System GRASS implementiert, d.h. dass PyWPS die Anfragen nicht selber verarbeitet sondern nur übersetzt (Cepicky, 2009). Im Zusammenhang mit dieser Arbeit spielt PyWPS vor allem als Vorbild der Umsetzung einer erfolgreichen WPS Orchestrierung durch de Jesus et al. (2012) eine Rolle.

Abbildung 13: Funktion von PyWPS innerhalb einer Client-Server Architektur



Quelle http://pywps.wald.intevation.org/documentation/how_it_works.html

Eine weitere Implementierung der OGC WPS Spezifikation ist der **52°North WPS**. Im Gegensatz zu PyWPS basiert diese komplett auf der Programmiersprache Java. Die standardisierte Bereitstellung von Geoverarbeitungswerkzeugen kann bei dem 52°North WPS nicht

nur auf GRASS Werkzeugen basieren, sondern unter anderen auch auf denen von ArcGIS (52°North Initiative for Geospatial Open Source Software GmbH, 2014).

2.4 Geoverarbeitungsagenda

Wie in der Einleitung bereits beschrieben, folgt diese Arbeit der von Brauner et al. (2009) vorgeschlagenen Geoverarbeitungsagenda. In diesem Zusammenhang sollen die drei Komponenten der Agenda Semantik, Orchestrierung und Performance in diesem Kapitel, vor allem bezüglich bereits geleisteter Forschungsarbeit und Definition noch einmal genauer betrachtet werden.

2.4.1 Semantik

Das Internet ist heutzutage hauptsächlich ein Kommunikations- und Informationsmittel für Menschen untereinander, jedoch kommunizieren auch Computer über das Internet. Deren Interaktion ist auf vorprogrammierten Vorgaben basiert und es ist Maschinen nicht möglich, selbstständig auf Informationssuche zu einer Fragestellung zu gehen, da die Inhalte für Maschinen von der Bedeutung her unverständlich sind. Der Erfinder des World Wide Web, Tim Berners Lee (1999) beschreibt schon 1999 seine Vision des semantischen Webs, das auch als Web 3.0 bezeichnet wird, indem diese Maschine-Maschine Kommunikation möglich ist. Angewandt auf das Thema dieser Arbeit sind sogenannte semantische Webservices interessant. Hier treten semantische Probleme bei der automatischen Orchestrierung auf, was Bernard et al. (2005) und Kiehle (2007) innerhalb der GI Science als die nächste große Grenze die es zu überschreiten gilt beschreiben.

In der Geoverarbeitungsagenda von Brauner et al. (2009) sind semantische Beschreibungen demnach auch einer der drei Hauptbestandteile. Die Autoren beschreiben den WPS als ein erstes Bestreben automatisches Auffinden, Verketteten und Ausführen von Geoverarbeitungsservices. Kiehle et al. (2007) sehen bei der automatischen Verkettung, komplett ohne menschliche Interaktion, noch eine große Lücke, die auf fehlenden semantischen Beschreibungen basiert.

In der OpenGIS Abstract Specification – Topic 12 (OGC, 2002) sind, wie bereits beschrieben, konzeptionelle Fundamente für die Entwicklung der OGC-Standards erläutert, welche auch in die ISO Spezifikation „ISO 19119 – Geographic Information Services“ (ISO, 2005) übernommen wurden. Auf allgemeine Web Technologien bezogen folgt diese Spezifikation dem Konzept einer SOA. Im Zusammenhang mit Interoperabilität werden zunächst verschiedene Ebenen beschrieben auf denen Interoperabilität erfolgen muss. Auf der Ebene der sogenannten Primitiven wird sowohl semantisch als auch physisch festgelegt was eine Zahl, ein Zeichen, eine Zeichenkette, eine Geometrie, ein Punkt, eine Linie etc. ist. Auf der Ebene der Features werden diese Primitiven für Attribute verwendet und zu Features gruppiert. Entscheidend ist an dieser Stelle der Feature-Typ. Hier muss das Anwendungsgebiet,

oder auch die Domain, bekannt sein um diesen zuordnen zu können. Die Semantik der Domain wird in den Metadaten beschrieben. Um die Interoperabilität zu gewährleisten muss also eine Instanz Auskunft über Objekte, deren Bedeutung und die erlaubten Operationen geben. Bis dato muss Interoperabilität in diesem Zusammenhang manuell gewährleistet werden.

2.4.2 Orchestrierung

Das zweite Hauptforschungsgebiet der Geoverarbeitungsagenda (Brauner et al., 2009) ist die Orchestrierung. Grundsätzlich kann die Zusammenstellung von Aktivitäten zu Arbeitsabläufen als Prozess oder Workflow bezeichnet werden (Melzer, 2010, p.240). Bei der Verkettung von Webservices muss zunächst zwischen **Orchestrierung** und **Choreographie** unterschieden werden. Bei der Orchestrierung handelt es sich um eine Ablaufbeschreibung und Definition der Nachrichtenflüsse auf Basis zentraler Steuerung. Bei einer Choreographie liegt keine zentral steuernde Instanz vor, die Steuerung läuft durch direkte Kommunikation der Webservices untereinander ab. Eine solche Zusammenstellung im Allgemeinen wird auch als Geschäftsprozess bezeichnet (Dengel, 2012, p.289). In dieser Arbeit findet die Orchestrierung von Webservices Anwendung, da die Webservices entweder durch einen Workflow oder den Benutzer gesteuert werden und somit eine steuernde Instanz vorliegt.

In der ISO Spezifikation „ISO 19119 – Geographic Information Services“ (ISO, 2005) sind drei verschiedene Arten der Verkettung von Webservices definiert:

- (1) Transparente oder vom Benutzer definierte Verkettung
- (2) Semitransparente oder Workflow gelenkte Verkettung
- (3) Opake oder zusammengefasste Verkettung

Bei der transparenten Verkettung von Webservices wird jeder Prozess vom Benutzer nacheinander aufgerufen. Bei der semitransparenten Verkettung sind die einzelnen Prozesse zu einem Workflow zusammengefasst, wobei der Benutzer weiß welche Prozesse dies sind. Bei der opaken, d.h. undurchsichtigen, Verkettung weiß der Benutzer nicht welche einzelnen Prozesse durch einen Workflow angesprochen werden (ISO, 2005).

Auch der OGC WPS kann in seiner Funktion als Webservice im Allgemeinen unter bestimmten Umständen orchestriert werden. In der WPS Spezifikation (OGC, 2007c, p.7) sind dazu drei Möglichkeiten beschrieben. Als Erstes wird die Benutzung von BPEL als steuernde Instanz genannt. BPEL (Business Process Execution Language) ist die Kurzform von WS-BPEL (Web Service Business Process Execution Language) und ist ein Standard der

Organisation OASIS (Organization for the Advancement of Structured Information Standards). Es handelt sich dabei um eine XML-basierte Beschreibungssprache für die Modellierung von Geschäftsprozessen und somit zur Ausführung von Workflows (OASIS, 2007). Diese Methode ist vor allem für komplexe Workflows geeignet. Als zweites wird der Nutzung eines WPS, der eine Sequenz von Webservices inklusive anderer WPS aufruft und als steuernde Instanz fungiert, genannt. Als drittes wird die Möglichkeit einer einfachen Verkettung in einem HTTP Get Request beschrieben (OGC, 2007c, p.7).

Brauner et al. (2009, p.4) belegen die Relevanz der Orchestrierung von Webservices für räumliche Anwendungen durch deren Komplexität. Viele Analysen und Verarbeitungen erfordern mehrere Arbeitsschritte. Dementsprechend war die Orchestrierung von räumlichen Webservices Teil vieler Forschungsarbeiten.

Eberle und Strobl (2012) haben in ihrer Arbeit eine Orchestrierung für das generieren und zur Verfügung stellen von Fernerkundungsprodukten umgesetzt. Dabei beschreiben sie mögliche Umsetzungen der Orchestrierung durch 52° North WPS, Apache ODE mit BPEL, und die Umsetzung mit einer internen Software des Deutschen Zentrums für Luft- und Raumfahrt (DLR). Im speziellen Fall ist die Umsetzung mit dem 52° North WPS, aufgrund der Verarbeitung großer Datenmengen in der Fernerkundung, nicht geeignet. Im Detail führen große Rasterdaten zu Speicherproblemen bei der Übersetzung von großem XML Code. Eine Umsetzung mit der Apache Software beschreiben die Autoren als zu komplex, sodass die Umsetzung letztendlich mit der internen Software erfolgt.

Auch Stollberg und Zipf (2007, p.242) und (2009, p.2) verweisen auf die Nutzung von BPEL und dessen Verbreitung als Standard für WSO. In Bezug auf die Orchestrierung von WPS verweisen sie weiter darauf, dass der WPS nur den Service selber, jedoch nicht die darunter liegenden Prozesse definiert. Diese werden durch die verschiedenen Implementierungen zur Verfügung gestellt. Dies bedeutet letztendlich auch, dass es keine vordefinierten WSDL Dateien für einen WPS geben kann. Letztendlich muss deswegen der Entwickler eine WSDL für jeden Prozess zusammen mit dem DescribeProcess Dokument entwickeln. Ein vorhandenes WSDL Dokument für jeden einzelnen Prozess ermöglicht es dann, diesen z.B. mit BPEL zu orchestrieren. Stollberg und Zipf (2009) sowie Lanig und Zipf (2010) haben in Ihrer Arbeit (2009) deswegen ein XML Schema entwickelt, das eine Dienstkette definiert um OWS zu erleichtern.

2.4.3 Performance

„Besides the current research, still some challenges remain open. Performance is an important issue in geoprocessing orchestration“ (Zhao, Foerster and Yue, 2012, p.9).

Friis-Christensen et al. (2007), Kiehle (2007) und Barinski (2008) weisen der Performance eine Schlüsselrolle im Zusammenhang mit Webservice Orchestrierung zu. Rechenleistung und Bandbreite sind den Autoren nach die limitierenden Faktoren. Friis-Christensen et al. (2007) beschreiben in ihrer Arbeit mögliche Ansätze zur Performance Verbesserung. Die Ansätze haben das Ziel der Limitierung des Datentransfers gemein. Dies kann über asynchrone Client-Server Kommunikation, eine Reduzierung der Rücksendung von Daten zur selben Verarbeitungsinstanz und über die Verarbeitung der Daten an ihrer Quelle geschehen. Letzteres geschieht über die Übertragung des Verarbeitungscode zum Client anstatt der Übertragung der Daten zum Server und wird auch als „Moving Code Paradigma“ bezeichnet. Während Brauner et al. (2009, p.5) noch keine Ansätze für eine derartige Umsetzung gefunden haben wurde dies von Müller et al. (2013) untersucht. Die interessanten Ansätze kommen zu dem Schluss, dass in diesem Bereich einheitliche Standards fehlen (Müller, Bernard and Kadner, 2013, p.199)

Barinski (2008) beschreibt in seiner Arbeit die Vorteile von Grid Computing, also verteilter Rechenleistung, im Zusammenhang mit der Performance. Die Autoren vergleichen die von der ISO (2005) definierten Verkettungsmethoden der transparenten Verkettung, der semitransparenten Verkettung und der opaken Verkettung. Sie schließen, dass die semitransparente Verkettung am besten mit BPEL funktioniert und so in manchen Fällen die Performance verbessert werden kann.

Eine praktische Untersuchung der Performance verschiedener WPS Implementierungen wurde von Poorazizi und Hunter (2014) durchgeführt. Untersucht wurden die fünf WPS Server 52°North, Deegree, GeoServer, PyWPS und Zoo. Es wurde ein WPS entwickelt der wiederholt mit verschiedenen Inputs an die Server gesendet wurde. Die Autoren kommen in ihrer Arbeit zu dem Ergebnis, dass der Degree Server die Anfragen durchschnittlich am schnellsten bearbeitet, jedoch keine signifikanten Unterschiede in der Antwortzeit vorhanden sind.

2.5 Workflow Management Systeme

Bei einem Workflow Management System handelt es sich um Software, die die Infrastruktur zum Aufbau, Ausführen und Überwachen wissenschaftlicher Arbeitsabläufe zur Verfügung stellt. Ein wichtiger Teil eines Workflow Management Systems ist das Überwachen und Koordinieren der einzelnen Komponenten eines Workflows während der Ausführung, womit die Komponenten einer Orchestrierung erfüllt sind. Mit der Entwicklung einer immer datenintensiver werdenden Wissenschaft im Allgemeinen und der wachsenden Abhängigkeiten von Computern, ist der Bedarf an eben solcher Software gewachsen. Dabei haben sich viele Werkzeuge entwickelt, die nur wenig umfangreiche und gut definierte Aufgaben ausführen können. Dies wiederum führt zu der Notwendigkeit von Software, die viele solcher Anwendungen in einem Workflow kombinieren und ausführen kann. Zu weiteren Schwierigkeiten kommt es in diesem Zusammenhang auch durch das Vorhandensein verschiedener und inkompatibler Datenformate, die die unterschiedlichen Services produzieren. An dieser Stelle löst Wissen aus der Informatik Probleme bei denen die Expertise von Wissenschaftlern spezifischer Domänen nicht ausreicht. Die technischen Details der Ausführung können nun von dem entwickelten Workflow Management System übernommen werden, die einfach zu nutzende Möglichkeiten bieten um verschiedene wissenschaftliche Experimente auszuführen (Taverna, 2010).

Ein Beispiel für ein solches Workflow Management System ist Taverna. Dieses Open Source Programm wurde ursprünglich für den Bereich der Bioinformatik entwickelt. Es ist jedoch generisch genug um Webservices anderer wissenschaftlicher Disziplinen einzubinden und auszuführen und ist so auch allgemein als Workflow Management System zu bezeichnen (Wolstencroft et al., 2013).

De Jesus et al. (2012) haben in ihrer Arbeit anhand der Orchestrierung eines PyWPS gezeigt, dass sich Taverna auch in diesem Zusammenhang eignet. Ihre Arbeit zeigt, im Gegensatz zu den vielen anderen Arbeiten aus dem Bereich der Orchestrierung von Webservices, durch die Nutzung des Workflow Management Systems Taverna einen benutzerfreundlichen Ansatz einer Orchestrierung. Durch die Nutzung von Taverna ist es möglich Webservices zu orchestrieren ohne eine Programmiersprache einzusetzen bzw. sogar zu erlernen oder die komplizierte Definition von Austauschformaten.

3 Offline Geoverarbeitung

In diesem Kapitel soll beschrieben werden worum es sich bei dem umzusetzenden Workflow handelt und wie dieser Offline mit ArcGIS erstellt wird. Zunächst werden einige allgemeine Dinge zur offline Geoverarbeitung mit ArcGIS beschrieben. Anschließend sollen die Motivation und das Anwendungsgebiet des umzusetzenden Workflows genauer erläutert werden. Schließlich wird die Funktionsweise des Workflows selber erläutert, sowie die Aufteilung des Workflows in einzelne Module für die spätere Orchestrierung.

3.1 Geoverarbeitung mit ArcGIS

Grundsätzlich Geoverarbeitung ist ein wesentlicher Teil der täglichen Arbeit mit GIS. Geoverarbeitung in ArcGIS besteht aus einer Vielzahl von Werkzeugen zum Ausführen der unterschiedlichsten Operationen, die von einfachen Buffern und Polygon-Overlays bis zu komplexen Regressionsanalyse und Bildklassifizierung reichen können. Teil der Geoverarbeitung in ArcGIS ist auch der ModelBuilder, dessen Ziel es ist mehr oder weniger komplexe Verarbeitungsschritte zu automatisieren (ESRI, 2014e). Neben der Erstellung eines Modells im ModelBuilder von ArcGIS und der Verarbeitung der Geodaten wird die Software auch zur Visualisierung der Ergebnisse und Verwaltung der Daten genutzt. Von den in Kapitel 2.1 beschriebenen vier funktionalen Aspekten aus dem Softwarebereich geographischer Informationssysteme Erfassung, Verwaltung, Analyse und Präsentation (Bartelme, 2005, p.64 ff) werden im Folgenden demnach drei Bereiche genutzt. Lediglich die Erfassung von Daten ist kein Teil dieser Arbeit. Im Folgenden Anwendungsbeispiel aus der Standortanalyse werden Daten der Open Street Map (OSM, 2015) verwendet.

3.2 Workflow und Anwendungsgebiet

Das im Rahmen dieser Arbeit entstandene Modell trägt den Namen ‚Population Distance‘ Modell. Das Ziel dieses Geoverarbeitungsmodells ist eine Entfernungsanalyse umliegender Punkt Features um einen bestimmten Punkt. Dabei wird eine Auswahl von Features in einer festzulegenden Distanz um den Punkt herum extrahiert. Anschließend werden in einer Distanzanalyse die Entfernungen der extrahierten Features zu dem Punkt ermittelt und in ein neues Feld der Attributtabelle geschrieben. Zusätzlich werden Statistiken zu der ermittelten Bevölkerungszahl in die Attributtabelle einer erstellten Buffergeometrie, die die Größe der vorher gewählten Distanz hat, geschrieben.

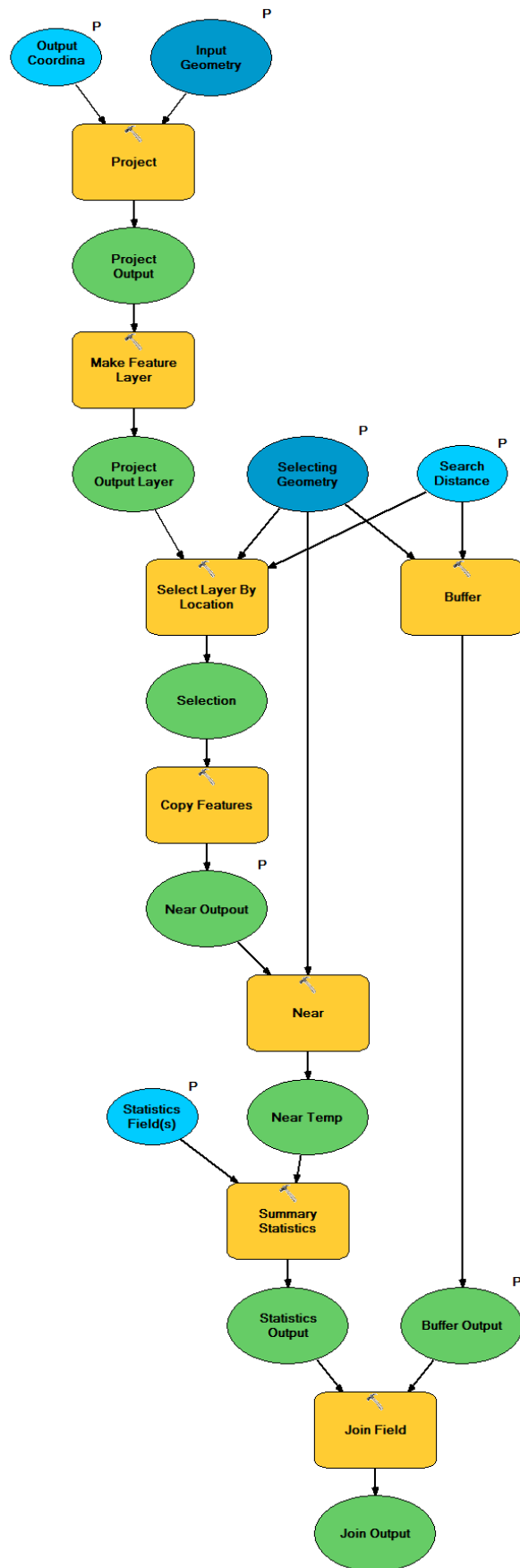
Anwendung kann dieses Modell in der Standortanalyse finden. Ein mögliches Szenario wäre beispielsweise die Planung einer neuen Fabrik oder eines Einkaufszentrums. Hier kann die spezifische Entfernung verschiedener Städte und deren Einwohner zu dem möglichen Standort untersucht werden. So kann festgestellt werden ob zum einen genügend potentielle Arbeitskräfte in einer bestimmten Entfernung zu einer Fabrik wohnen, oder zum anderen ob genügend potentielle Kunden in der Umgebung des möglichen Einkaufszentrums wohnen.

Abbildung 14: ‚Population Distance‘ Modell zeigt die Struktur des erstellten Modells im ArcGIS ModelBuilder. Die einzelnen Verarbeitungsschritte und Module werden im folgenden Kapitel genauer erläutert. Insgesamt werden acht Werkzeuge genutzt, welche mit einem gelben, abgerundeten Rechteck dargestellt sind. Als Input für dieses Modell werden zwei verschiedene Geometrien benötigt, zum einen die ‚Input Geometry‘ und zum anderen die ‚Selecting Geometry‘. Die Inputs des Modells sind in einer dunkelblauen Ellipse dargestellt, während der Output eines Verarbeitungsschritts in einer grünen Ellipse dargestellt ist. Bei den Eingangsgeometrien ist ein hochgestelltes P zugeordnet. Ein hochgestelltes P markiert einen Parameter für das Modell, d.h., dass dieser vor der Ausführung des Modells vom Nutzer bestimmt werden muss. Das Modell erstellt letztendlich zwei Outputs, welche in diesem Fall als Parameter markiert sind und der Speicherort festgelegt werden muss. Bei allen anderen Outputs des Modells handelt es sich um temporäre Daten die nach Fertigstellung des Modells gelöscht werden. Schließlich sind neben den Eingangsgeometrien bei manchen Geoverarbeitungswerkzeugen noch andere Parameter anzugeben. Diese sind in einer hellblauen, kleineren Ellipse dargestellt.

Die Datengrundlage dieser Analyse bilden Daten der Open Street Map (OSM, 2015). Im Speziellen wird zum einen die OSM Places Punkt Feature Class des Bundeslandes Hessen genutzt und zum anderen ein einzelnes Punkt Feature der Kleinstadt Grünberg in Hessen.

Zur Vorbereitung der Daten wird die Datei auf doppelte Einträge untersucht und diese entfernt. Weiter werden auch Siedlungen ohne Einwohner entfernt. Diese haben auf der einen Seite keine Auswirkungen auf das Ergebnis und auf der anderen Seite wird das Ergebnis, vor allem in Hinblick auf eine Bereitstellung auf dem ArcGIS Server, kompakt gehalten.

Abbildung 14: „Population Distance“ Modell

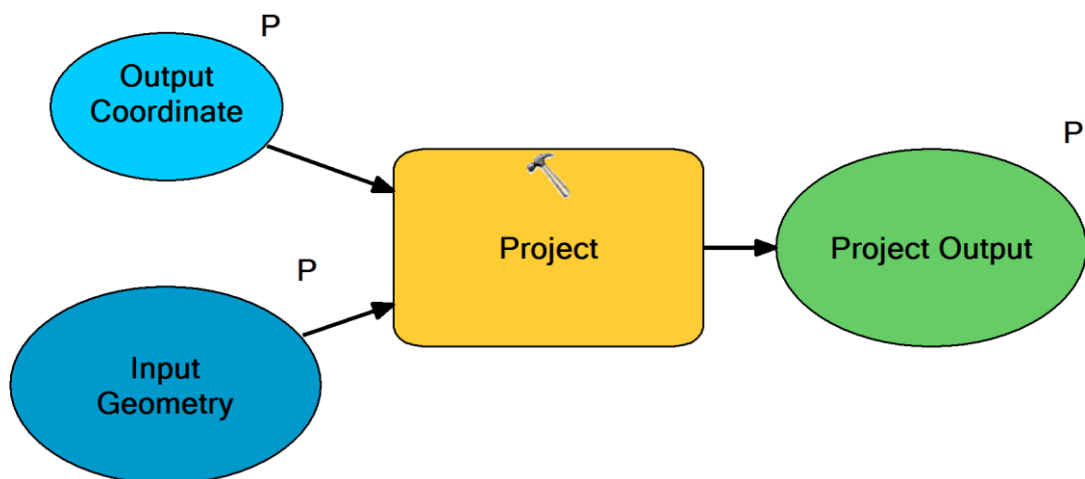


3.3 Module

Das Ziel dieser Arbeit ist, wie beschrieben, die Orchestrierung eines Webservices um dessen Performance zu messen. Als Anwendungsfall soll das erstellte Modell dienen, welches dafür in einzelne Module aufgeteilt wurde um es anschließend wieder zusammenführen zu können. Im Folgenden werden die benutzten Werkzeuge des ‚Population Distance‘ Modells anhand der erstellten Module erläutert. Ziel der Modularisierung dieses Modells ist es auch die Wiederverwendbarkeit der einzelnen Webservices zu erhöhen. Während das Modell als Ganzes nur im engen Rahmen des beschriebenen Anwendungsfalls zum Einsatz kommen kann, können die einzelnen Module beliebig mit anderen Webservices aus diesem Bereich kombiniert werden.

1) Project

Abbildung 15: Modul 1: Project



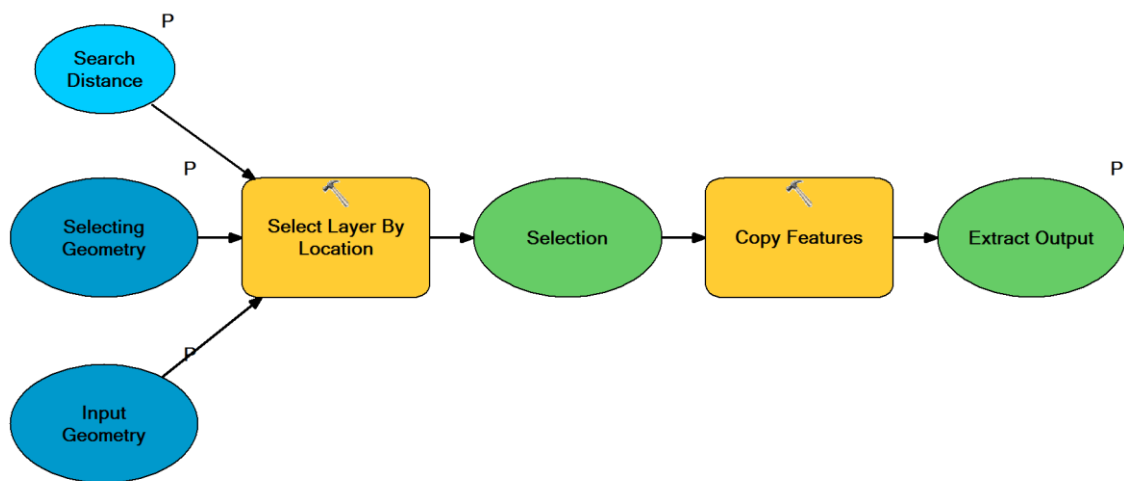
Quelle Eigenes Modell mittels ArcGIS 10.2 ModelBuilder

Aufgrund der Bestimmung des Modells Features in einer bestimmten Entfernung zu einem anderen Feature zu bestimmen, ist es wichtig korrekte Distanzberechnungen durchzuführen. Dafür ist es notwendig den Datensatz in eine geeignete, soweit wie möglich Längentreue Projektion zu überführen. Grundsätzlich besitzen die OSM Daten WGS 84 als geodätisches Referenzsystem. Je nach Anwendungsgebiet kann der Datensatz z.B. in die passende UTM Zone der jeweiligen Region projiziert werden um die Längentreue zu gewährleisten. Als Voreinstellung ist für dieses Modell demnach WGS 84, UTM Zone 32 für Deutschland ausgewählt.

Im Speziellen wird im ersten Modul dieses Geoverarbeitungsmodells der Parameter „Input Geometry“, also das OSM Städte Feature, mit dem Tool ‚Project‘ (ESRI, 2014c) projiziert. Die Projektion ist weiter auch für das Near Tool von entscheidender Bedeutung. Unwichtig ist hingegen die Projektion des „Selecting Geometry“ Features, also in diesem Fall die Punktgeometrie der Stadt Grünberg. Ob dieser projiziert ist oder nicht, ändert nichts an der Auswahl. Der Output dieses Moduls ist wiederum der Input für das Modul ‚Extract by Location‘.

2) Extract by Location

Abbildung 16: Modul 2: Extract by Location

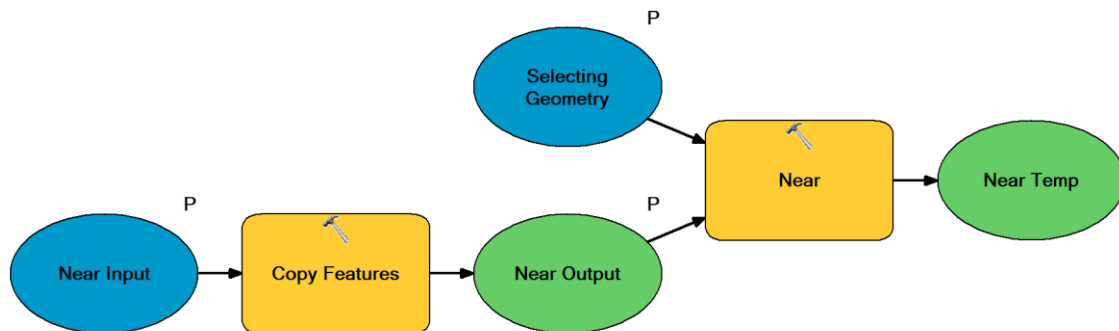


Quelle Eigenes Modell mittels ArcGIS 10.2 ModelBuilder

Das Modul ‚Extract by Location‘ existiert in dieser Form nicht innerhalb der ArcGIS Werkzeug Palette. Es ist zusammengesetzt aus den Werkzeugen ‚Select Layer by Location‘ (ESRI, 2014d) und Copy (ESRI, 2014a). Die ‚Input Geometry‘ ist die projizierte Feature Class der Städte in Hessen und ‚Selecting Geometry‘ ist die Stadt Grünberg. Dazu kommt noch der Parameter der ‚Search Distance‘, mit dem festgelegt wird in welchem Umkreis um die ‚Selecting Geometry‘ die projizierten Städte der ‚Input Geometry‘ ausgewählt werden. Damit die Auswahl permanent wird, muss diese mit dem ‚Copy‘ Werkzeug in eine eigene Feature Class kopiert werden. Der Output dieses Moduls stellt nun den Input für das Modul ‚Near‘ dar.

3) Near

Abbildung 17: Modul 3: Near



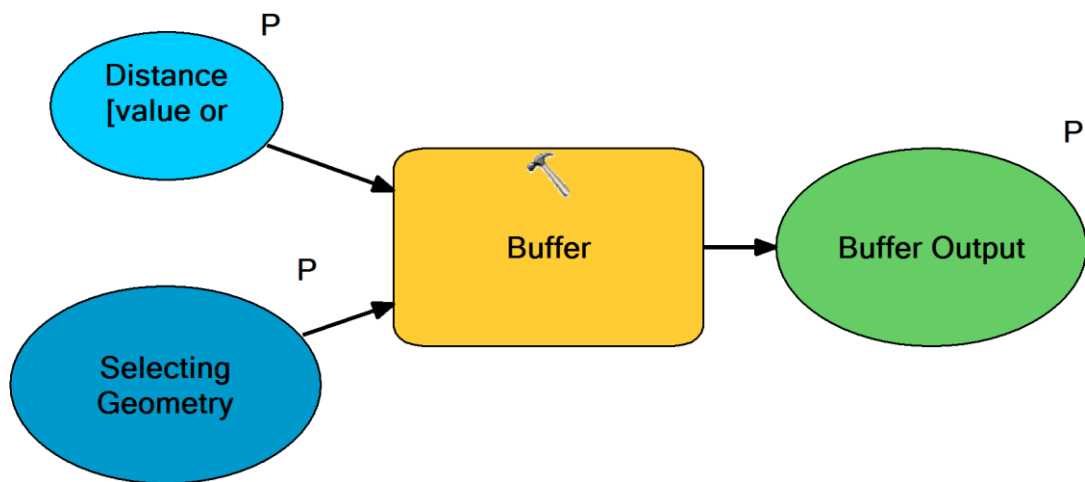
Quelle Eigenes Modell mittels ArcGIS 10.2 ModelBuilder

In diesem Modul wird mittels des ArcGIS Werkzeugs ‚Near‘ (ESRI, 2014b) nun die eigentliche Distanzberechnung der umliegenden Orte zu der ‚Selecting Geometry‘ durchgeführt. Wie schon zu Beginn erwähnt, ist die Projektion auch für diesen Verarbeitungsschritt von entscheidender Bedeutung. Die zu berechnenden Distanzen werden in der Einheit des Koordinatensystems des Eingangsdatensatzes berechnet. Bei einem unprojizierten Datensatz im geodätischen Referenzsystem WGS 84 wäre die Einheit in der die Distanz angegeben wird z.B. Dezimalgrad. Wenn das Modul als eigenständiger Webservice genutzt wird muss der Datensatz also projiziert sein. Innerhalb des ‚Population Distance‘ Modells und der Orchestrierung der einzelnen Webservices wird die Projektion des Datensatzes, wie im ersten Schritt beschrieben, von dem ‚Project‘ Modul vorgenommen.

Das ‚Near‘ Werkzeug erstellt bei der Berechnung keine neue Feature Class sondern schreibt die Distanzen in ein neues Feld der Attributtabelle der alten Feature Class. Deswegen sind in Abbildung 17 der eigentliche Input als ‚Near Output‘, und der eigentliche Output als ‚Near Temp‘ bezeichnet. Des Weiteren macht es die Modularisierung notwendig, dass dieses Modul, wie das Modul ‚Extract by Location‘, ebenfalls das ‚Copy‘ Werkzeug beinhaltet. Falls das Modul als eigenständiger Webservice genutzt wird ist so sichergestellt, dass die Ausgangsdaten nicht verändert werden und eine neue Feature Class erstellt wird. Der Nachteil ist, dass bei der Orchestrierung der einzelnen Module, das ‚Copy‘ Werkzeug durch dieses und das ‚Extract by Location‘ Modul doppelt ausgeführt werden.

4) Buffer

Abbildung 18: Modul 4: Buffer

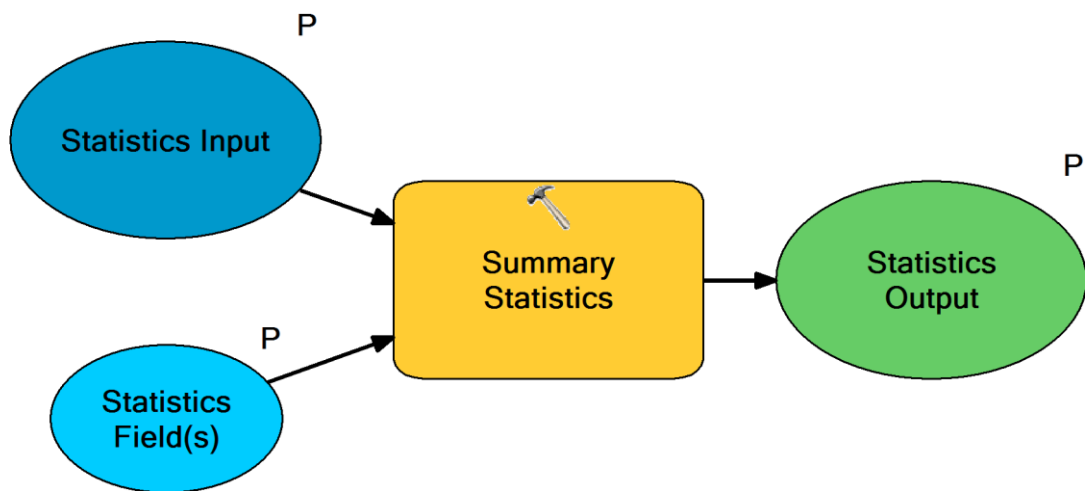


Quelle Eigenes Modell mittels ArcGIS 10.2 ModelBuilder

Mit dem Werkzeug ‚Buffer‘ wird in diesem Modul eine einfache Buffergeometrie um die ‚Selecting Geometry‘ erstellt. Ein zweiter wichtiger Parameter für dieses Modul ist die Distanz in der der Buffer erstellt werden soll. Die Voreinstellung beträgt 50 Kilometer. Im Zusammenhang mit dem gesamten Modell hätte der Buffer auch als auswählende Instanz innerhalb des ‚Extract by Location‘ Moduls dienen können. Im Hinblick auf die Modularisierung und der Wiederverwendbarkeit als eigenständiger Webservice wurde dieses Modul jedoch als einzelnes erstellt.

5) Statistics

Abbildung 19: Modul 5: Statistics

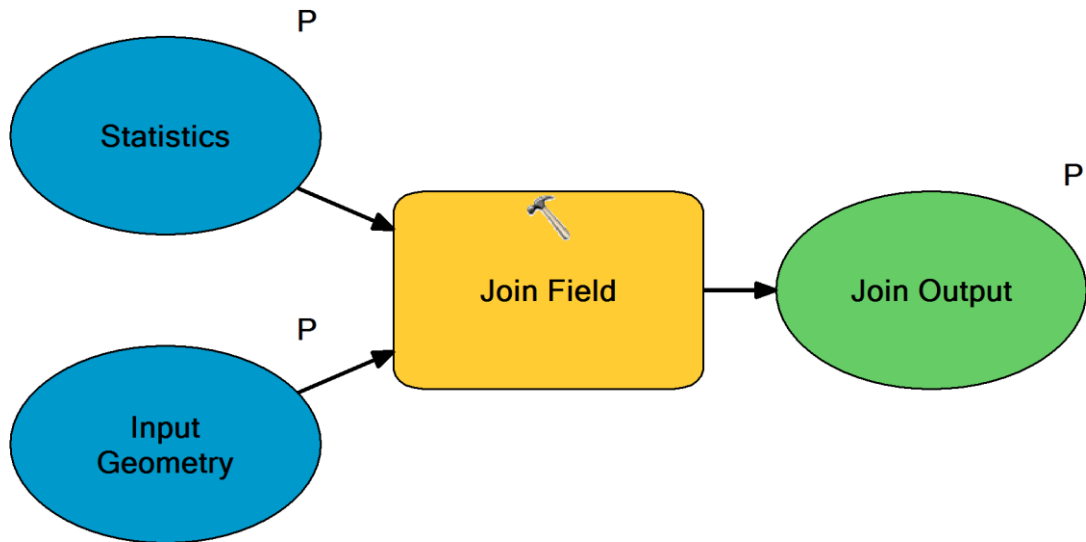


Quelle Eigenes Modell mittels ArcGIS 10.2 ModelBuilder

Im Modul ‚Statistics‘ werden mittels des ‚Summary Statistics‘ Werkzeugs verschiedene Statistiken des Eingangsdatensatzes berechnet. Innerhalb des ‚Population Distance‘ Modells und der späteren Orchestrierung werden diese als Attribute für die erstellte Buffergeometrie verwendet. Als eigenständiger Webservice ist eine Tabelle der Output dieses Moduls. Die Statistiken sind nur offline als Parameter einstellbar. In einem bereitgestellten Modell sind diese nicht änderbar weswegen die Voreinstellung übernommen wird. Im speziellen werden für diesen Anwendungsfall die Summe, der Mittelwert, das Minimum und das Maximum der Einwohnerzahlen der jeweiligen Städte berechnet.

6) Join

Abbildung 20: Modul 6: Join



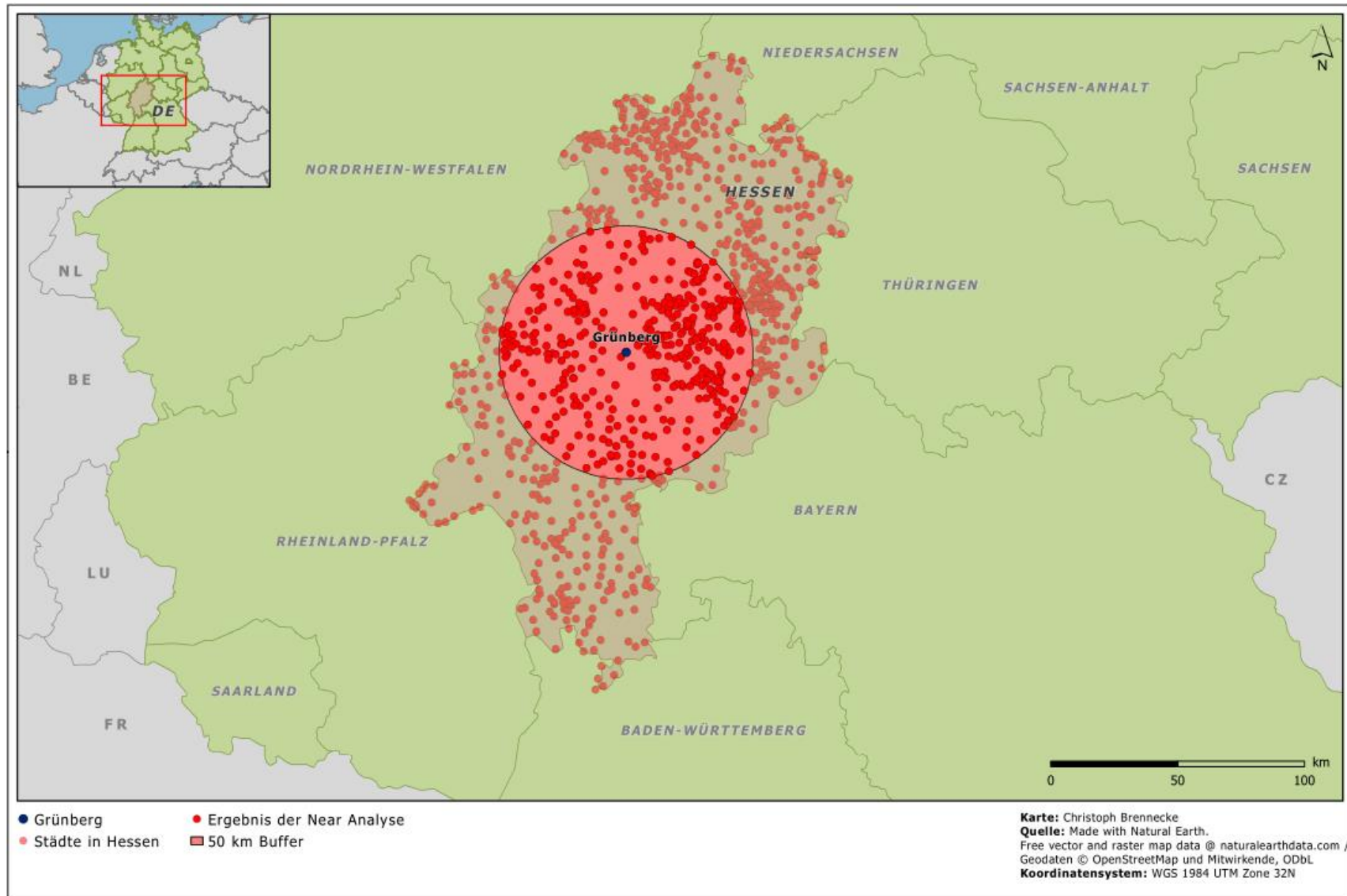
Quelle Eigenes Modell mittels ArcGIS 10.2 ModelBuilder

Mit dem Modul ‚Join‘ werden Attribute einer Eingangstabelle mit der Attributtabelle einer Eingangsgeometrie verbunden. Da dies auf Basis des Object ID Feldes geschieht, müssen beide Datensätze die entsprechende Form haben. Als eigenständiger Webservice besitzt dieses Modul demnach nicht den vollen Funktionsumfang des zugrundeliegenden ‚Join Field‘ Werkzeugs. Im Anwendungsfall sorgt dieses Modul jedoch dafür, dass die errechneten Statistiken mit der Attributtabelle der erstellten Buffergeometrie verbunden werden. Der Output des gesamten Modells und der Orchestrierung umfasst somit nur zwei Geometrien und nicht noch eine extra Tabelle.

3.4 Ergebnis

Abbildung 21 zeigt die Darstellung des Ergebnisses der Analyse nach einfacher Ausführung des Modells in ArcMap. Zu sehen ist das Bundesland Hessen und die benachbarten Bundesländer der Bundesrepublik Deutschland. Innerhalb Hessens sind alle großen und kleinen Städte sowie die Dörfer des Bundeslandes dargestellt die Teil des OSM_Places Shapefiles sind und wie beschrieben mehr als null Einwohner haben. Diese Städte sind in einem weniger gesättigten Rot dargestellt. Das eigentliche Ergebnis der Analyse ist zum einen die Buffergeometrie und zum anderen die darin enthaltenen Städte, die in einem gesättigten Rot dargestellt sind. Den Ausgangspunkt für den Buffer bildet das Punkt Feature der Stadt Grünberg, das in Blau dargestellt ist.

Abbildung 21: Ergebnis des ‚Population Distance‘ Modells



Quelle Eigene Darstellung mittels ArcGIS 10.2 / Geodaten: (OSM, 2015; Natural Earth, 2015)

Abbildung 22 zeigt die ersten 30 Features in der Attributtabelle der erstellten Near Feature Class als Ergebnis der Distanzanalyse. Die komplette Attributtabelle der Near Feature Class befindet sich im Anhang. Zu sehen sind nur die für die Aussage wichtigen Attribute der Feature Class, die anderen wurden ausgeblendet. Das Attribut ‚Near_DIST‘ zeigt die Entfernung der jeweiligen Stadt zu der Ausgangsgeometrie in Metern. Die einzelnen Features sind auch nach diesem Feld aufsteigend geordnet. Die weiteren Attribute sind der Name der Stadt und dessen Typ, der je nach Größe die Attribute ‚city‘, ‚town‘ und ‚village‘ zugeordnet sind. Das Attribut ‚population‘ enthält die Einwohnerzahl der jeweiligen Stadt, die im weiteren Verlauf der Analyse ausgewertet wird.

Abbildung 22: Attributtabelle des erstellten Near Features

Near					
	Shape *	name	type	population	NEAR_DIST
▶	Point	Grünberg	town	13828	0,000044
	Point	Queckborn	suburb	1418	2682,798963
	Point	Mücke	town	10187	5492,201773
	Point	Laubach	town	10330	5833,134443
	Point	Merlau	village	1184	6035,694804
	Point	Groß-Eichen	village	954	8222,851207
	Point	Wettsaasen	village	212	8481,429792
	Point	Reiskirchen	town	10784	9375,989328
	Point	Ruppertenrod	village	856	9568,65965
	Point	Sellnrod	village	809	9621,724999
	Point	Rabenau	town	5396	9840,709063
	Point	Höckersdorf	village	250	10385,739065
	Point	Ober-Ohmen	village	790	11343,013583
	Point	Bobenhausen II	village	436	12440,635033
	Point	Gemünden	town	3216	12622,592245
	Point	Lich	town	13557	12719,511088
	Point	Hainbach	village	300	13115,540715
	Point	Fernwald	town	6680	13244,046886
	Point	Buseck	town	13276	13377,840687
	Point	Hungen	town	12918	13575,163269
	Point	Allendorf (Lumda)	town	4061	13738,416971
	Point	Betzenrod	village	673	14023,264931
	Point	Ermenrod	village	301	14327,42028
	Point	Otterbach	village	96	14580,680198
	Point	Zeilbach	village	235	14665,989102
	Point	Götzen	village	337	15114,126695
	Point	Schotten	town	11485	15237,783959
	Point	Homberg (Ohm)	town	7913	15371,99446
	Point	Feldatal	town	2850	15568,813208
	Point	Stumpertenrod	village	410	15830,126185

Quelle Attributtabelle aus ArcGIS 10.2

Abbildung 23 zeigt die Attributtabelle der erstellten Buffergeometrie mit den errechneten Statistiken zu der Einwohnerzahl innerhalb der festgelegten Distanz. Es ist zu sehen, dass innerhalb eines Umkreises von 50 km um die Stadt Grünberg 1.667.165 Menschen wohnen. Die niedrigste Einwohnerzahl eines Dorfes ist 5, während die Stadt Marburg mit 78.412 die höchste Einwohnerzahl der umliegenden Siedlungen hat. Durchschnittlich leben in den Siedlungen 4.457 Menschen. Anhand dieser Zahlen und den Entfernungen der einzelnen

Orte zu Grünberg könnte nun eine erste Einschätzung darüber gemacht werden, ob sich ein Einkaufszentrum oder eine neu Fabrik an diesem Standort lohnen würde.

Abbildung 23: Attributtabelle der erstellten Buffergeometrie

Buffer					
	Shape *	SUM popula	MEAN popul	MIN popula	MAX popula
▶	Polygon	1667165	4457,660428	5	78412

Quelle Attributtabelle aus ArcGIS 10.2

4 Online Geoverarbeitung

Die meisten Nutzer traditioneller Geoinformationssysteme nutzen oft nur einen kleinen Teil der vorhandenen Möglichkeiten. Nutzer verketteter, bereitgestellter Webservices hingegen können sich auf die Daten und Operationen konzentrieren die benötigt werden ohne bestimmte Programme und Funktionen zu installieren, zu lernen und zu bezahlen (Gehlot and Verbree, 2006, p.2). Die Definition des WPS führt also zu einer hohen Wiederverwendbarkeit einzelner Komponenten (Kiehle, Greve and Heier, 2006, p.8). Im Folgenden wird dementsprechend das erstellte Geoverarbeitungsmodell in einen Webservice umgewandelt.

4.1 Online Bereitstellung mit ArcGIS for Server

Geoverarbeitungsmodelle die innerhalb von ArcGIS Desktop mit dem ModelBuilder erstellt wurden, können unter dem Einsatz der Server Software ArcGIS for Server als Webservice bereitgestellt werden. Ein detailliertes Tutorial für die Bereitstellung eines Webservices mit ArcGIS for Server befindet sich im Anhang. Es ist zu sagen, dass die Bereitstellung über eine komfortable graphische Benutzeroberfläche geleitet wird. Die bereitgestellten Modelle finden sich unter:

Listing 1: URL der bereitgestellten Modelle auf ArcGIS for Server

http://zgis209.geo.sbg.ac.at/arcgis/manager/#f=MA_Brennecke_30.05.2015

Neben dem ‚PopulationDistance‘ Modell werden auch die in Kapitel 3.3 erstellten Module als Webservice auf dem Server bereitgestellt. Ein so erstellter Webservice wird sowohl SOAP als auch REST basiert bereitgestellt. Die WPS Funktionalität des Webservices kann optional aktiviert werden.

Grundsätzlich ist zu sagen, dass ArcGIS die REST Schnittstelle der Webservices nutzt um diese innerhalb von ArcMap auszuführen. Ein Geoverarbeitungsservice von ArcGIS Server wird wie ein normales Werkzeug ausgeführt. Die einzelnen Parameter werden über die graphische Benutzeroberfläche abgefragt. Zusätzlich zu dieser Ausführungsmethode ist es auch möglich den Webservice über den Browser auszuführen. Wie in Kapitel 2.3.2.2 beschrieben existiert für die REST Schnittstelle genau eine URL um den Service auszuführen. Bei dem ursprünglich erstellten Webservice ‚PopulationDistance‘ wurden bei der Online Bereitstellung auf dem Server die Eingabeparameter als Variable definiert um den Service so flexibel wie möglich zu halten. Im Zusammenhang mit der Ausführung des REST basierten Webservice im Browser müssen die beiden Geometrien leider kompliziert definiert werden. Um dieser Prozedur aus dem Weg zu gehen wurde für die beispielhafte Ausführung des

REST basierten Webservice im Browser, dasselbe Modell nochmal zusammen mit den Geometrien Grünberg und Hessen bereitgestellt. Dieser spezielle Webservice heißt ‚REST_PopulationDistance‘. Die URL dieses Services ist in Listing 2 zu sehen.

Listing 2: REST-URL des Webservices

http://zgis209.geo.sbg.ac.at/arcgis/rest/services/MA_Brennecke_30.05.2015/REST_PopulationDistance/GPServer/Population_Distance/submitJob

Im folgenden Kapitel wird die einfache Ausführung des erstellten WPS beschrieben. Anschließend werden zwei verschiedene Ansätze der Orchestrierung gezeigt.

4.2 Ausführen des WPS

Um einen WPS auszuführen stehen, wie in Kapitel 2.3.5.1 beschrieben, drei Operationen zu Verfügung. Mit dem GetCapabilities Request in Listing 3 werden die allgemeinen Fähigkeiten des Webservices abgefragt. Als Antwort auf diese Anfrage wird ein XML Dokument zurückgegeben, das direkt im Webbrowser untersucht werden kann. Je nachdem wie genau der Service dokumentiert ist, können hier Metadaten wie ein kurzes Abstrakt, Schlüsselwörter und Kontaktinformationen des Anbieters aufgeführt sein. Die gesamte Antwort auf den GetCapabilities Requests findet sich im Anhang.

Listing 3: GetCapabilities Request via HTTP Get

http://zgis209.geo.sbg.ac.at/arcgis/services/MA_Brennecke_30.05.2015/PopulationDistance/GPServer/WPSServer?SERVICE=WPS&ACCEPTVERSIONS=1.0.0&REQUEST=GetCapabilities

Die wichtigste Information der Antwort ist jedoch welche Prozesse der WPS genau zur Verfügung stellt. Gegen Ende des XML Dokuments findet sich der sogenannte Identifier, der in diesem Fall den Namen des Modells ‚PopulationDistance‘ besitzt. Mit diesem Identifier kann nun der DescribeProcess Request, wie in Listing 4 zu sehen, definiert werden. Die Antwort auf diese Anfrage beinhaltet wiederum die Informationen die nötig sind um den WPS Execute Request auszuführen. Die gesamte Antwort auf den DescribeProcess Requests findet sich im Anhang.

Listing 4: DescribeProcess Request via HTTP Get

http://zgis209.geo.sbg.ac.at/arcgis/services/MA_Brennecke_30.05.2015/PopulationDistance/GPServer/WPSServer?SERVICE=WPS&VERSION=1.0.0&REQUEST=DescribeProcess&Identifier=PopulationDistance

Mit den Informationen aus der Antwort dieser Anfrage lässt sich nun letztendlich der Execute Request definieren. Im Gegensatz zu den vorherigen Anfragen ist ein Execute Request etwas komplizierter zu erstellen, da in diesem die verschiedenen Eingangsdatensätze definiert werden müssen. Wenn der spezielle Service es zulässt, ist eine Ausführung auch durch einen HTTP-Get Befehl möglich. Dies ist jedoch nur für sehr einfache Services sinnvoll, da die Befehle, je nachdem wie viele Parameter definiert werden müssen, sehr lang werden kann. Grundsätzlich ist eine Ausführung eines WPS jedoch durch einen HTTP-Post Befehl vorgesehen. Um einen solchen Befehl auszuführen bedarf es einer speziellen Browser Erweiterung. In diesem Fall kommt der Postman REST Client

(Postdot Technologies, 2015) für Google Chrome zum Einsatz. Zunächst muss die URL des Services, wie in Listing 5 zu sehen, angegeben werden.

Listing 5: *URL des Execute Requests via HTTP Post*

http://zgis209.geo.sbg.ac.at/arcgis/services/MA_Brennecke_30.05.2015/PopulationDistance/GPServer/WPSServer

Der eigentliche Inhalt eines HTTP-Post Befehls wird mit einem XML Dokument definiert und ist in Listing 6 zu sehen. In diesem XML Dokument müssen zu Beginn einige allgemeine Informationen und XML Namensräume definiert werden, auf dessen Basis der Befehl anschließend ausgeführt werden kann. Diese beginnen mit dem XML-Tag `<wps:Execute>`. Anschließend werden die eigentlichen Informationen angegeben. Anschließend wird der auszuführende Prozess mit dem XML-Tag `<ows:Identifier>` definiert. Es handelt sich um den Prozess ‚PopulationDistance‘. Aus dem DescribeProcess Request wird ersichtlich, dass dieser Prozess vier verschiedene Inputs benötigt. Die Inputs sind mit dem XML-Tag `<wps:DataInputs>` umschlossen.

Danach kennzeichnet das XML-Tag `<ows:Identifier>` die einzelnen Inputs. Der erste Input ist ‚Selecting_Geometry‘. Aus dem DescribeProcess Dokument wird ersichtlich, dass es sich bei diesem Input um den Datentyp Complex Data handelt und im GML Format vorliegen muss. Der Input wird über ein WFS GetFeature Request zur Verfügung gestellt. Es handelt sich dabei um die Stadt Grünberg in Hessen. Der zweite Input ist ‚Input_Geometry‘ vom Typ Complex Data und wird ebenfalls durch einen WFS bereitgestellt. Es handelt sich um die Städte des Bundeslandes Hessen. Der dritte Input ist ‚Search_Distance‘ vom Typ Literal Data. Das bedeutet, dass es sich in diesem Fall um einen Zahlenwert handelt mit dem die Größe des Untersuchungsgebiets angegeben wird. Im Beispiel sind 50 km als Größe angegeben. Die Einheit ist in Kilometern angegeben kann jedoch, wie aus dem DescribeProcess Dokument ersichtlich, auch eine andere Einheit annehmen. Der vierte Input ist ‚Output_Coordinate_System‘. Hier wird die notwendige Projektion für die korrekte Distanzberechnung angegeben. Der vorgegebene Wert ist WGS 84, UTM Zone 32.

Schließlich wird mit dem XML-Tag `<wps:ResponseForm>` definiert, in welcher Form die Antwort auf diese Anfrage erfolgen soll. Das Tag `<wps:ResponseDocument storeExecuteResponse="true">` besagt, dass die Antwort des Execute Requests auf dem Server gespeichert werden soll. Der Tag `<wps:Output asReference="true">` zeigt an, dass der Output als Referenz angegeben werden soll. Dies bedeutet, dass letztendlich ein Link zu einem GML Feature angegeben wird. Im Anhang findet sich die gesamte Antwort auf den Execute Request. Mit dem Tag `<ows:Identifier>` werden die zu erzeugenden

Outputs ‚Buffer_Output‘ und ‚Near_Outpout‘ angegeben. Auch diese Informationen sind aus dem DescribeProcess Dokument ersichtlich. Mit dem Tag `</wps:Execute>` wird der gesamte Execute Request abgeschlossen.

Listing 6: XML Inhalt des Execute Requests via HTTP Post

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<wps:Execute xmlns:wps="http://www.opengis.net/wps/1.0.0" service="WPS"
version="1.0.0" language="en-US"
  xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xlink="http://www.w3.org/1999/xlink">
  <ows:Identifier>PopulationDistance</ows:Identifier>
  <wps>DataInputs>
    <wps:Input>
      <ows:Identifier>Selecting_Geometry</ows:Identifier>
      <wps:Reference
xlink:href="http://zgis209.geo.sbg.ac.at/arcgis/ser-
vices/MA_Brennecke_30.05.2015/Master/MapServer/WFSServer?SER-
VICE=WFS&VERSION=1.1.0&REQUEST=GetFeature&type-
Names=MA_Brennecke_30.05.2015_Master:Gruenberg" method="GET"/>
      </wps:Input>
      <wps:Input>
        <ows:Identifier>Input_Geometry</ows:Identifier>
        <wps:Reference
xlink:href="http://zgis209.geo.sbg.ac.at/arcgis/ser-
vices/MA_Brennecke_30.05.2015/Master/MapServer/WFSServer?SER-
VICE=WFS&VERSION=1.1.0&REQUEST=GetFeature&type-
Names=MA_Brennecke_30.05.2015_Master:Hessen" method="GET"/>
        </wps:Input>
      <wps:Input>
        <ows:Identifier>Search_Distance</ows:Identifier>
        <wps>Data>
          <wps:LiteralData uom="kilometers">50</wps:LiteralData>
        </wps>Data>
      </wps:Input>
      <wps:Input>
        <ows:Identifier>Output_Coordinate_System</ows:Identifier>
        <wps>Data>
          <wps:LiteralData>PROJCS['WGS_1984_UTM_Zone_32N',GE-
OGCS['GCS_WGS_1984',DATUM['D_WGS_1984',SPHE-
ROID['WGS_1984',6378137.0,298.257223563]],PRIMEM['Green-
wich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['Transverse_Mer-
cator'],PARAMETER['False_Easting',500000.0],PARAMETER['False_North-
ing',0.0],PARAMETER['Central_Meridian',9.0],PARAMETER['Scale_Fac-
tor',0.9996],PARAMETER['Latitude_Of_Origin',0.0],UNIT['Me-
ter',1.0]]</wps:LiteralData>
        </wps>Data>
      </wps:Input>
    </wps>DataInputs>
    <wps:ResponseForm>
      <wps:ResponseDocument storeExecuteResponse="true">
        <wps:Output asReference="true">
          <ows:Identifier>Buffer_Output</ows:Identifier>
        </wps:Output>
        <wps:Output asReference="true">
          <ows:Identifier>Near_Output</ows:Identifier>
        </wps:Output>
      </wps:ResponseDocument>
    </wps:ResponseForm>
  </wps:Execute>

```

4.3 Orchestrierung

Der erste Ansatz einer Orchestrierung von Webservices in dieser Arbeit folgte der Umsetzung von de Jesus et al. (2012), die in ihrer Arbeit, wie beschrieben, einen PyWPS mittels der Taverna Workbench orchestriert haben. Anschließend wird gezeigt wie eine Orchestrierung der erstellten Webservices, ArcGIS intern mit dem ModelBuilder umgesetzt werden kann.

4.3.1 Orchestrierung mit Taverna

Wie in Kapitel 2.5 beschrieben handelt es sich bei Taverna um ein Workflow Management System, das generisch genug ist um auch für Geo-Webservices eingesetzt zu werden. De Jesus et al. (2012) folgend wurde als erster Ansatz in dieser Arbeit versucht, einen mit ArcGIS erstellten WPS mit Taverna zu orchestrieren. Leider konnte dieser Ansatz nicht erfolgreich umgesetzt werden. Im Folgenden werden die aufgetretenen Probleme erläutert und die Vorgehensweise mit der Orchestrierung eines PyWPS in Taverna verglichen. Ein Tutorial für eine erfolgreiche Orchestrierung mit PyWPS findet sich im Anhang.

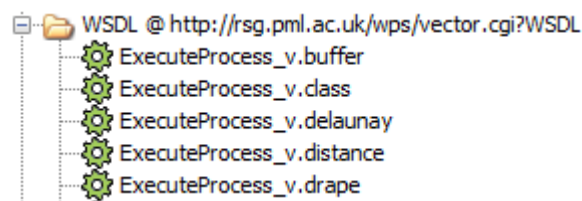
Grundsätzlich greift Taverna mit der SOAP-URL eines Webservices auf die WSDL Datei des Services zu und bindet diesen so in das System ein. Die SOAP-URL des in dieser Arbeit entstandenen Webservices ist in Listing 7 zu sehen.

Listing 7: SOAP-URL des Webservices

http://zgis209.geo.sbg.ac.at/arcgis/services/MA_Brennecke_30.05.2015/PopulationDistance/GPServer?wsdl

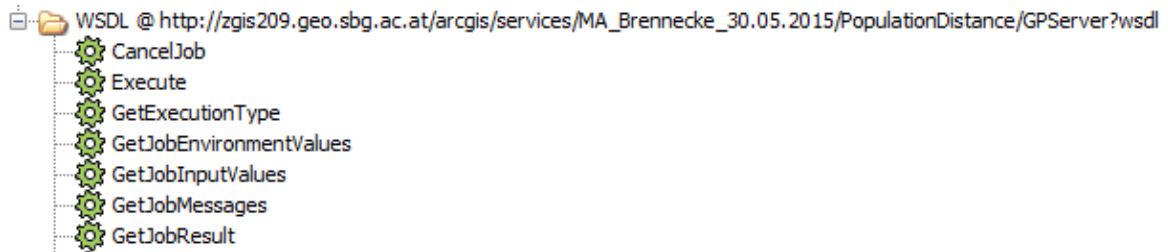
Im Vergleich zu der Einbindung der WSDL Datei eines WPS der Remote Sensing Group des Plymouth Marine Laboratories (PML) (Plymouth Marine Laboratory, 2015) zeigen sich schon auf den ersten Blick deutliche Unterschiede. Abbildung 24 und Abbildung 25 zeigen Screenshots der in Taverna eingebundenen WSDL Dateien der beiden Webservices.

Abbildung 24: Prozesse des PML WPS in Taverna



Quelle (Plymouth Marine Laboratory, 2015)

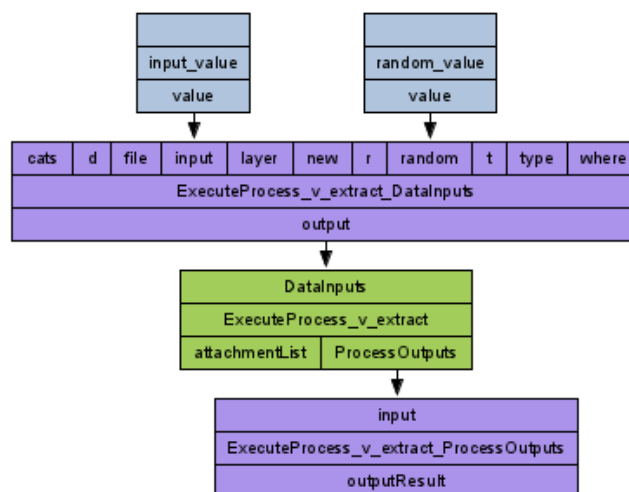
Abbildung 25: Prozesse des erstellten WPS in Taverna



Quelle Eigener Webservice

Das Problem ist, dass die Prozesse des PML WPS eindeutig ausgewiesen sind, während sich bei dem mit ArcGIS erstellten Webservice kein sinnvoller Prozess ausfindig machen lässt. Abbildung 26 zeigt die Ausführung des v.extract Prozesses des PML WPS in Taverna. Auch hier sind eindeutig die Parameter auszumachen, die für eine erfolgreiche Ausführung des WPS nötig sind. Der Prozess v.extract extrahiert in diesem Fall zufällig eine gewählte Anzahl von Punkten von der Ausgangsgeometrie. Demnach ist zum einen der Parameter ‚input‘ und zum anderen der Parameter ‚random‘ angegeben. Der Input ist mit einem Beispieldatensatz des PML definiert und der Parameter ‚random‘ mit dem Wert 50. Zum anderen besteht auch die Möglichkeit durch den Parameter ‚where‘ eine Abfrage bestimmter Punkte zu generieren. Im Hinblick auf die einfache Ausführung eines WPS, wie in Kapitel 4.2, ist zu sagen, dass die graphische Benutzeroberfläche einige Vorteile bietet und eine Orchestrierung in diesem Zusammenhang, auch durch andere Funktionen in Taverna, sehr Benutzerfreundlich ist.

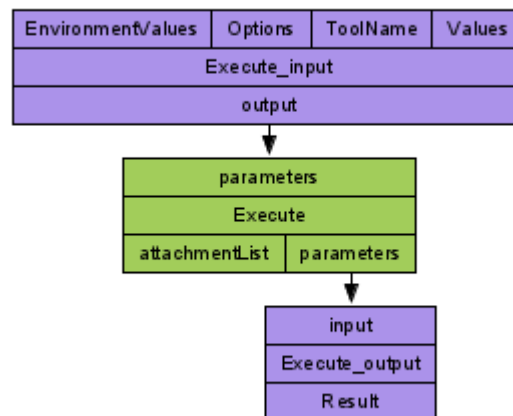
Abbildung 26: Ausführung des v.extract Prozesses des PML WPS



Quelle (Plymouth Marine Laboratory, 2015)

Abbildung 27 zeigt einen der durch die WSDL Datei in Taverna vorhandenen Prozesse des erstellten Webservices ‚Population Distance‘. Es ist zu sehen, dass im Gegensatz zu dem PML Prozess keine Parameter definiert sind mit denen eine Ausführung des Prozesses möglich wäre. Zusammenfassend ist demnach zu sagen, dass die automatische Erstellung eines für Taverna geeigneten WPS gescheitert ist.

Abbildung 27: Prozess des erstellten WPS in Taverna



Quelle Eigener Webservice

Die automatische Definition durch ArcGIS ist in diesem Zusammenhang nicht auf eine Verwendung in dieser Form abgestimmt. Das Scheitern der Orchestrierung mit Taverna ist letztendlich auf das zurückzuführen, was schon Stollberg und Zipf (2009, p.2) beschrieben. Die fehlende automatische Erstellung der WSDL Dateien für den WPS (siehe Kapitel 2.4.2). Der SOAP-basierte Webservice, der in diesem Fall eine WSDL Datei zur Verfügung stellt hat demnach in diesem Fall nichts mit der WPS Funktion des Services zu tun. Da das Ziel dieser Arbeit auch die Verwendung einer benutzerfreundlichen Orchestrierungsmethode ist, wird auf die komplizierte manuelle Definition von WSDL Dateien für den WPS oder den Einsatz von BPEL für die Orchestrierung verzichtet. Stattdessen wird die Orchestrierung in ArcGIS anwendungsintern mit dem ModelBuilder durchgeführt.

4.3.2 Alternative Orchestrierung mit dem ArcGIS Model Builder

Als alternative zu der Orchestrierung mit Taverna werden nun die Webservices der einzelnen Module innerhalb von ArcGIS mit dem ModelBuilder orchestriert. Dabei werden die einzelnen Webservices, wie bei der ursprünglichen Erstellung des Modells ‚Population Distance‘, wie normale Werkzeuge behandelt und dementsprechend miteinander verknüpft. Wie zuvor werden die Eingangsgeometrien, sowie die notwendige Projektion und Distanz,

als Parameter definiert. Das Ergebnis entspricht dem des ursprünglichen Modells. Schließlich wird das aus den einzelnen Webservices erstellte Modell auch auf ArcGIS for Server bereitgestellt. Auch diese bereitgestellte Orchestrierung kann in derselben Form ausgeführt werden wie die vorherigen und auch das Ergebnis ist letztendlich dasselbe. Bei der Nutzung der verschiedenen Webservices und Modelle fallen lediglich die unterschiedlichen Laufzeiten auf. Im folgenden Kapitel werden diese letztendlich miteinander verglichen um eine Aussage über die Effizienz der verschiedenen Webservices treffen zu können.

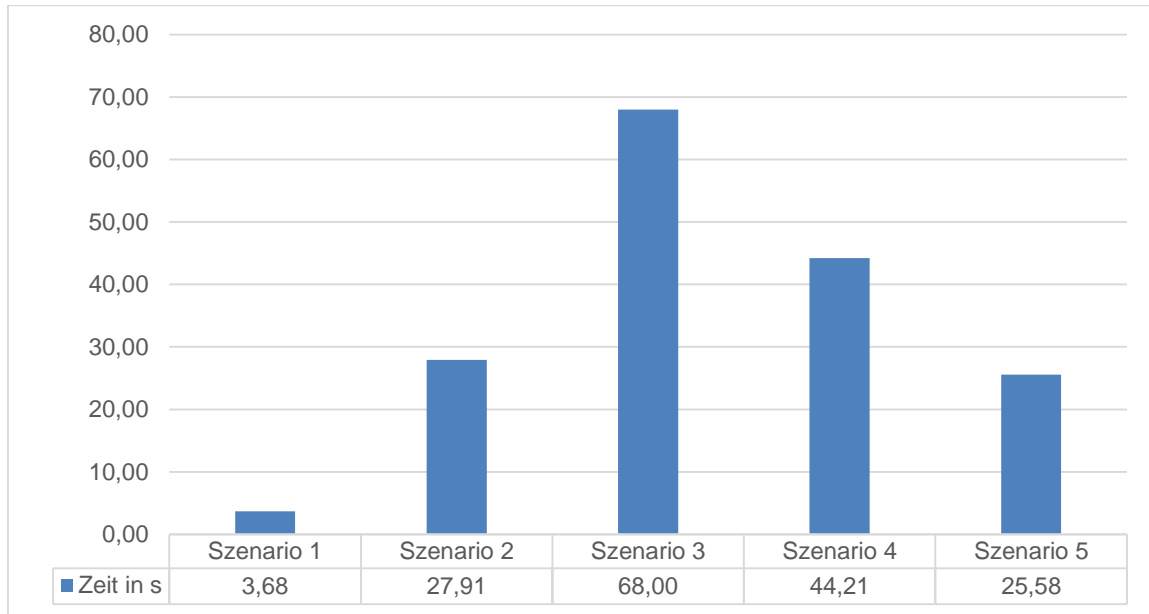
Bezugnehmend auf die theoretischen Ansätze der Orchestrierung von Webservices aus Kapitel 2.4.2 ist zu sagen, dass sich die von der OGC (2007c, p.7) für einen WPS vorgeschlagenen Möglichkeiten der Orchestrierung auch in dieser Methode wiederfinden lassen. Es handelt sich zwar nicht um eine Orchestrierung WPS, jedoch fungiert wie beschrieben, eine Webservice als übergeordnete Instanz der eine Sequenz von anderen Webservices aufruft und ausführt. Es handelt sich in diesem Fall auch um eine Form der Semitransparenten Verkettung bei der einzelne Prozesse zu einem Workflow zusammengefasst sind und der Benutzer weiß welche dies sind.

4.4 Performance der Orchestrierung

Um eine Aussage über die Effizienz der erstellten Webservices machen zu können, wird im Folgenden die Performance von fünf Szenarien miteinander verglichen. Szenario 1 stellt die einfache Ausführung des erstellten Geoverarbeitungsmodells ‚Population Distance‘ dar. In Szenario 2 wurde die Performance des bereitgestellten Services auf ArcGIS Server gemessen. Szenario 3 zeigt die Performance der mittels des ModelBuilders erstellten Orchestrierung der einzelnen Webservice Module. Szenario 4 stellt die Performance dieser Orchestrierung, die wieder als Webservice auf ArcGIS Server bereitgestellt wurde, dar. In Szenario 5 wurde die Zeit der einfachen Ausführung des ‚Population Distance‘ WPS gemessen. Die Zeitmessung von Szenario 1-4 basiert auf der ArcGIS internen Anzeige der Ausführungszeit. Die Zeitmessung von Szenario 5 basiert auf der Zeitmessung des Postman Rest Client, der zur Ausführung des WPS genutzt wurde. Die Protokolle der einzelnen Messungen finden sich im Anhang.

Abbildung 28 zeigt ein Säulendiagramm in dem die Zeitmessung der erläuterten Szenarien sowohl graphisch, als auch tabellarisch dargestellt ist. Die Y-Achse des Diagramms zeigt die Zeit der Ausführung in Sekunden und die X-Achse die fünf verschiedenen Szenarien. Es ist zu sagen, dass ausgehend von der einfachen Ausführung des erstellten Geoverarbeitungsmodells im ModelBuilder von ArcGIS deutliche Unterschiede in der Laufzeit zu sehen sind. Während die Berechnung in Szenario 1 nur 3,68 Sekunden dauert, sind es bei dem bereitgestellten Modell schon 27,91 Sekunden. Überraschend ist zu sehen, dass die Ausführung der orchestrierten Webservice Module im ModelBuilder in Szenario 3 länger dauert als die Ausführung desselben Modells, welches wieder auf ArcGIS Server bereitgestellt wurde. Zu erklären ist diese Differenz in der Ausführung damit, dass der Desktop Client in Szenario 3 jedes einzelne Modul vom Server laden und ausführen muss. In Szenario 4 kommt die gesamte Berechnung vom Server. Weiter ist interessant zu sehen, dass Szenario 2 und 5 ähnliche Laufzeiten haben. Es gibt also in diesem Fall keine signifikanten Unterschiede zwischen der Ausführung eines REST basierten Webservices und der Ausführung des WPS.

Abbildung 28: Ergebnisse der Performancemessung



Quelle Eigene Darstellung

Ausgehend von der zu Beginn dieser Arbeit definierten Forschungsfrage „Wie wirkt sich die Modularisierung von Geo-Webservices auf deren Effizienz aus?“ ist zu sagen, dass sich durch die Modularisierung solcher Webservices deutliche Unterschiede in den Laufzeiten ergeben. Aus dem speziellen Anwendungsfall heraus, indem einzelne bereitgestellte Module mittels des ModelBuilders orchestriert und wiederum online bereitgestellt wurden, ergibt sich, dass der Austausch von Geoverarbeitungsmöglichkeiten auf andere Weise effizienter zu sein scheint. So bietet ArcGIS z.B. die Möglichkeit Geoverarbeitungsmodelle als sogenanntes ‚Geoprocessing Package‘ zu teilen. Nichtsdestotrotz kann es sinnvoll sein einzelne Module als Webservice zur Verfügung zu stellen, wenn die einzelnen Modelle eine gewisse Komplexität besitzen die nicht ohne weiteres umgesetzt werden kann.

5 Fazit

In dieser Masterarbeit „Geoverarbeitung im Wandel: Vom Desktop in das World Wide Web“ sollte zunächst ein Offline Workflow entwickelt werden, der anschließend modularisiert und wieder orchestriert sowie letztendlich als Webservice bereitgestellt werden sollte.

Um auf die von Brauner et al. (2009) eingeführte Geoverarbeitungsagenda zurückzukommen (siehe Abbildung 1), kann im Zusammenhang mit dieser Arbeit folgendes festgehalten werden. Bezüglich der **Orchestrierung** wurde versucht dem anwenderfreundlichen Ansatz von de Jesus et al. (2012) zu folgen und diesen auf mit ArcGIS erstellte Webservices zu übertragen. Leider scheiterte dieser Versuch der Orchestrierung eines mit ArcGIS erstellten WPS in der Taverna Workbench. Die Gründe dafür liegen in der fehlenden automatischen Erstellung von WSDL Dateien für jeden einzelnen Prozess eines WPS, was auch Stollberg und Zipf (2009) in Ihrer Arbeit beschrieben haben. Da das Ziel dieser Arbeit auch die Verwendung einer benutzerfreundlichen Orchestrierungsmethode war, wurde auf die komplizierte manuelle Definition von WSDL Dateien für den WPS oder den Einsatz von BPEL für die Orchestrierung verzichtet. Stattdessen wurde die Orchestrierung in ArcGIS anwendungsintern mit dem ModelBuilder durchgeführt.

Anhand dieser Orchestrierung wurde schließlich die **Performance** der erstellten Webservices gemessen und miteinander verglichen. Die Antwort auf die Forschungsfrage „**Wie wirkt sich die Modularisierung von Geo-Webservices auf deren Effizienz aus?**“ die zu Beginn dieser Arbeit gestellt wurde lautet demnach wie folgt: Letztendlich ist die Desktop Ausführung des erstellten Geoverarbeitungsmodells der Orchestrierung der Webservices deutlich überlegen. An dieser Stelle eignet sich der Austausch des Modells auf andere Art und Weise deutlich mehr. Auch ist es schwer diese Nachteile mit den Vorteilen, wie Verfügbarkeit, Installationsaufwand, weniger Schulungsaufwand aufzuwiegen. Sinnvoll ist die angewandte Methode allenfalls, wenn es sich um sehr komplexe Geoverarbeitungsmodelle handelt, die nicht ohne weiteres nachgebaut werden können.

6 Literaturverzeichnis

52°North Initiative for Geospatial Open Source Software GmbH, 2014. *52°North WPS*. [online] Available at: <<http://52north.org/communities/geoprocessing/wps/index.html>> [Accessed 31 Jan. 2015].

Abel, D.J., Ooi, B.C., Tan, K.-L. and Tan, S.H., 1998. Towards integrated geographical information processing. *International Journal of Geographical Information Science*, 12(4), pp.353–371.

Alameh, N., 2003. Chaining geographic information web services. *IEEE Internet Computing*, 7(5), pp.22–29.

Andrae, C., 2012. *Simple features: die OpenGIS- und ISO-Spezifikationen für Verwaltung, Analyse und Übergabe einfacher Geoobjekte in modularen Architekturen und SQL-Datenbanken*. Berlin: Wichmann.

Baranski, B., 2008. Grid computing enabled web processing service. *GI-Days, Münster*, p.12.

Bartelme, N., 2005. *Geoinformatik: Modelle, Strukturen, Funktionen*. Berlin: Springer.

Baun, C., Kunze, M., Nimis, J. and Tai, S., 2011. *Cloud Computing*. Informatik im Fokus. [online] Berlin, Heidelberg: Springer Berlin Heidelberg. Available at: <<http://link.springer.com/10.1007/978-3-642-18436-9>> [Accessed 7 Oct. 2014].

Bernard, L., Craglia, M., Gould, M. and Kuhn, W., 2005. Towards an SDI research agenda. In: *11th EC GIS & GIS Workshop-ESDI: Setting the Framework-Abstracts Handbook*. [online] pp.147–151. Available at: <http://www.territorisardi.it/documenti/6_83_20060201090548.pdf#page=155> [Accessed 22 Jan. 2015].

Bernard, L., Mäs, S., Müller, M., Henzen, C. and Brauner, J., 2014. Scientific geodata infrastructures: challenges, approaches and directions. *International Journal of Digital Earth*, 7(7), pp.613–633.

Berners-Lee, T., 1999. *Weaving the Web: the original design and ultimate destiny of the World Wide Web by its inventor*. 1st ed ed. San Francisco: HarperSanFrancisco.

Brauner, J., Foerster, T., Schaeffer, B. and Baranski, B., 2009. Towards a research agenda for geoprocessing services. In: *12th AGILE International Conference on Geographic Information Science*. [online] Hanover, Germany: IKG, Leibniz University of Hanover. Available at: <http://m.agile-online.org/Conference_Paper/CDs/agile_2009/AGILE_CD/pdfs/124.pdf> [Accessed 29 Sep. 2014].

Bundesamt für Kartographie und Geodäsie, 2014a. *GDI-DE: Geodaten vernetzen*. [online] GDI-DE: Geodateninfrastruktur Deutschland. Available at: <http://www.geoportal.de/DE/GDI-DE/gdi-de_artikel.html%3bjsessionid=E843C46A7E0471BA7C12558CE13CA5A1?lang=de> [Accessed 6 Oct. 2014].

Bundesamt für Kartographie und Geodäsie, 2014b. *GDI-DE: INSPIRE*. [online] Grenzen aufheben mit INSPIRE. Available at: <<http://www.geoportal.de/DE/GDI-DE/INSPIRE/inspire.html?lang=de>> [Accessed 5 Dec. 2014].

Cepicky, J., 2009. *PyWPS*. [online] Welcome to PyWPS. Available at: <<http://pywps.wald.intevation.org/index.html>> [Accessed 31 Jan. 2015].

Coleman, D.J., Georgiadou, Y., Labonte, J. and others, 2009. Volunteered Geographic Information: the nature and motivation of producers. *International Journal of Spatial Data Infrastructures Research*, 4(1), pp.332–358.

Craglia, M. and Johnston, A., 2004. Assessing the impacts of spatial data infrastructures: Methods and gaps. In: *7th AGILE Conference on Geographic Information Science, Heraklion, Greece*. [online] Available at: <http://plone.itc.nl/agile_old/Conference/greece2004/papers/1-1-1_Craglia.pdf> [Accessed 22 Sep. 2014].

Crompvoets, J. and Bregt, A., 2003. World status of national spatial data clearinghouses. *URISA journal*, 15(1), pp.43–50.

Dengel, A., 2012. *Semantische Technologien Grundlagen - Konzepte - Anwendungen*. Heidelberg: Spektrum Akademischer Verl.

Eberle, J. and Strobl, C., 2012. Web-based geoprocessing and workflow creation for generating and providing remote sensing products. *Geomatica*, 66(1), pp.13–26.

ESRI, 2014a. *ArcGIS Help 10.2 - Copy (Data Management)*. [online] Available at: <<http://resources.arcgis.com/en/help/main/10.2/index.html#//001700000051000000>> [Accessed 21 Feb. 2015].

ESRI, 2014b. *ArcGIS Help 10.2 - Near (Analysis)*. [online] Available at: <<http://resources.arcgis.com/en/help/main/10.2/index.html#//00080000001q000000>> [Accessed 21 Feb. 2015].

ESRI, 2014c. *ArcGIS Help 10.2 - Project (Data Management)*. [online] Available at: <<http://resources.arcgis.com/en/help/main/10.2/index.html#//00170000007m000000>> [Accessed 21 Feb. 2015].

ESRI, 2014d. *ArcGIS Help 10.2 - Select Layer By Location (Data Management)*. [online] Available at: <<http://resources.arcgis.com/en/help/main/10.2/index.html#//001700000072000000>> [Accessed 21 Feb. 2015].

ESRI, 2014e. *ArcGIS Help 10.2 - Was ist Geoverarbeitung?*. [online] Available at: <<http://resources.arcgis.com/de/help/main/10.2/index.html#//002s00000001000000>> [Accessed 31 Jan. 2015].

ESRI, 2015a. *ArcGIS for Desktop*. [online] Available at: <<http://www.esri.com/software/arcgis/arcgis-for-desktop>> [Accessed 13 Feb. 2015].

ESRI, 2015b. *ArcGIS for Server*. [online] Available at: <<http://www.esri.com/software/arcgis/arcgisserver>> [Accessed 13 Feb. 2015].

Fielding, R.T., 2000. *Architectural Styles and the Design of Network-based Software Architectures*. [online] University of California, Irvine. Available at: <http://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf> [Accessed 13 Feb. 2015].

Friis-Christensen, A., Ostländer, N., Lutz, M. and Bernard, L., 2007. Designing service architectures for distributed geoprocessing: Challenges and future directions. *Transactions in GIS*, 11(6), pp.799–818.

Fu, P., 2011. *Web GIS: principles and applications*. 1st ed ed. Redlands, California: ESRI Press.

Gehlot, S. and Verbree, E., 2006. Web-based sharing of a geo-processing chain: combination and dissemination of data and services. In: *Proceedings of the International Symposium on Geospatial databases for Sustainable Development*, eds. S. Nayak & SK Pathan, ISPRS

TC Committee IV Geo-databases and Digital Mapping, Goa, India. [online] Available at: <http://www.gdmc.nl/publications/2006/Web-based_sharing_geo-processing_chain.pdf> [Accessed 21 Nov. 2014].

Geodaten Infrastruktur Brandenburg ed., 2005. *Übersicht der ISO Standards zu Geographischen Informationen / Geomatik*. Available at: <http://www.geobasis-bb.de/GeoPortal1/pdf/GIB_Uebersicht_ISO_Standards.pdf>.

Google, 2015. *Chrome-Browser*. [online] Available at: <<https://www.google.de/chrome/browser/desktop/index.html>> [Accessed 13 Feb. 2015].

Hofer, B., 2013. Geospatial Cyberinfrastructure and Geoprocessing Web—A Review of Commonalities and Differences of E-Science Approaches. *ISPRS International Journal of Geo-Information*, 2(3), pp.749–765.

ISO, 2005. *ISO 19119:2005 Geographic Information Services, ISO TC 211*. [online] Available at: <<http://www.isotc211.org/>> [Accessed 20 Jan. 2015].

ISO, 2014. *About ISO*. [online] Available at: <<http://www.iso.org/iso/home/about.htm>> [Accessed 4 Dec. 2014].

De Jesus, J., Walker, P., Grant, M. and Groom, S., 2012. WPS orchestration using the Taverna workbench: The eScience approach. *Computers & Geosciences*, 47, pp.75–86.

Kiehle, C., Greve, K. and Heier, C., 2006. Standardized geoprocessing—taking spatial data infrastructures one step further. In: *Proceedings of the 9th AGILE International Conference on Geographic Information Science. Visegrád, Hungary*. [online] Available at: <http://m.agile-online.org/Conference_Paper/CDs/agile_2006/papers/a273.pdf> [Accessed 21 Nov. 2014].

Kiehle, C., Greve, K. and Heier, C., 2007. Requirements for Next Generation Spatial Data Infrastructures-Standardized Web Based Geoprocessing and Web Service Orchestration. *Transactions in GIS*, 11(6), pp.819–834.

Lange, N. d., 2006. *Geoinformatik in Theorie und Praxis*. Berlin: Springer.

LANIG, S. and ZIPF, A., 2010. Standardkonforme Geodatenverarbeitung und Dienstorientierung am Beispiel der Solarpotenzialanalyse mittels OGC Web Processing Service (WPS). *Angewandte Geoinformatik 2010*. [online] Available at: <<https://www.vde-verlag.de/proceedings-en/537495123.html>> [Accessed 6 Nov. 2014].

Melzer, I., 2010. *Service-orientierte Architekturen mit Web Services: Konzepte - Standards - Praxis*. Heidelberg: Spektrum, Akad. Verl.

Michael, C. and Ames, D.P., 2007. Evaluation of the OGC Web Processing Service for Use in a Client-Side GIS. *OSGeo Journal*, [online] 1(1). Available at: <<https://foss4g2008.org/index.php/journal/article/download/69/55>> [Accessed 14 Oct. 2014].

Müller, M., Bernard, L. and Kadner, D., 2013. Moving code – Sharing geoprocessing logic on the Web. *ISPRS Journal of Photogrammetry and Remote Sensing*, 83, pp.193–203.

Natural Earth, 2015. *Natural Earth*. [online] Available at: <<http://www.naturalearthdata.com/>> [Accessed 19 Feb. 2015].

Neis, P., Zielstra, D., Zipf, A. and Strunck, A., 2010. Empirische Untersuchungen zur Datenqualität von OpenStreetMap-Erfahrungen aus zwei Jahren Betrieb mehrerer OSM-Online-Dienste. *Angewandte Geoinformatik 2010*. [online] Available at: <<https://www.vde-verlag.de/proceedings-en/537495055.html>> [Accessed 23 Sep. 2014].

- OASIS, 2007. *Web Services Business Process Execution Language*. [online] Available at: <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html#_Toc164738475> [Accessed 13 Feb. 2015].
- OGC, 2002. The OpenGIS Abstract Specification. Topic 12: OpenGIS Service Architecture. *Version 4.3*, [online] OGC 02-112. Available at: <http://pdi.topografia.upm.es/m.manso/docencia/IDE_plan92_ITT/IDES2009/6_Metadatos_Servicios/ISO19119_OGC_Service_Architecture.pdf> [Accessed 4 Dec. 2014].
- OGC, 2005. Web Feature Service (WFS) Implementation Specification. *Version: 1.1.0*, OGC 04-094.
- OGC, 2006. Web Map Server (WMS) Implementation Specification. *Version: 1.3.0*, OGC 06-042.
- OGC, 2007a. Catalogue Services Specification (Standard). *Version 2.0.2*, OGC 07-006r1.
- OGC, 2007b. Geography Markup Language (GML) Encoding Standard. *Version: 3.2.1*, OGC 07-036.
- OGC, 2007c. Web Processing Service (Standard). *Version 1.0.0*, OGC 05-007r7.
- OGC, 2009. The OpenGIS Abstract Specification. Topic 5: Features. *Version: 5.0*, OGC 08-126.
- OGC, 2012. Web Coverage Service (WCS) Interface Standard. *Version 2.0.1*, OGC 09-110r4.
- OGC, 2014. *Web Processing Service | OGC*. [online] Available at: <<http://www.opengeospatial.org/standards/wps>> [Accessed 28 Aug. 2014].
- OGC, 2015. *Open Geospatial Consortium | OGC*. [online] Available at: <<http://www.opengeospatial.org/>> [Accessed 18 Jan. 2015].
- OSM, 2015. *OpenStreetMap Deutschland: Die freie Wiki-Weltkarte*. [online] Available at: <<http://www.openstreetmap.de/index.html>> [Accessed 31 Jan. 2015].
- Plymouth Marine Laboratory, 2015. *Plymouth Marine Laboratory WPS*. [online] Available at: <<https://rsg.pml.ac.uk/wps/index.html>> [Accessed 31 Jan. 2015].
- Poorazizi, M.E. and Hunter, A.J.S., 2014. *Evaluation of Web Processing Service Frameworks*.
- Poore, B.S., 2011. Users as essential contributors to spatial cyberinfrastructures. *Proceedings of the National Academy of Sciences*, 108(14), pp.5510–5515.
- Postdot Technologies, 2015. *Postman | Supercharge your API workflow*. [online] Available at: <<http://www.getpostman.com/>> [Accessed 13 Feb. 2015].
- Schaeffer, B., 2008. Towards a transactional web processing service. *Proceedings of the GI-Days, Münster*. [online] Available at: <<http://www.gi-tage.de/archive/2008/downloads/acceptedPapers/Papers/Schaeffer.pdf>> [Accessed 17 Nov. 2014].
- Shade, S., Ostländer, N., Granell, C., Shulz, M., McInerney, D., Dubois, G., Vaccari, L., Chinosi, M., Díaz, L., Bastin, L. and others, 2012. Which Service Interfaces fit the Model Web? In: *GEOProcessing 2012, The Fourth International Conference on Advanced Geographic Information Systems, Applications, and Services*. [online] pp.1–6. Available at:

<http://www.thinkmind.org/index.php?view=article&articleid=geoprocessing_2012_1_10_30081> [Accessed 29 Sep. 2014].

Staub, P., 2009. *Über das Potenzial und die Grenzen der semantischen Interoperabilität von Geodaten: Ein operationelles Verfahren zur Nutzung verteilter Systeme in Geodaten-Infrastrukturen*. [online] ETH Zürich. Available at: <<http://e-collection.library.ethz.ch/eserv/eth:41507/eth-41507-02.pdf>> [Accessed 18 Jan. 2015].

Stewart, C.A., Simms, S., Plale, B., Link, M., Hancock, D.Y. and Fox, G.C., 2010. What is cyberinfrastructure. In: *Proceedings of the 38th annual ACM SIGUCCS fall conference*. [online] ACM, pp.37–44. Available at: <<http://dl.acm.org/citation.cfm?id=1878347>> [Accessed 25 Sep. 2014].

Stollberg, B. and Zipf, A., 2007. OGC web processing service interface for web service orchestration aggregating geo-processing services in a bomb threat scenario. In: *Web and Wireless Geographical Information Systems*. [online] Springer, pp.239–251. Available at: <http://link.springer.com/chapter/10.1007/978-3-540-76925-5_18> [Accessed 18 Nov. 2014].

Stollberg, B. and Zipf, A., 2009. Development of a WPS Process Chaining Tool and Application in a Disaster Management Use Case for Urban Areas. *Urban and Regional Data Management: UDMS 2009 Annual*, p.269.

Taverna, 2010. *What is a Workflow Management System? | Taverna*. [online] Available at: <<http://www.taverna.org.uk/introduction/what-is-a-workflow-management-system/>> [Accessed 10 Jan. 2015].

Taverna, 2015. *Taverna - open source and domain independent Workflow Management System*. [online] Available at: <<http://www.taverna.org.uk/>> [Accessed 13 Feb. 2015].

W3C, 2004a. Web services architecture. *Standard, W3C World*, p.10.

W3C, 2004b. *XML Schema Part 2: Datatypes Second Edition*. [online] Available at: <<http://www.w3.org/TR/xmlschema-2/#built-in-primitive-datatypes>> [Accessed 31 Jan. 2015].

W3C, 2014. *About W3C*. [online] Available at: <<http://www.w3.org/Consortium/>> [Accessed 8 Oct. 2014].

Wolstencroft, K., Haines, R., Fellows, D., Williams, A., Withers, D., Owen, S., Soiland-Reyes, S., Dunlop, I., Nenadic, A., Fisher, P., Bhagat, J., Belhajjame, K., Bacall, F., Hardisty, A., Nieva de la Hidalga, A., Balcazar Vargas, M.P., Sufi, S. and Goble, C., 2013. The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud. *Nucleic Acids Research*, 41(W1), pp.W557–W561.

Yang, C., Raskin, R., Goodchild, M. and Gahegan, M., 2010. Geospatial Cyberinfrastructure: Past, present and future. *Computers, Environment and Urban Systems*, 34(4), pp.264–277.

Zhao, P., Foerster, T. and Yue, P., 2012. The Geoprocessing Web. *Computers & Geosciences*, 47, pp.3–12.

7 Anhang

7.1 Attributtabelle des Ergebnisses der Near Analyse

Shape *	name	type	population	NEAR_DIST
Point	Grünberg	town	13828	0,000044
Point	Queckborn	suburb	1418	2682,79896
Point	Mücke	town	10187	5492,20177
Point	Laubach	town	10330	5833,13444
Point	Merlau	village	1184	6035,6948
Point	Groß-Eichen	village	954	8222,85121
Point	Wettsaasen	village	212	8481,42979
Point	Reiskirchen	town	10784	9375,98933
Point	Ruppertenrod	village	856	9568,65965
Point	Sellnrod	village	809	9621,725
Point	Rabenau	town	5396	9840,70906
Point	Höckersdorf	village	250	10385,7391
Point	Ober-Ohmen	village	790	11343,0136
Point	Bobenhausen II	village	436	12440,635
Point	Gemünden	town	3216	12622,5922
Point	Lich	town	13557	12719,5111
Point	Hainbach	village	300	13115,5407
Point	Fernwald	town	6680	13244,0469
Point	Buseck	town	13276	13377,8407
Point	Hungen	town	12918	13575,1633
Point	Allendorf (Lumda)	town	4061	13738,417
Point	Betzenrod	village	673	14023,2649
Point	Ermenrod	village	301	14327,4203
Point	Otterbach	village	96	14580,6802
Point	Zeilbach	village	235	14665,9891
Point	Götzen	village	337	15114,1267
Point	Schotten	town	11485	15237,784
Point	Homberg (Ohm)	town	7913	15371,9945
Point	Feldatal	town	2850	15568,8132
Point	Stumpertenrod	village	410	15830,1262
Point	Groß-Felda	village	966	16326,9356
Point	Ulrichstein	town	3257	16744,5313
Point	Ehringshausen	village	734	16829,6637
Point	Kestrich	village	315	16873,1306
Point	Rudingshain	village	412	16920,8886
Point	Ilshausen	village	106	17109,4725
Point	Michelbach	village	412	17153,6173
Point	Köddingen	village	440	17927,0419
Point	Staufenberg	town	8233	18114,9687
Point	Ebsdorfergrund	town	9001	18223,7231
Point	Windhausen	village	293	18309,666
Point	Wingershausen	suburb	426	18392,9449
Point	Pohlheim	town	17859	18505,6052
Point	Busenborn	village	239	18761,9882
Point	Lollar	town	10169	19064,5264
Point	Berstadt	village	1730	19597,0474
Point	Rebgeshain	village	265	19699,9182
Point	Erbenhausen	village	274	19873,6686
Point	Gießen	town	75349	20223,0096
Point	Münzenberg	town	5700	20282,4764
Point	Nidda	town	18189	20336,5557
Point	Meiches	village	550	21261,1763
Point	Engelrod	village	665	21429,9881
Point	Ober-Breidenbach	village	381	21464,2151
Point	Lehrbach	village	357	21696,5657
Point	Heimertshausen	village	342	21834,8586
Point	Ober-Gleen	village	547	22089,4482
Point	Wolfshausen	village	357	22248,9512
Point	Wettenberg	town	12470	22322,0723
Point	Kirtorf	town	3532	22329,0239
Point	Roth	village	805	22510,906
Point	Fronhausen	town	3983	22513,1929
Point	Strebendorf	village	286	22609,7262
Point	Eichelhain	village	256	22657,4683
Point	Linden	town	12271	22663,3881
Point	Storndorf	village	795	22682,406
Point	Zell	village	791	22712,221
Point	Romrod	town	3080	22834,7074
Point	Kaulstoß	village	202	22999,2256
Point	Echzell	town	5925	23022,9112
Point	Hörgenau	village	284	23121,6834
Point	Langgöns	town	11719	23150,5245
Point	Lang-Göns	suburb	6526	23210,7303
Point	Argenstein	village	370	23246,4715

Point	Heuchelheim	town	7634	23363,2963
Point	Wölfersheim	town	9593	23445,085
Point	Sichenhausen	village	253	23558,4439
Point	Vadenrod	village	365	23577,0982
Point	Rockenberg	town	4009	23914,5888
Point	Eichenrod	village	346	24038,8209
Point	Niederwalgern	village	1494	24201,0277
Point	Billertshausen	village	228	24248,3026
Point	Wenkbach	village	690	24312,3215
Point	Dirlammen	village	453	24319,0922
Point	Vetzberg	village	844	24712,9132
Point	Niederweimar	village	2576	24779,1307
Point	Lanzenhain	village	582	24806,8729
Point	Ohmes	village	566	24909,744
Point	Hopfmansfeld	village	351	24974,9219
Point	Herchenhain	village	610	25075,1422
Point	Gedern	town	7855	25130,8483
Point	Hüttenberg	town	10443	25220,5078
Point	Liederbach	village	560	25251,4687
Point	Weimar (Lahn)	town	7053	25264,6191
Point	Hirzenhain	village	2994	25334,9145
Point	Biebertal	town	10285	25335,0784
Point	Stedebach	village	25	25377,916
Point	Kirchhain	town	16367	25688,9125
Point	Dutenhofen	suburb	3152	25713,1379
Point	Angenrod	village	581	25831,4797
Point	Rodheim-Bieber	village	5200	25854,6627
Point	Schwalmtal	town	2850	26091,4497
Point	Hartmannshain	village	293	26097,0125
Point	Ober-Sorg	village	116	26129,8095
Point	Oberweimar	village	680	26141,5734
Point	Fellingshausen	village	1611	26262,1963
Point	Ranstadt	village	5023	26348,9627
Point	Allmenrod	village	380	26369,0977
Point	Stadtallendorf	town	21523	26495,524
Point	Unter-Sorg	village	60	26742,7659
Point	Kehna	village	90	26745,5876
Point	Leusel	village	825	26815,1908
Point	Hopfgarten	village	286	26875,0297
Point	Reichelsheim	town	6920	26875,0503
Point	Germershausen	isolated_dwelin	10	26880,3235
Point	Butzbach	town	25328	26970,6772
Point	Seibelsdorf	village	344	27025,7881
Point	Dornholzhausen	suburb	1064	27107,6775
Point	Wallenrod	village	818	27127,0376
Point	Ortenberg	town	9222	27254,6217
Point	Antriftal	town	2007	27514,2472
Point	Niederkleen	suburb	1216	27543,1246
Point	Herbstein	town	5159	27625,3999
Point	Renzendorf	village	195	27646,659
Point	Hergersdorf	village	190	27709,2856
Point	Nieder-Weisel	village	2492	27752,1455
Point	Marburg	town	78412	27759,2432
Point	Lahnau	town	8325	27882,1269
Point	Ober-Mörlen	town	6005	27884,536
Point	Sickendorf	village	197	27943,9068
Point	Vockenrod	village	349	28073,8041
Point	Lohra	town	5779	28076,591
Point	Alsfeld	town	16351	28184,4297
Point	Brauerschwend	village	629	28373,4197
Point	Bermuthshain	village	659	28413,6359
Point	Reibertenrod	village	304	28739,4445
Point	Frischborn	village	1118	28783,7017
Point	Heblös	village	370	28914,0444
Point	Grebenhain	town	5204	29030,8327
Point	Bad Nauheim	town	31176	29058,187
Point	Waldgirmes	village	3324	29069,6549
Point	Ostheim	village	1038	29234,5946
Point	Staden	suburb	813	29281,7613
Point	Rainrod	village	431	29789,8656
Point	Schwabenrod	village	384	29916,1124
Point	Oberkleen	suburb	1195	29944,7056
Point	Weiershausen	village	80	30015,0905
Point	Crainfeld	village	413	30024,49
Point	Vaitshain	village	97	30057,4009
Point	Bindsachsen	village	920	30290,5633
Point	Rimlos	village	270	30328,4114
Point	Eudorf	village	490	30371,1224
Point	Blitzenrod	village	680	30612,1542
Point	Garbenheim	suburb	2080	30779,4845
Point	Neustadt (Hessen)	town	9115	30808,5535
Point	Altenschlirf	village	476	30856,8242
Point	Wehrda	suburb	6558	30883,9852
Point	Glauberg	village	3221	30890,6556
Point	Nösberts-Weidmoos	village	225	30896,198

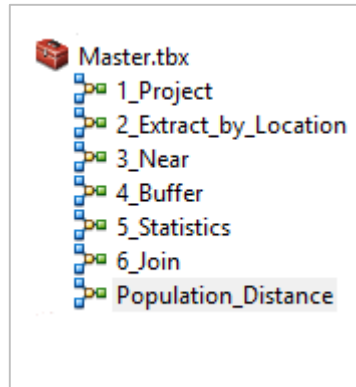
Point	Naunheim	suburb	3882	30927,2975
Point	Maar	village	1822	31181,6239
Point	Florstadt	town	8784	31188,3346
Point	Cölbe	town	6944	31263,474
Point	Blasbach	suburb	994	31435,4592
Point	Lauterbach	town	14462	31455,3804
Point	Friedberg (Hessen)	town	27543	31879,7866
Point	Bernsdorf	hamlet	46	31959,471
Point	Espa	suburb	666	32020,638
Point	Hohenahr	town	5096	32204,2753
Point	Rudlos	village	79	32345,8925
Point	Eifa	village	753	32407,5558
Point	Kefenrod	village	3038	32466,1911
Point	Steinfurt	village	233	32478,9644
Point	Cleeberg	suburb	1052	32530,975
Point	Nesselbrunn	village	90	32554,5669
Point	Rauschenberg	town	4600	32613,7965
Point	Nieder-Moos	village	400	32656,0921
Point	Michelbach	village	2067	33155,2326
Point	Gladenbach	town	12548	33235,9398
Point	Hattendorf	village	574	33317,6884
Point	Willingshausen	town	5294	33370,5372
Point	Heisters	village	105	33442,893
Point	Metzlos	village	186	33451,7867
Point	Elbenrod	village	415	33674,0514
Point	Metzlos-Gehaag	village	200	33677,9621
Point	Büches	village	600	33683,4933
Point	Goßfelden	village	2387	33827,0421
Point	Wernges	village	240	33967,8933
Point	Wiesental	village	190	33972,084
Point	Wünschen-Moos	village	54	34087,1936
Point	Salz	village	350	34113,0203
Point	Altenstadt	village	11938	34257,9617
Point	Angersbach	village	2707	34423,6255
Point	Rinderbügen	village	1087	34602,7755
Point	Zahmen	village	154	34635,9168
Point	Büdingen	town	21551	34902,4673
Point	Aßlar	town	13600	35068,7925
Point	Lahntal	town	7018	35148,4488
Point	Diedenshausen	village	140	35296,786
Point	Schrecksbach	town	3354	35339,4067
Point	Niddatal	town	9070	35417,8387
Point	Schwarz	village	720	35597,1626
Point	Josbach	village	400	35917,8844
Point	Reichlos	village	140	36182,4088
Point	Lingelbach	village	712	36317,668
Point	Erbstadt	suburb	1400	36317,7803
Point	Freiensteinau	town	3438	36387,056
Point	Wöllstadt	village	6239	36412,8113
Point	Limeshain	village	5469	36417,7755
Point	Bechlingen	suburb	750	36443,8812
Point	Birstein	village	6539	36444,3937
Point	Grebenua	town	2657	36590,8648
Point	Bermoll	suburb	255	36595,7906
Point	Jossa	village	482	36607,4509
Point	Landenhausen	village	1419	36617,9797
Point	Blankenau	village	763	36777,2252
Point	Reimenrod	village	141	37142,1402
Point	Waldsolms	town	5151	37145,1379
Point	Udenhausen	village	361	37164,4466
Point	Eichen	suburb	2000	37214,12
Point	Oberlemp	suburb	447	37387,6123
Point	Bad Endbach	town	8649	37441,9512
Point	Weidenau	village	360	37470,5712
Point	Poppenrod	village	201	37548,1014
Point	Wernborn	village	1500	37608,8328
Point	Willofs	village	389	37712,949
Point	Rosbach vor der Höhe	town	12100	37774,1018
Point	Pfaffenrod	village	120	37961,2109
Point	Hosenfeld	village	4745	38021,9472
Point	Wohratal	town	2514	38036,3283
Point	Gersrod	hamlet	72	38068,5173
Point	Eulersdorf	village	162	38117,8746
Point	Berghausen	suburb	1047	38124,4615
Point	Brungershausen	hamlet	103	38173,6172
Point	Otrau	village	2463	38287,1728
Point	Reinhards	village	150	38287,1972
Point	Bieben	village	215	38321,9142
Point	Wetter	town	9355	38325,3651
Point	Schletzenhausen	village	402	38385,8675
Point	Wincherode	village	65	38423,6417
Point	Werdorf	suburb	3142	38436,4033
Point	Müs	village	1105	38463,9235
Point	Günterod	village	1000	38472,8078
Point	Bischoffen	town	3490	38477,3968

Point	Bad Salzschlirf	town	3180	38781,5758
Point	Brandlos	village	150	38902,0032
Point	Neukirchen	village	600	39031,4084
Point	Brachtal	village	5354	39146,3639
Point	Hauswurz	village	948	39252,6972
Point	Nausis	village	308	39750,3016
Point	Eichenau	village	158	39772,9697
Point	Grebenau	village	834	39780,1856
Point	Dautphetal	municipality	12048	39967,3103
Point	Bruchmühle	hamlet	5	39980,2481
Point	Schwalmstadt	town	19283	39982,6606
Point	Hammersbach	village	4749	40225,6056
Point	Amönau	village	868	40245,744
Point	Riebelsdorf	village	777	40325,3169
Point	Uffhausen	village	450	40382,3111
Point	Kauppen	village	83	40457,2707
Point	Gilsberg	town	3503	40597,4307
Point	Dreisbach	village	298	40682,2445
Point	Rückershausen	village	181	40694,1486
Point	Bottenhorn	village	1400	40721,1067
Point	Nidderau	town	20095	40822,5077
Point	Hintersteinau	suburb	750	40832,6193
Point	Braunfels	town	11322	40832,9753
Point	Ehringshausen	town	9459	40908,1559
Point	Neukirchen	suburb	4294	40953,7313
Point	Ronneburg	village	3226	40954,1228
Point	Neukirchen (Knüll)	town	7262	40954,9248
Point	Ostheim	suburb	2967	40978,4881
Point	Großenlüder	town	3642	41141,5486
Point	Wallersdorf	village	272	41148,9117
Point	Windecken	suburb	2967	41421,3113
Point	Usingen	town	13346	41494,9838
Point	Siegbach	town	2926	41698,3652
Point	Asterode	village	500	41918,2865
Point	Buchenrod	village	320	42006,4061
Point	Gehau	hamlet	182	42112,5587
Point	Mittenaar	town	5028	42164,7857
Point	Karben	town	21627	42210,3735
Point	Moisheid	village	356	42235,4767
Point	Grävenwiesbach	town	5257	42314,213
Point	Steinperf	village	880	42340,3505
Point	Wehrheim	village	9300	42458,7197
Point	Hatterode	village	340	42508,4471
Point	Gründau	town	14779	42749,319
Point	Gemünden	town	4102	42878,4569
Point	Leun	town	6010	42943,8054
Point	Rosenthal	town	2250	43007,0738
Point	Friedrichsdorf	town	24525	43305,2921
Point	Schlitz	town	10272	43306,6695
Point	Katzenfurt	village	1982	43387,4454
Point	Kressenbach	village	350	43542,6803
Point	Wallroth	suburb	1004	43799,4943
Point	Schöneck	town	11535	43829,181
Point	Neuberg	village	5226	43882,7537
Point	Breitenbach am Herz-	village	1922	44081,161
Point	Christerode	village	283	44081,3415
Point	Münchhausen	village	3500	44267,6325
Point	Wächtersbach	town	12446	44350,0963
Point	Angelburg	town	3555	44354,1071
Point	Giesel	suburb	1062	44381,0776
Point	Hutzdorf	village	898	44571,2553
Point	Hergertsmühle	hamlet	5	44584,0787
Point	Tiefenbach	suburb	1116	44863,4544
Point	Lütterz	village	80	44972,1698
Point	Sinn	town	3599	45084,0106
Point	Greifenthal	village	318	45135,1666
Point	Schönstein	hamlet	146	45161,3678
Point	Happelsmühle	hamlet	5	45191,8924
Point	Bad Soden-Salmünster	town	13927	45222,8625
Point	Weilmünster	town	9427	45231,3977
Point	Seigertshausen	village	785	45398,4858
Point	Edingen	village	1182	45458,8487
Point	Niederdorfelden	town	2959	45556,1118
Point	Obernhain	village	1600	45604,7439
Point	Densberg	village	500	45621,8352
Point	Pfordt	village	395	45733,3367
Point	Queck	village	764	45778,6109
Point	Bruchköbel	town	20574	45818,4084
Point	Hauptschwenda	village	166	45891,2978
Point	Oberjossa	village	270	46048,1786
Point	Gelnhausen	town	21853	46187,4559
Point	Langenselbold	town	12944	46325,7922
Point	Neu-Anspach	town	14658	46368,7614
Point	Rimbach	village	415	46408,1195
Point	Fleisbach	village	1855	46451,8617

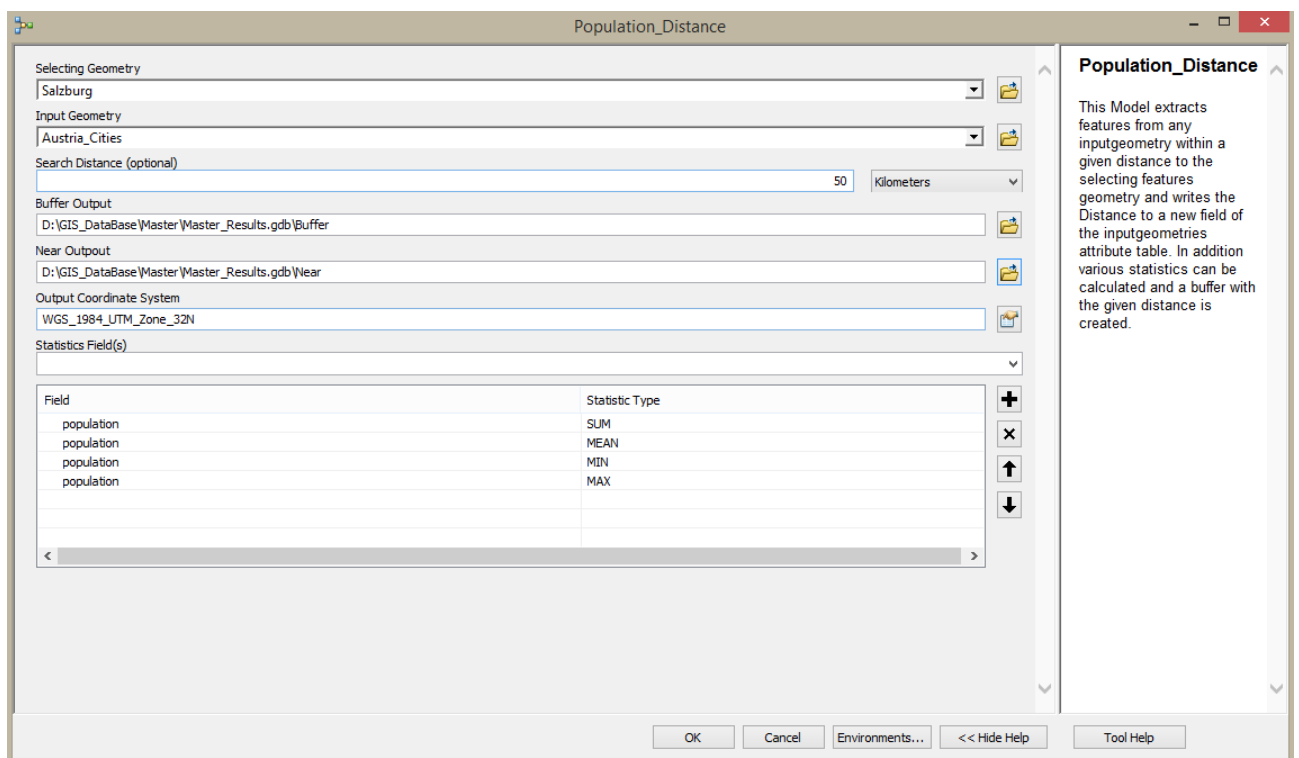
Point	Sandlofs	village	334	46503,9252
Point	Oberaula	village	3439	46557,0952
Point	Flieden	village	8775	46607,873
Point	Ulm	village	716	46839,5743
Point	Jesberg	village	2668	46875,8911
Point	Fraurombach	village	250	46917,3016
Point	Gomfritz	hamlet	195	46972,6117
Point	Ober-Wegfurth	hamlet	61	47046,0579
Point	Dodenhausen	village	413	47132,163
Point	Bad Homburg vor der	town	52000	47248,5089
Point	Steinau an der Straße	town	11160	47262,6555
Point	Biedenkopf	town	13766	47289,4583
Point	Unter-Wegfurth	hamlet	110	47336,9068
Point	Merkenbach	village	1598	47416,0482
Point	Allendorf	village	1309	47489,0104
Point	Unter-Schwarz	village	287	47495,4792
Point	Herborn	town	21088	47581,3399
Point	Niederjossa	village	760	47650,6558
Point	Erlensee	town	13141	47719,7445
Point	Greifenstein	town	7382	47826,383
Point	Allendorf	hamlet	151	47921,3404
Point	Holzhausen	village	880	47921,8645
Point	Weilrod	town	6527	48085,611
Point	Bad Vilbel	town	30927	48268,402
Point	Niederzell	village	1357	48286,1851
Point	Schlüchtern	town	17276	48341,7133
Point	Breidenbach	town	6952	48364,4897
Point	Hattenbach	village	610	48540,1012
Point	Neuenhaßlau	village	3202	48656,4766
Point	Neuhof	town	11327	48832,7839
Point	Willersdorf	village	627	48879,6008
Point	Neuental	town	3369	48903,7744
Point	Bad Orb	town	10063	48946,3339
Point	Solms	hamlet	102	48972,9801
Point	Hasselroth	town	7379	49049,2818
Point	Hirschberg	village	292	49206,2715
Point	Schwarzenborn	village	1170	49423,2746
Point	Elm	village	1346	49549,9795
Point	Hörbach	village	1369	49552,1557
Point	Löhnberg	town	4463	49634,4707
Point	Harmerz	village	1078	49639,0722
Point	Frielendorf	village	8152	49643,1681
Point	Burgwald	village	5140	49713,1996
Point	Gondsroth	village	1180	49767,2287
Point	Linsengericht	town	9786	49840,5064

7.2 Tutorial Online Bereitstellung eines Geoverarbeitungsservices mit ArcGIS for Server

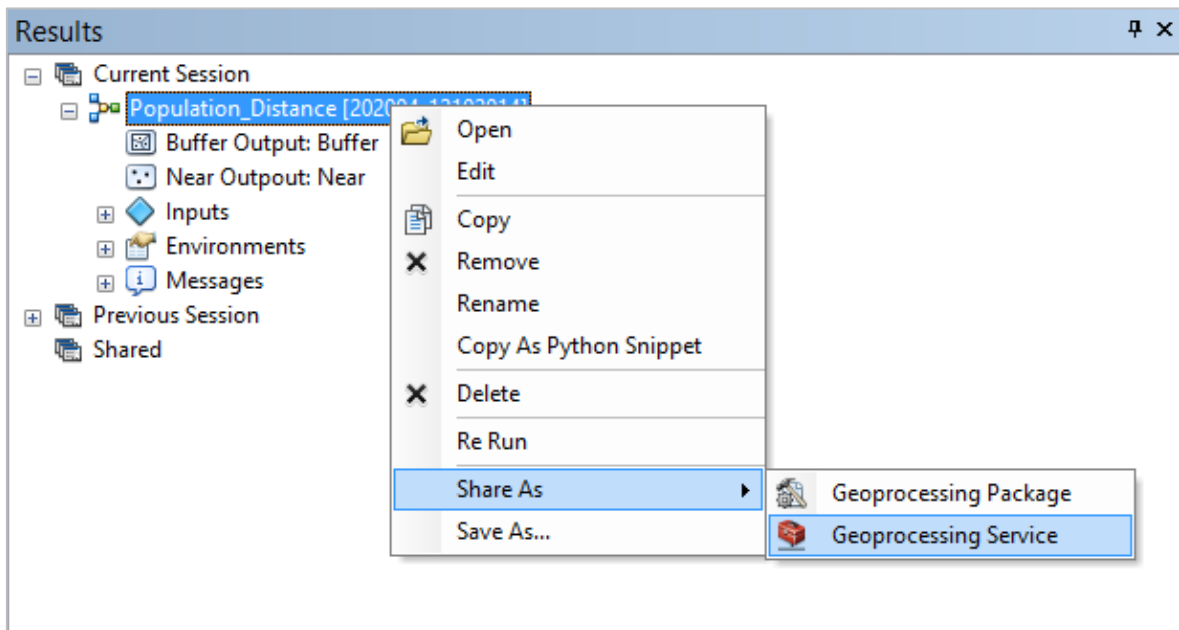
- Model aus der lokalen Toolbox auswählen



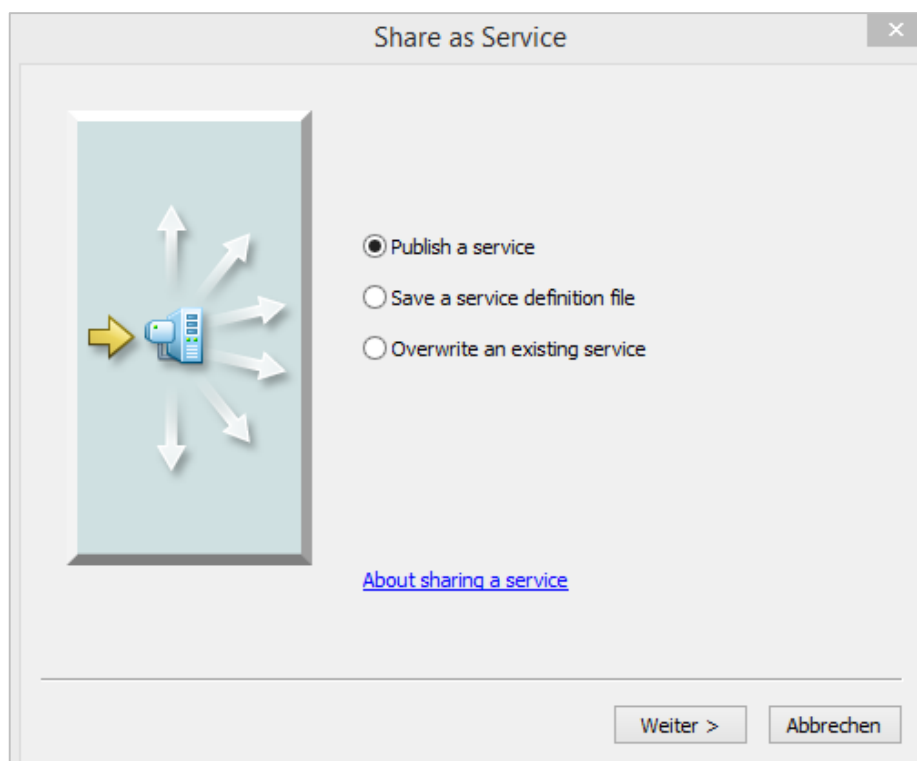
- Model mit den gewünschten Parametern ausführen



- Im Results Window



- Publish a service



- Server und Service Name festlegen

Publish a Service

Choose a connection

arcgis on zgis209.geo.sbg.ac.at (publisher)

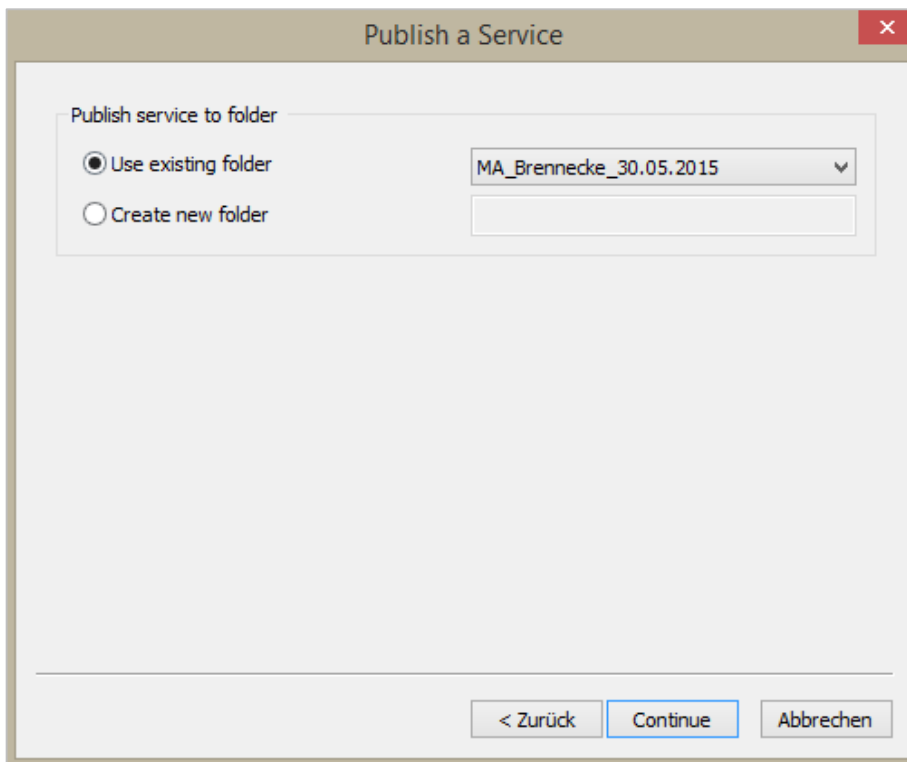
Server type: ArcGIS Server

Service name

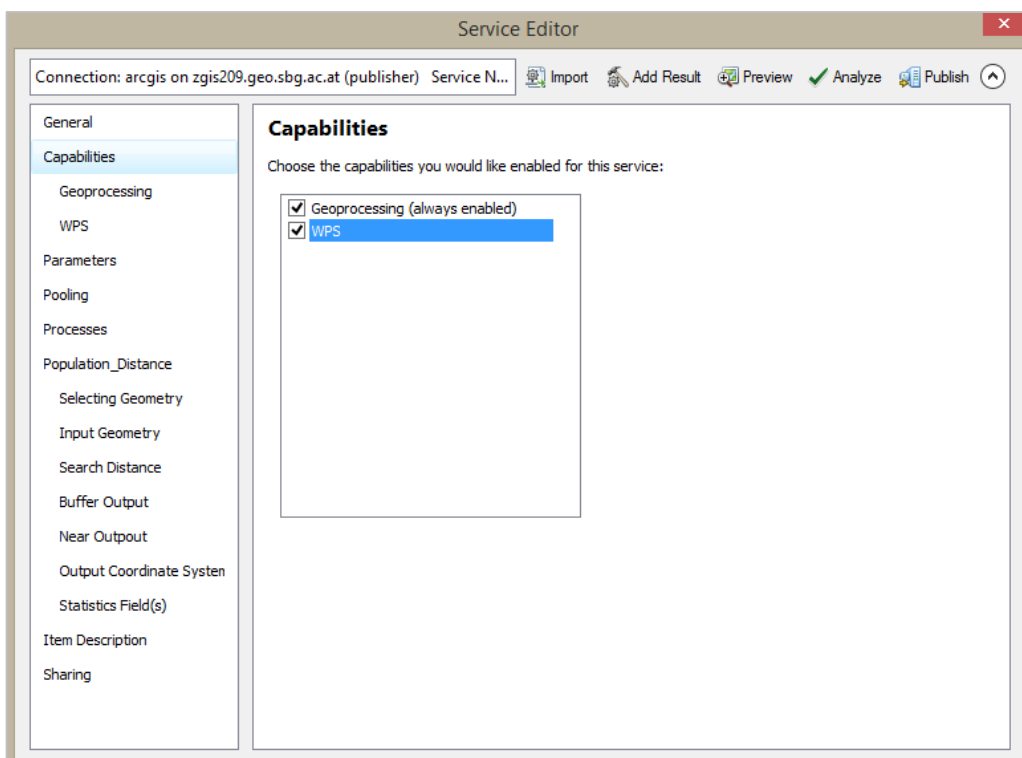
PopulationDistance

< Zurück Weiter > Abbrechen

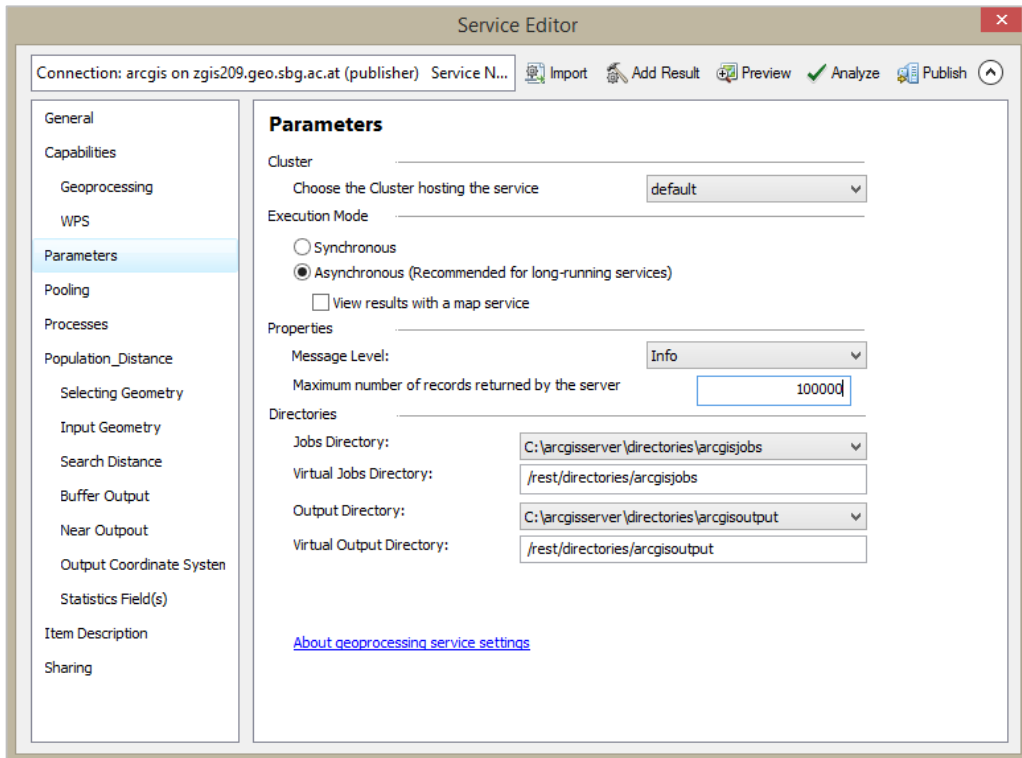
- Ordner auf Server festlegen



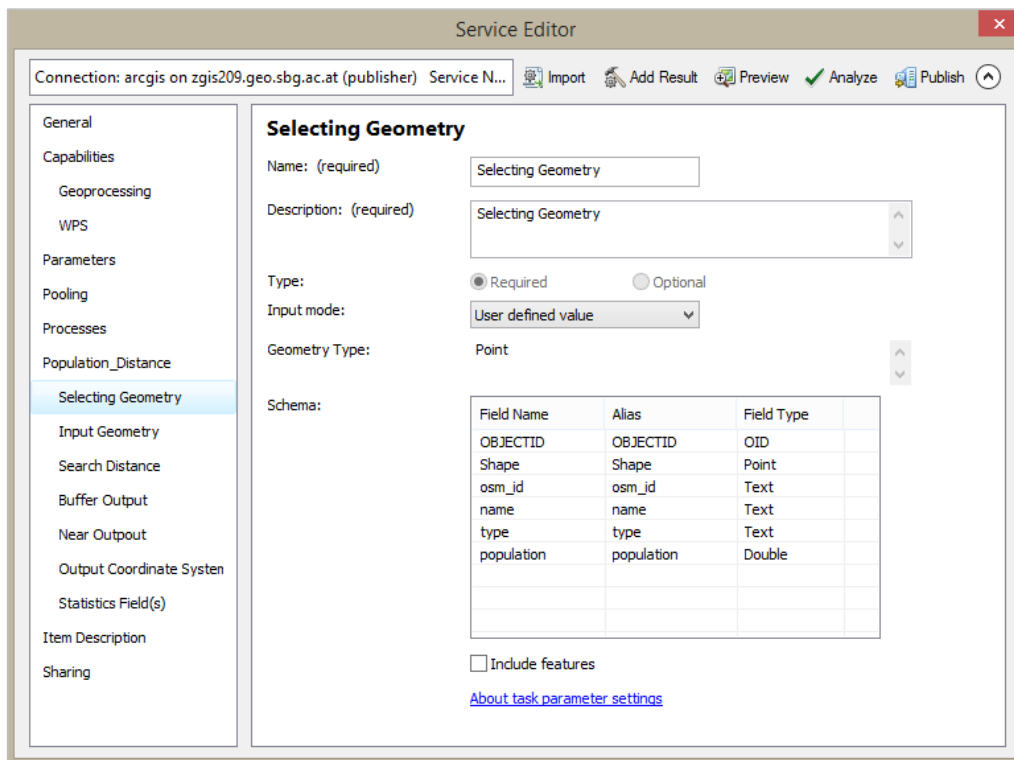
- Im Service Editor zuerst die WPS Capability aktivieren



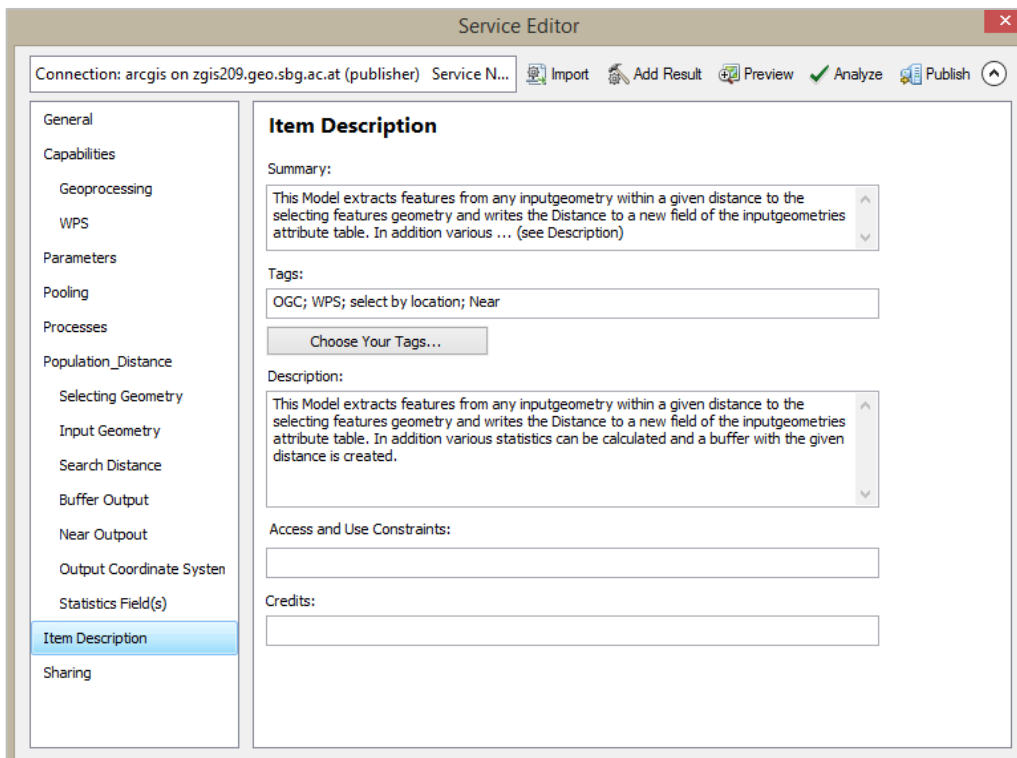
- Unter Parameter das Message Level auf Info einstellen um bei möglichen Fehlern genaue Informationen zu erhalten
- Maximum Number of records returned by the server sind bei diesem Model auf 100.000 erhöht. Der Default Wert beträgt 1.000. Bei der Auswahl der Städte kann es je nach Radius zu mehr als 10.000 kommen



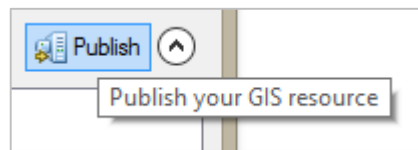
- Sowohl bei der Selecting Geometry als auch bei der Input Geometry den Input Mode auf User defined value stellen
- Wichtig ist bei allen Model Parametern, dass das Description Feld ausgefüllt ist



- Unter Item Description die Felder Summary, Tags und Description ausfüllen



- Geoprocessing Service veröffentlichen



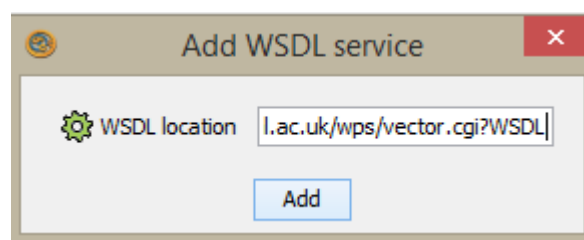
7.3 Tutorial Ausführung eines PyWPS in Taverna

Ausführung des Voronoi PyWPS

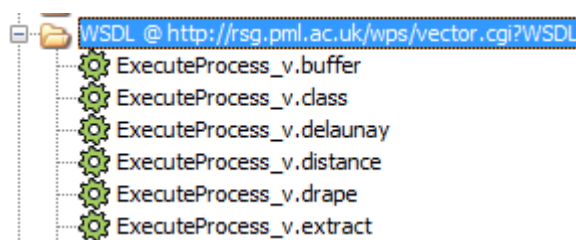
→ Service Panel



- Import New Service
- WSDL Service
- WSDL location: <https://rsg.pml.ac.uk/wps/vector.cgi?WSDL>
- Add

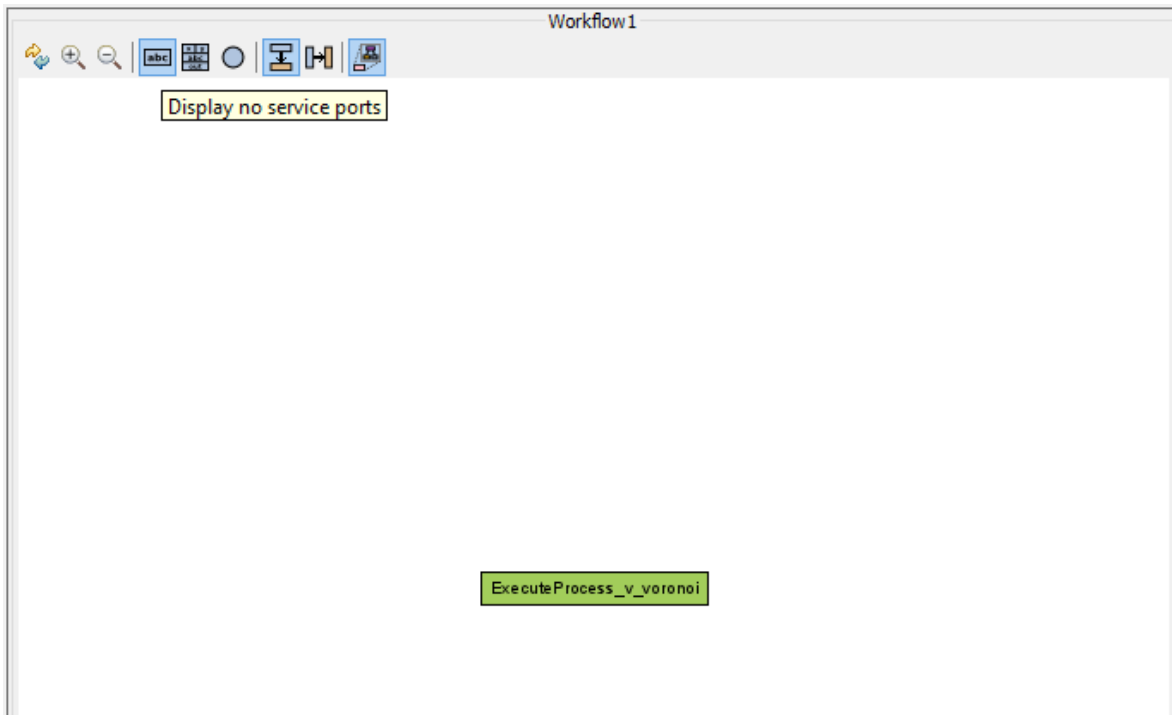


→ Gewünschten Prozess per Drag&Drop in den Workflow Bereich ziehen

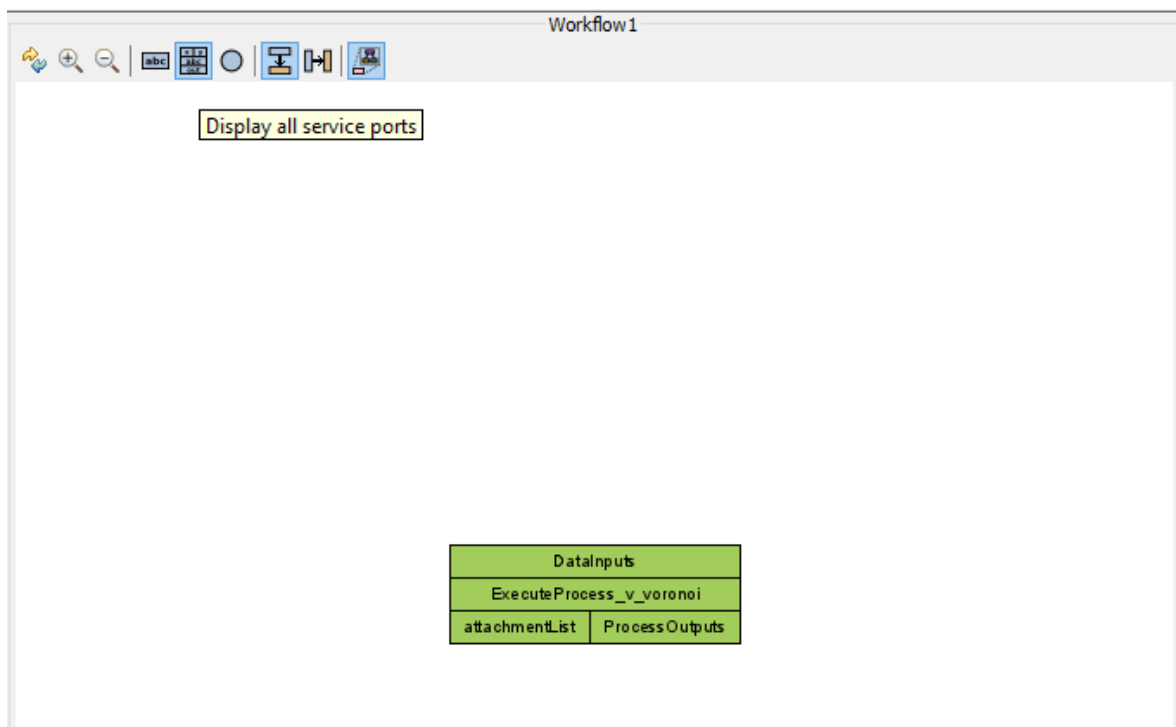


→ Workflow Bereich

- ExecuteProcess_v.voronoj
- Wechsel von Display no service ports...

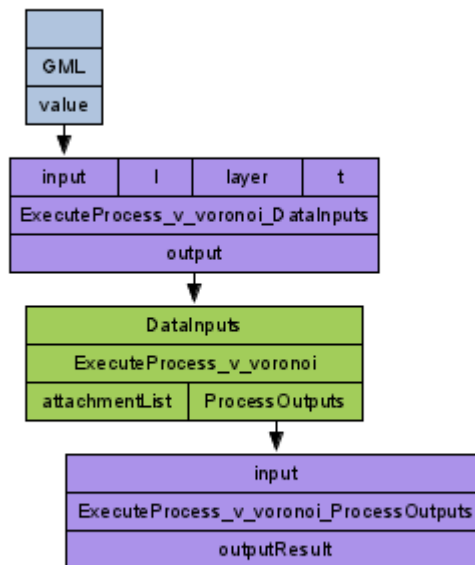


→ ...zu Display all service ports

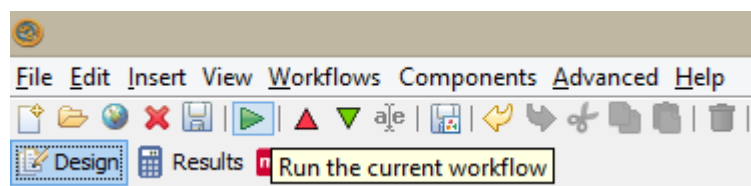


- Rechtsklick ExecuteProcess_v.voronoj
- Add XML Input Splitter...
 - DataInputs

- ➔ Rechtsklick auf Input des XML Input Splitters
 - Constant value
 - Testdaten einfügen (wichtig http statt https)
 - http://rsg.pml.ac.uk/wps/testdata/random_points.gml
- ➔ Rechtsklick ExecuteProcess_v.voronoi
 - Add XML Output Splitter...
 - ProcessOutputs



- ➔ Taverna Werkzeugzeiste
 - Run the current workflow



- ➔ Nachdem der Workflow erfolgreich abgelaufen ist Klick auf ProcessOutputs
 - Invocation
 - Save value
 - Voronoi.gml

Workflow runs Delete all Delete

Click on a run to see its values
Click on a service in the diagram
to see intermediate values (if available)

Workflow 1 2015-01-16 23:53:58
Workflow 1 2015-01-16 23:52:23

Graph Progress report

```

graph TD
    GML[GML] --> ExecuteProcess_v_voronoi_DataInputs[ExecuteProcess_v_voronoi_DataInputs]
    ExecuteProcess_v_voronoi_DataInputs --> ExecuteProcess_v_voronoi[ExecuteProcess_v_voronoi]
    ExecuteProcess_v_voronoi --> ExecuteProcess_v_voronoi_ProcessOutputs[ExecuteProcess_v_voronoi_ProcessOutputs]
  
```

Finished Pause Cancel Edit executed workflow Refresh intermediate values Show workflow results

Intermediate values for the service **ExecuteProcess_v_voronoi_ProcessOutputs** Invocation started 2015-01-16 23:54:08, ended 2015-01-16 23:54:08 (47 ms) Save iteration values

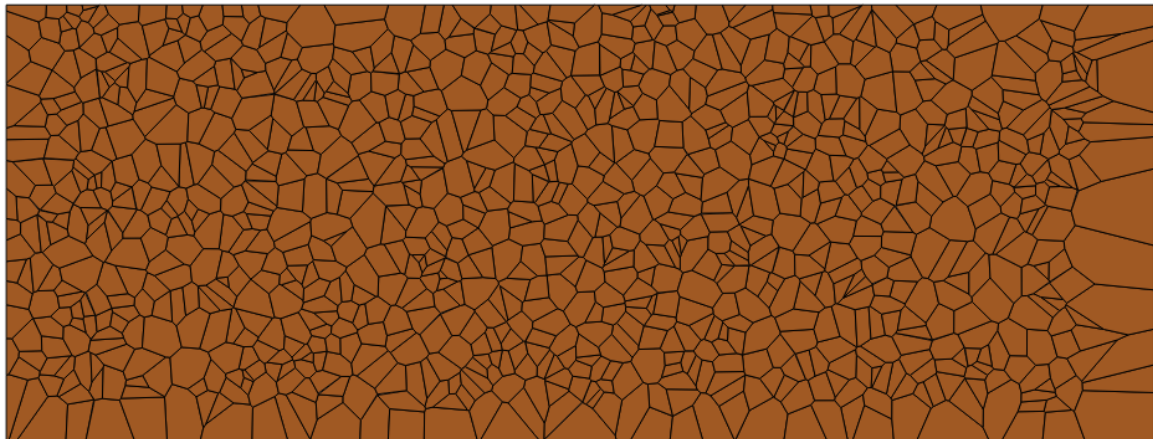
View all input

Value type: Text Refresh Wrap text Save value

```

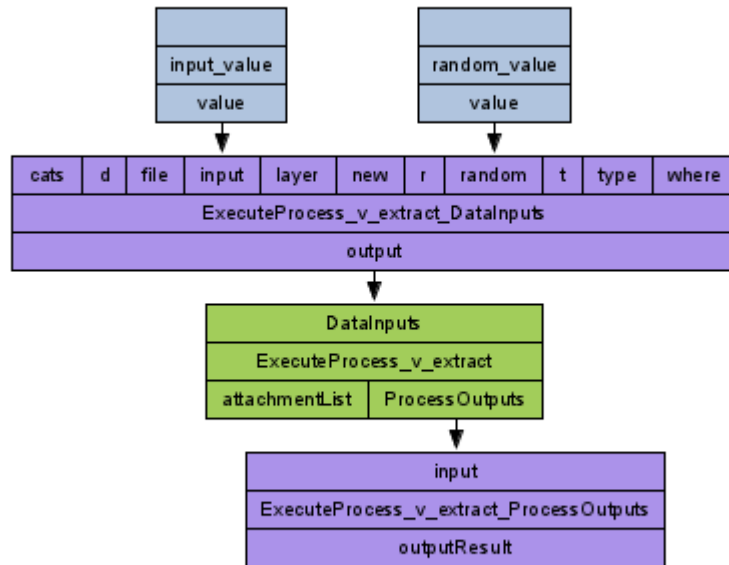
<ExecuteProcess_v_voronoiResponse><outputResult><ogr:FeatureCollection xmlns:gml="http://www.opengis.net/gml" xmlns:ogr="http://ogr.maptools.org/" xmlns:ows="http://www.opengis.net/ows" xmlns:boundedBy="http://www.opengis.net/gml:boundedBy">
  <gml:Box>
    <gml:coord><gml:X>120583.58846725</gml:X><gml:Y>-5259.56415397</gml:Y><gml:Z>0</gml:Z></gml:coord>
    <gml:coord><gml:X>975343.378355999</gml:X><gml:Y>319919.98907641</gml:Y><gml:Z>0</gml:Z></gml:coord>
  </gml:Box>
</ogr:FeatureCollection>
</outputResult>
</ExecuteProcess_v_voronoiResponse>
  
```

➔ Visualisierung der Voronoi GML Datei in QGIS

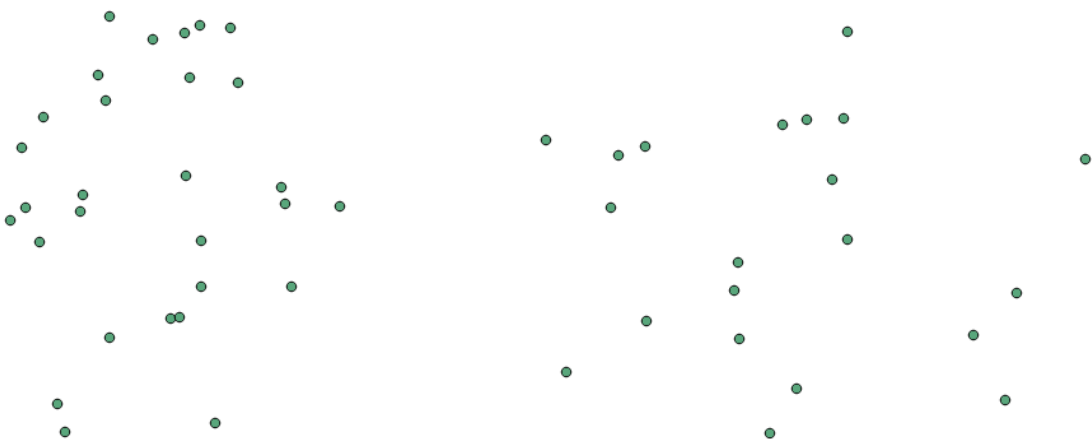


➔ Ein weiterer PyWPS Prozess

- v.extract



- ➔ Der Input ist wieder die random_points
 - http://rsg.pml.ac.uk/wps/testdata/random_points.gml
- ➔ Zusätzlich zum Input ist noch der Parameter random mit dem Wert 50 definiert
 - Es werden 50 zufällige Punkte extrahiert
 - Anmerkung: Leider konnte keine korrekte where clause definiert werden, die die gewünschten Features zurückgibt.
- ➔ Visualisierung der 50 Punkte in QGIS



7.4 WPS Requests des ‚Population Distance‘ Modells

7.4.1 Get Capabilities Request

GetCapabilities Request via HTTP Get:

http://zgis209.geo.sbg.ac.at/arcgis/services/MA_Brennecke_30.05.2015/PopulationDistance/GPServer/WPSServer?SERVICE=WPS&ACCEPTVERSIONS=1.0.0&REQUEST=GetCapabilities

XML Antwort:

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:Capabilities version="1.0.0" service="WPS" xml:lang="en-US" updateSequence="1" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsGetCapabilities_response.xsd" xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xlink="http://www.w3.org/1999/xlink">
  <ows:ServiceIdentification>
    <ows:Abstract>This Model extracts features from any inputgeometry within a given distance to the selecting features geometry and writes the Distance to a new field of the inputgeometries attribute table. In addition various statistics can be calculated and a buffer with the given distance is created.</ows:Abstract>
    <ows:Keywords>
      <ows:Keyword>OGC; WPS; select by location; Near</ows:Keyword>
    </ows:Keywords>
    <ows:ServiceType>WPS</ows:ServiceType>
    <ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
    <ows:Fees/>
    <ows:AccessConstraints>none</ows:AccessConstraints>
  </ows:ServiceIdentification>
  <ows:ServiceProvider>
    <ows:ProviderName/>
    <ows:ServiceContact>
      <ows:IndividualName>C. Brennecke</ows:IndividualName>
      <ows:PositionName/>
      <ows:ContactInfo>
        <ows:Phone/>
        <ows:Address/>
        <ows:OnlineResource
xlink:href="http://zgis209.geo.sbg.ac.at/arcgis/services/MA_Brennecke_30.05.2015/PopulationDistance/GPServer/WPSServer"/>
      </ows:OnlineResource>
      <ows:HoursOfService/>
      <ows:ContactInstructions/>
    </ows:ContactInfo>
  </ows:ServiceContact>
</ows:ServiceProvider>
  <ows:OperationsMetadata>
    <ows:Operation name="GetCapabilities">
      <ows:DCP>
        <ows:HTTP>
          <ows:Get
xlink:href="http://zgis209.geo.sbg.ac.at/arcgis/services/MA_Brennecke_30.05.2015/PopulationDistance/GPServer/WPSServer?"/>
        </ows:Get>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
</ows:OperationsMetadata>
</wps:Capabilities>
```

```

        <ows:Post
xlink:href="http://zgis209.geo.sbg.ac.at/arcgis/ser-
vices/MA_Brennecke_30.05.2015/PopulationDistance/GPServer/WPSServer"/>
        </ows:HTTP>
    </ows:DCP>
</ows:Operation>
<ows:Operation name="DescribeProcess">
    <ows:DCP>
        <ows:HTTP>
            <ows:Get
xlink:href="http://zgis209.geo.sbg.ac.at/arcgis/ser-
vices/MA_Brennecke_30.05.2015/PopulationDistance/GPServer/WPSServer?"/>
            <ows:Post
xlink:href="http://zgis209.geo.sbg.ac.at/arcgis/ser-
vices/MA_Brennecke_30.05.2015/PopulationDistance/GPServer/WPSServer"/>
            </ows:HTTP>
        </ows:DCP>
    </ows:Operation>
<ows:Operation name="Execute">
    <ows:DCP>
        <ows:HTTP>
            <ows:Get
xlink:href="http://zgis209.geo.sbg.ac.at/arcgis/ser-
vices/MA_Brennecke_30.05.2015/PopulationDistance/GPServer/WPSServer?"/>
            <ows:Post
xlink:href="http://zgis209.geo.sbg.ac.at/arcgis/ser-
vices/MA_Brennecke_30.05.2015/PopulationDistance/GPServer/WPSServer"/>
            </ows:HTTP>
        </ows:DCP>
    </ows:Operation>
</ows:OperationsMetadata>
<wps:ProcessOfferings>
    <wps:Process wps:processVersion="1">
        <ows:Identifier>PopulationDistance</ows:Identifier>
        <ows:Title>Population_Distance</ows:Title>
        <ows:Abstract/>
    </wps:Process>
</wps:ProcessOfferings>
<wps:Languages>
    <wps:Default>
        <ows:Language>en-US</ows:Language>
    </wps:Default>
    <wps:Supported>
        <ows:Language>en-US</ows:Language>
    </wps:Supported>
</wps:Languages>
</wps:Capabilities>

```

7.4.2 Describe Process Request

Describe Process Request via HTTP Get:

http://zgis209.geo.sbg.ac.at/arcgis/services/MA_Brennecke_30.05.2015/PopulationDistance/GPServer/WPSServer?SERVICE=WPS&VERSION=1.0.0&REQUEST=DescribeProcess&Identifier=PopulationDistance

XML Antwort:

```

<?xml version="1.0" encoding="UTF-8"?>
<wps:ProcessDescriptions version="1.0.0" service="WPS" xml:lang="en-US"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.open-
gis.net/wps/1.0.0/wpsDescribeProcess_response.xsd"
xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.open-
gis.net/ows/1.1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xlink="http://www.w3.org/1999/xlink">
  <ProcessDescription wps:processVersion="1.0.0" storeSupported="true"
statusSupported="false">
    <ows:Identifier>PopulationDistance</ows:Identifier>
    <ows:Title>Population_Distance</ows:Title>
    <ows:Abstract/>
    <DataInputs>
      <Input maxOccurs="1" minOccurs="1">
        <ows:Identifier>Selecting_Geometry</ows:Identifier>
        <ows:Title>Selecting Geometry</ows:Title>
        <ComplexData>
          <Default>
            <Format>
              <MimeType>text/xml</MimeType>
              <Encoding>UTF-8</Encoding>
              <Schema>http://schemas.open-
gis.net/gml/3.1.1/base/gml.xsd</Schema>
            </Format>
          </Default>
          <Supported>
            <Format>
              <MimeType>text/xml</MimeType>
              <Encoding>UTF-8</Encoding>
              <Schema>http://schemas.open-
gis.net/gml/3.1.1/base/gml.xsd</Schema>
            </Format>
          </Supported>
        </ComplexData>
      </Input>
      <Input maxOccurs="1" minOccurs="1">
        <ows:Identifier>Input_Geometry</ows:Identifier>
        <ows:Title>Input Geometry</ows:Title>
        <ComplexData>
          <Default>
            <Format>
              <MimeType>text/xml</MimeType>
              <Encoding>UTF-8</Encoding>
              <Schema>http://schemas.open-
gis.net/gml/3.1.1/base/gml.xsd</Schema>
            </Format>
          </Default>
          <Supported>
            <Format>
              <MimeType>text/xml</MimeType>
              <Encoding>UTF-8</Encoding>
              <Schema>http://schemas.open-
gis.net/gml/3.1.1/base/gml.xsd</Schema>
            </Format>
          </Supported>
        </ComplexData>
      </Input>
      <Input maxOccurs="1" minOccurs="0">
        <ows:Identifier>Search_Distance</ows:Identifier>
        <ows:Title>Search Distance</ows:Title>
        <LiteralData>
          <ows:DataType ows:refer-
ence="http://www.w3.org/TR/xmlschema-2/#double">double</ows:DataType>

```

```

        <UOMs>
          <Default>
            <ows:UOM>kilometers</ows:UOM>
          </Default>
          <Supported>
            <ows:UOM>unknown</ows:UOM>
            <ows:UOM>inches</ows:UOM>
            <ows:UOM>points</ows:UOM>
            <ows:UOM>feet</ows:UOM>
            <ows:UOM>yards</ows:UOM>
            <ows:UOM>miles</ows:UOM>
            <ows:UOM>nauticalmiles</ows:UOM>
            <ows:UOM>millimeters</ows:UOM>
            <ows:UOM>centimeters</ows:UOM>
            <ows:UOM>meters</ows:UOM>
            <ows:UOM>kilometers</ows:UOM>
            <ows:UOM>decimaldegrees</ows:UOM>
            <ows:UOM>decimeters</ows:UOM>
          </Supported>
        </UOMs>
        <ows:AnyValue/>
        <DefaultValue>50</DefaultValue>
      </LiteralData>
    </Input>
    <Input maxOccurs="1" minOccurs="1">
      <ows:Identifier>Output_Coordinate_System</ows:Identifier>
      <ows:Title>Output Coordinate System</ows:Title>
      <LiteralData>
        <ows:DataType ows:reference="http://www.w3.org/TR/xmlschema-2/#string">string</ows:DataType>
        <ows:AnyValue/>
        <DefaultValue>PROJCS['WGS_1984_UTM_Zone_32N',GEOGCS['GCS_WGS_1984',DATUM['D_WGS_1984',SPHEROID['WGS_1984',6378137.0,298.257223563]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['Transverse_Mercator'],PARAMETER['False_Easting',500000.0],PARAMETER['False_Northing',0.0],PARAMETER['Central_Meridian',9.0],PARAMETER['Scale_Factor',0.9996],PARAMETER['Latitude_Of_Origin',0.0],UNIT['Meter',1.0]]</DefaultValue>
      </LiteralData>
    </Input>
    <Input maxOccurs="1" minOccurs="0">
      <ows:Identifier>output_spatial_reference</ows:Identifier>
      <ows:Title>output_spatial_reference</ows:Title>
      <LiteralData>
        <ows:DataType ows:reference="http://www.w3.org/TR/xmlschema-2/#">string</ows:DataType>
        <ows:AnyValue/>
        <DefaultValue/>
      </LiteralData>
    </Input>
  </DataInputs>
  <ProcessOutputs>
    <Output>
      <ows:Identifier>Buffer_Output</ows:Identifier>
      <ows:Title>Buffer Output</ows:Title>
      <ComplexOutput>
        <Default>
          <Format>
            <MimeType>text/xml</MimeType>
            <Encoding>UTF-8</Encoding>
            <Schema>http://schemas.opengis.net/gml/3.1.1/base/gml.xsd</Schema>
          </Format>
        </Default>
      </ComplexOutput>
    </Output>
  </ProcessOutputs>

```

```

        </Format>
    </Default>
</Supported>
    <Format>
        <MimeType>text/xml</MimeType>
        <Encoding>UTF-8</Encoding>
        <Schema>http://schemas.open-
gis.net/gml/3.1.1/base/gml.xsd</Schema>
    </Format>
</Supported>
</ComplexOutput>
</Output>
<Output>
    <ows:Identifier>Near_Outpout</ows:Identifier>
    <ows:Title>Near Outpout</ows:Title>
    <ComplexOutput>
        <Default>
            <Format>
                <MimeType>text/xml</MimeType>
                <Encoding>UTF-8</Encoding>
                <Schema>http://schemas.open-
gis.net/gml/3.1.1/base/gml.xsd</Schema>
            </Format>
        </Default>
    </Supported>
    <Format>
        <MimeType>text/xml</MimeType>
        <Encoding>UTF-8</Encoding>
        <Schema>http://schemas.open-
gis.net/gml/3.1.1/base/gml.xsd</Schema>
    </Format>
</Supported>
    <Format>
        <MimeType>text/xml</MimeType>
        <Encoding>UTF-8</Encoding>
        <Schema>http://schemas.open-
gis.net/gml/3.1.1/base/gml.xsd</Schema>
    </Format>
</Supported>
</ComplexOutput>
</Output>
</ProcessOutputs>
</ProcessDescription>
</wps:ProcessDescriptions>

```

7.4.3 Antwort des Execute Requests

XML Antwort auf den WPS Execute Request:

```

<?xml version="1.0" encoding="utf-8" ?>
<wps:ExecuteResponse version='1.0.0' service='WPS' xml:lang='en-US' ser-
viceInstance='http://zgis209.geo.sbg.ac.at/arcgis/ser-
vices/MA_Brennecke_30.05.2015/PopulationDistance/GPSServer/WPSServer?re-
quest=getcapabilities&service=WPS&version=1.0.0' xsi:schemaLoca-
tion='http://www.opengis.net/wps/1.0.0 http://schemas.open-
gis.net/wps/1.0.0/wpsExecute_response.xsd'
  xmlns:wps='http://www.opengis.net/wps/1.0.0'
  xmlns:ows='http://www.opengis.net/ows/1.1'
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns:xlink='http://www.w3.org/1999/xlink'>
  <wps:Process wps:processVersion='1'>
    <ows:Identifier>PopulationDistance</ows:Identifier>
    <ows:Title>Population_Distance</ows:Title>
    <ows:Abstract/>
  </wps:Process>
  <wps:Status creationTime='2015-02-24T00:08:39'>
    <wps:ProcessSucceeded/>

```



```

</wps:Status>
<wps:ProcessOutputs>
  <wps:Output>
    <ows:Identifier>Buffer_Output</ows:Identifier>
    <ows:Title>Buffer_Output</ows:Title>
    <wps:Reference mimeType='text/xml'
href='http://zgis209.geo.sbg.ac.at/arcgis/rest/directories/arcgisout-
put/MA_Brennecke_30.05.2015/PopulationDistance_GPS-
erver/ ags_j3654937d31b94c988cfel339ecb45855Buffer Output.xml' />
    </wps:Output>
  <wps:Output>
    <ows:Identifier>Near_Outpout</ows:Identifier>
    <ows:Title>Near_Outpout</ows:Title>
    <wps:Reference mimeType='text/xml'
href='http://zgis209.geo.sbg.ac.at/arcgis/rest/directories/arcgisout-
put/MA_Brennecke_30.05.2015/PopulationDistance_GPS-
erver/ ags_j3654937d31b94c988cfel339ecb45855Near Outpout.xml' />
    </wps:Output>
  </wps:ProcessOutputs>
</wps:ExecuteResponse>

```

7.5 Messprotokolle der ausgeführten Modelle

Szenario 1: ArcGIS interne Zeitmessung

```

Executing: PopulationDistance Grünberg Hessen "50 Kilometers" D:\GIS_Data-
Base\Master\Master_Results.gdb\Buffer D:\GIS_DataBase\Master\Master_Re-
sults.gdb\Near PROJCS['WGS_1984_UTM_Zone_32N',GEOGCS['GCS_WGS_1984',DA-
TUM['D_WGS_1984',SPHEROID['WGS_1984',6378137.0,298.257223563]],PRIMEM['Green-
wich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['Transverse_Merca-
tor'],PARAMETER['False_Easting',500000.0],PARAMETER['False_Northing',0.0],PA-
RAMETER['Central_Meridian',9.0],PARAMETER['Scale_Factor',0.9996],PARAME-
TER['Latitude_Of_Origin',0.0],UNIT['Meter',1.0]] "population SUM;population
MEAN;population MIN;population MAX"
Start Time: Wed Feb 18 17:14:15 2015
Executing (Buffer): Buffer Grünberg D:\GIS_DataBase\Master\Master_Re-
sults.gdb\Buffer "50 Kilometers" FULL ROUND ALL #
Start Time: Wed Feb 18 17:14:16 2015
Succeeded at Wed Feb 18 17:14:16 2015 (Elapsed Time: 0,69 seconds)
Executing (Project): Project Hessen D:\GIS_DataBase\Master\Master_Re-
sults.gdb\Input_Proj PROJCS['WGS_1984_UTM_Zone_32N',GEOGCS['GCS_WGS_1984',DA-
TUM['D_WGS_1984',SPHEROID['WGS_1984',6378137.0,298.257223563]],PRIMEM['Green-
wich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['Transverse_Merca-
tor'],PARAMETER['False_Easting',500000.0],PARAMETER['False_Northing',0.0],PA-
RAMETER['Central_Meridian',9.0],PARAMETER['Scale_Factor',0.9996],PARAME-
TER['Latitude_Of_Origin',0.0],UNIT['Meter',1.0]] # GEOGCS['GCS_WGS_1984',DA-
TUM['D_WGS_1984',SPHEROID['WGS_1984',6378137.0,298.257223563]],PRIMEM['Green-
wich',0.0],UNIT['Degree',0.0174532925199433]]
Start Time: Wed Feb 18 17:14:16 2015
Succeeded at Wed Feb 18 17:14:17 2015 (Elapsed Time: 0,46 seconds)
Executing (Make Feature Layer): MakeFeatureLayer D:\GIS_DataBase\Master\Mas-
ter_Results.gdb\Input_Proj Input_Proj_Layer # # "OBJECTID OBJECTID VISIBLE
NONE;Shape Shape VISIBLE NONE;osm_id osm_id VISIBLE NONE;name name VISIBLE
NONE;type type VISIBLE NONE;population population VISIBLE NONE"
Start Time: Wed Feb 18 17:14:17 2015
Succeeded at Wed Feb 18 17:14:17 2015 (Elapsed Time: 0,07 seconds)
Executing (Select Layer By Location): SelectLayerByLocation Input_Proj_Layer
WITHIN_A_DISTANCE Grünberg "50 Kilometers" NEW_SELECTION
Start Time: Wed Feb 18 17:14:17 2015
Succeeded at Wed Feb 18 17:14:17 2015 (Elapsed Time: 0,03 seconds)
Executing (Copy Features): CopyFeatures Input_Proj_Layer D:\GIS_DataBase\Mas-
ter\Master_Results.gdb\Near # 0 0 0
Start Time: Wed Feb 18 17:14:17 2015
Succeeded at Wed Feb 18 17:14:18 2015 (Elapsed Time: 0,62 seconds)

```

```

Executing (Near): Near D:\GIS_DataBase\Master\Master_Results.gdb\Near Grünberg
# NO_LOCATION NO_ANGLE
Start Time: Wed Feb 18 17:14:18 2015
Found 374 feature(s)
Succeeded at Wed Feb 18 17:14:18 2015 (Elapsed Time: 0,35 seconds)
Executing (Summary Statistics): Statistics D:\GIS_DataBase\Master\Master_Re-
sults.gdb\Near D:\GIS_DataBase\Master\Master_Results.gdb\Statistics "popula-
tion SUM;population MEAN;population MIN;population MAX" #
Start Time: Wed Feb 18 17:14:18 2015
Succeeded at Wed Feb 18 17:14:19 2015 (Elapsed Time: 0,35 seconds)
Executing (Join Field): JoinField D:\GIS_DataBase\Master\Master_Re-
sults.gdb\Buffer OBJECTID D:\GIS_DataBase\Master\Master_Results.gdb\Statistics
OBJECTID SUM_population;MEAN_population;MIN_population;MAX_population
Start Time: Wed Feb 18 17:14:19 2015
Succeeded at Wed Feb 18 17:14:19 2015 (Elapsed Time: 0,25 seconds)
Succeeded at Wed Feb 18 17:14:19 2015 (Elapsed Time: 3,68 seconds)

```

Szenario 2: ArcGIS interne Zeitmessung

```

Executing: PopulationDistance Grünberg Hessen "50 Kilometers" PRO-
JCS['WGS_1984_UTM_Zone_32N',GEOGCS['GCS_WGS_1984',DATUM['D_WGS_1984',SPHE-
ROID['WGS_1984',6378137.0,298.257223563]],PRIMEM['Greenwich',0.0],UNIT['De-
gree',0.0174532925199433]],PROJECTION['Transverse_Mercator'],PARAME-
TER['False_Easting',500000.0],PARAMETER['False_Northing',0.0],PARAMETER['Cen-
tral_Meridian',9.0],PARAMETER['Scale_Factor',0.9996],PARAMETER['Lati-
tude_Of_Origin',0.0],UNIT['Meter',1.0]]
Start Time: Tue Feb 24 16:10:47 2015
Executing (PopulationDistance): PopulationDistance "Feature Set" "Feature Set"
"50 Kilometers" C:\arcgisserver\directories\arcgis-
jobs\ma_brennecke_30.05.2015\populationdistance_gps-
erver\j920f8b7cba384b5d812ab11e37da45a1\scratch\scratch.gdb\Buffer C:\arcgis-
server\directories\arcgisjobs\ma_brennecke_30.05.2015\populationdistance_gps-
erver\j920f8b7cba384b5d812ab11e37da45a1\scratch\scratch.gdb\Near PRO-
JCS['WGS_1984_UTM_Zone_32N',GEOGCS['GCS_WGS_1984',DATUM['D_WGS_1984',SPHE-
ROID['WGS_1984',6378137.0,298.257223563]],PRIMEM['Greenwich',0.0],UNIT['De-
gree',0.0174532925199433]],PROJECTION['Transverse_Mercator'],PARAME-
TER['False_Easting',500000.0],PARAMETER['False_Northing',0.0],PARAMETER['Cen-
tral_Meridian',9.0],PARAMETER['Scale_Factor',0.9996],PARAMETER['Lati-
tude_Of_Origin',0.0],UNIT['Meter',1.0]]
Start Time: Tue Feb 24 16:10:56 2015
Executing (PopulationDistance): PopulationDistance "Feature Set" "Feature Set"
"50 Kilometers" C:\arcgisserver\directories\arcgis-
jobs\ma_brennecke_30.05.2015\populationdistance_gps-
erver\j920f8b7cba384b5d812ab11e37da45a1\scratch\scratch.gdb\Buffer C:\arcgis-
server\directories\arcgisjobs\ma_brennecke_30.05.2015\populationdistance_gps-
erver\j920f8b7cba384b5d812ab11e37da45a1\scratch\scratch.gdb\Near PRO-
JCS['WGS_1984_UTM_Zone_32N',GEOGCS['GCS_WGS_1984',DATUM['D_WGS_1984',SPHE-
ROID['WGS_1984',6378137.0,298.257223563]],PRIMEM['Greenwich',0.0],UNIT['De-
gree',0.0174532925199433]],PROJECTION['Transverse_Mercator'],PARAME-
TER['False_Easting',500000.0],PARAMETER['False_Northing',0.0],PARAMETER['Cen-
tral_Meridian',9.0],PARAMETER['Scale_Factor',0.9996],PARAMETER['Lati-
tude_Of_Origin',0.0],UNIT['Meter',1.0]] "population SUM;population MEAN;popula-
tion MIN;population MAX"
Start Time: Tue Feb 24 16:10:56 2015
Executing (Buffer): Buffer "Feature Set" C:\arcgisserver\directories\arcgis-
jobs\ma_brennecke_30.05.2015\populationdistance_gps-
erver\j920f8b7cba384b5d812ab11e37da45a1\scratch\scratch.gdb\Buffer "50 Kilome-
ters" FULL ROUND ALL #
Start Time: Tue Feb 24 16:10:59 2015
Succeeded at Tue Feb 24 16:11:01 2015 (Elapsed Time: 2.25 seconds)
Executing (Project): Project "Feature Set" C:\arcgisserver\directories\arcgis-
jobs\ma_brennecke_30.05.2015\populationdistance_gps-
erver\j920f8b7cba384b5d812ab11e37da45a1\scratch\scratch.gdb\Input_Proj PRO-
JCS['WGS_1984_UTM_Zone_32N',GEOGCS['GCS_WGS_1984',DATUM['D_WGS_1984',SPHE-
ROID['WGS_1984',6378137.0,298.257223563]],PRIMEM['Greenwich',0.0],UNIT['De-
gree',0.0174532925199433]],PROJECTION['Transverse_Mercator'],PARAME-
TER['False_Easting',500000.0],PARAMETER['False_Northing',0.0],PARAMETER['Cen-
tral_Meridian',9.0],PARAMETER['Scale_Factor',0.9996],PARAMETER['Lati-

```



```

Executing (4_Buffer): Modell "Feature Set" "50 Kilometers" C:\arcgisserver\directories\arcgisjobs\ma_brennecke_30.05.2015\buffer_gps-
erver\j759d59046ae3471aa7e7bf01f0685fc0\scratch\scratch.gdb\Buffer
Start Time: Tue Feb 24 16:30:09 2015
Executing (Modell): Modell "Feature Set" "50 Kilometers" C:\arcgisserver\directories\arcgisjobs\ma_brennecke_30.05.2015\buffer_gps-
erver\j759d59046ae3471aa7e7bf01f0685fc0\scratch\scratch.gdb\Buffer
Start Time: Tue Feb 24 16:30:09 2015
Executing (Buffer): Buffer "Feature Set" C:\arcgisserver\directories\arcgis-
jobs\ma_brennecke_30.05.2015\buffer_gps-
erver\j759d59046ae3471aa7e7bf01f0685fc0\scratch\scratch.gdb\Buffer "50 Kilome-
ters" FULL ROUND ALL #
Start Time: Tue Feb 24 16:30:10 2015
Succeeded at Tue Feb 24 16:30:12 2015 (Elapsed Time: 2.00 seconds)
Succeeded at Tue Feb 24 16:30:12 2015 (Elapsed Time: 3.52 seconds)
Succeeded at Tue Feb 24 16:30:12 2015 (Elapsed Time: 3.53 seconds)
Succeeded at Tue Feb 24 16:30:15 2015 (Elapsed Time: 6,06 seconds)
Executing (1_Project): Project Hessen PROJCS['WGS_1984_UTM_Zone_32N',GE-
OGCS['GCS_WGS_1984',DATUM['D_WGS_1984',SPHE-
ROID['WGS_1984',6378137.0,298.257223563]],PRIMEM['Greenwich',0.0],UNIT['De-
gree',0.0174532925199433]],PROJECTION['Transverse_Mercator'],PARAME-
TER['False_Easting',500000.0],PARAMETER['False_Northing',0.0],PARAMETER['Cen-
tral_Meridian',9.0],PARAMETER['Scale_Factor',0.9996],PARAMETER['Lati-
tude_Of_Origin',0.0],UNIT['Meter',1.0]]
Start Time: Tue Feb 24 16:30:15 2015
Executing (1_Project): Project "Feature Set" PROJCS['WGS_1984_UTM_Zone_32N',GE-
OGCS['GCS_WGS_1984',DATUM['D_WGS_1984',SPHE-
ROID['WGS_1984',6378137.0,298.257223563]],PRIMEM['Greenwich',0.0],UNIT['De-
gree',0.0174532925199433]],PROJECTION['Transverse_Mercator'],PARAME-
TER['False_Easting',500000.0],PARAMETER['False_Northing',0.0],PARAMETER['Cen-
tral_Meridian',9.0],PARAMETER['Scale_Factor',0.9996],PARAMETER['Lati-
tude_Of_Origin',0.0],UNIT['Meter',1.0]] c:\arcgisserver\directories\arcgis-
jobs\ma_brennecke_30.05.2015\project_gps-
erver\ja1893fd7bbf34b19afecd197e2718a75\scratch\scratch.gdb\Input_Proj
Start Time: Tue Feb 24 16:30:21 2015
Executing (Project): Project "Feature Set" PROJCS['WGS_1984_UTM_Zone_32N',GE-
OGCS['GCS_WGS_1984',DATUM['D_WGS_1984',SPHE-
ROID['WGS_1984',6378137.0,298.257223563]],PRIMEM['Greenwich',0.0],UNIT['De-
gree',0.0174532925199433]],PROJECTION['Transverse_Mercator'],PARAME-
TER['False_Easting',500000.0],PARAMETER['False_Northing',0.0],PARAMETER['Cen-
tral_Meridian',9.0],PARAMETER['Scale_Factor',0.9996],PARAMETER['Lati-
tude_Of_Origin',0.0],UNIT['Meter',1.0]] c:\arcgisserver\directories\arcgis-
jobs\ma_brennecke_30.05.2015\project_gps-
erver\ja1893fd7bbf34b19afecd197e2718a75\scratch\scratch.gdb\Input_Proj
Start Time: Tue Feb 24 16:30:21 2015
Executing (Project): Project "Feature Set" c:\arcgisserver\directories\arcgis-
jobs\ma_brennecke_30.05.2015\project_gps-
erver\ja1893fd7bbf34b19afecd197e2718a75\scratch\scratch.gdb\Input_Proj PRO-
JCS['WGS_1984_UTM_Zone_32N',GEOGCS['GCS_WGS_1984',DATUM['D_WGS_1984',SPHE-
ROID['WGS_1984',6378137.0,298.257223563]],PRIMEM['Greenwich',0.0],UNIT['De-
gree',0.0174532925199433]],PROJECTION['Transverse_Mercator'],PARAME-
TER['False_Easting',500000.0],PARAMETER['False_Northing',0.0],PARAMETER['Cen-
tral_Meridian',9.0],PARAMETER['Scale_Factor',0.9996],PARAMETER['Lati-
tude_Of_Origin',0.0],UNIT['Meter',1.0]] # GEOGCS['GCS_WGS_1984',DA-
TUM['D_WGS_1984',SPHEROID['WGS_1984',6378137.0,298.257223563]],PRIMEM['Green-
wich',0.0],UNIT['Degree',0.0174532925199433]]
Start Time: Tue Feb 24 16:30:22 2015
Succeeded at Tue Feb 24 16:30:24 2015 (Elapsed Time: 1.79 seconds)
Succeeded at Tue Feb 24 16:30:24 2015 (Elapsed Time: 3.73 seconds)
Succeeded at Tue Feb 24 16:30:24 2015 (Elapsed Time: 3.74 seconds)
Succeeded at Tue Feb 24 16:30:29 2015 (Elapsed Time: 13,13 seconds)
Executing (2_Extract_by_Location): ExtractByLocation "Feature Set" Grünberg "50
Kilometers"
Start Time: Tue Feb 24 16:30:29 2015
Executing (2_Extract_by_Location): ExtractByLocation "Feature Set" "Feature
Set" "50 Kilometers" c:\arcgisserver\directories\arcgis-
jobs\ma_brennecke_30.05.2015\extractbylocation_gps-
erver\j0cf33896c6c64d8bbf3ebecabe4d6d65\scratch\scratch.gdb\Extraction
Start Time: Tue Feb 24 16:30:35 2015

```

```

Executing (ExtractByLocation): ExtractByLocation "Feature Set" "Feature Set"
"50 Kilometers" c:\arcgisserver\directories\arcgis-
jobs\ma_brennecke_30.05.2015\extractbylocation_gps-
erver\j0cf33896c6c64d8bbf3ebecabe4d6d65\scratch\scratch.gdb\Extraction
Start Time: Tue Feb 24 16:30:35 2015
Executing (Make Feature Layer): MakeFeatureLayer "Feature Set" Input_Proj_Layer
# # "OBJECTID OBJECTID VISIBLE NONE;Shape Shape VISIBLE NONE;osm_id osm_id VIS-
IBLE NONE;name name VISIBLE NONE;type type VISIBLE NONE;population population
VISIBLE NONE"
Start Time: Tue Feb 24 16:30:37 2015
Succeeded at Tue Feb 24 16:30:37 2015 (Elapsed Time: 0.01 seconds)
Executing (Select Layer By Location): SelectLayerByLocation Input_Proj_Layer
WITHIN_A_DISTANCE "Feature Set" "50 Kilometers" NEW_SELECTION
Start Time: Tue Feb 24 16:30:37 2015
Succeeded at Tue Feb 24 16:30:37 2015 (Elapsed Time: 0.04 seconds)
Executing (Copy Features): CopyFeatures Input_Proj_Layer c:\arcgisserver\direc-
tories\arcgisjobs\ma_brennecke_30.05.2015\extractbylocation_gps-
erver\j0cf33896c6c64d8bbf3ebecabe4d6d65\scratch\scratch.gdb\Extraction # 0 0 0
Start Time: Tue Feb 24 16:30:37 2015
Succeeded at Tue Feb 24 16:30:41 2015 (Elapsed Time: 3.84 seconds)
Succeeded at Tue Feb 24 16:30:41 2015 (Elapsed Time: 5.99 seconds)
Succeeded at Tue Feb 24 16:30:41 2015 (Elapsed Time: 6.00 seconds)
Succeeded at Tue Feb 24 16:30:45 2015 (Elapsed Time: 16,58 seconds)
Executing (3_Near): Model Grünberg "Feature Set"
Start Time: Tue Feb 24 16:30:45 2015
Executing (3_Near): Model "Feature Set" "Feature Set" c:\arcgisserver\directo-
ries\arcgisjobs\ma_brennecke_30.05.2015\near_gps-
erver\ja0261c4e93444bce9e3150f89b213e2f\scratch\scratch.gdb\Near
Start Time: Tue Feb 24 16:30:49 2015
Executing (Model): Model "Feature Set" "Feature Set" c:\arcgisserver\directo-
ries\arcgisjobs\ma_brennecke_30.05.2015\near_gps-
erver\ja0261c4e93444bce9e3150f89b213e2f\scratch\scratch.gdb\Near
Start Time: Tue Feb 24 16:30:49 2015
Executing (Copy Features): CopyFeatures "Feature Set" c:\arcgisserver\directo-
ries\arcgisjobs\ma_brennecke_30.05.2015\near_gps-
erver\ja0261c4e93444bce9e3150f89b213e2f\scratch\scratch.gdb\Near # 0 0 0
Start Time: Tue Feb 24 16:30:50 2015
Succeeded at Tue Feb 24 16:30:54 2015 (Elapsed Time: 3.83 seconds)
Executing (Near): Near c:\arcgisserver\directories\arcgis-
jobs\ma_brennecke_30.05.2015\near_gps-
erver\ja0261c4e93444bce9e3150f89b213e2f\scratch\scratch.gdb\Near
in_memory\{BEB7E0A8-53BD-4403-ADD6-4FA0C7F6EC39} # NO_LOCATION NO_ANGLE PLANAR
Start Time: Tue Feb 24 16:30:55 2015
Determining data processing extents...
Building a neighborhood index from the Near Features...
Generating Near Table...
Found 374 feature(s)
Succeeded at Tue Feb 24 16:30:55 2015 (Elapsed Time: 0.87 seconds)
Succeeded at Tue Feb 24 16:30:56 2015 (Elapsed Time: 7.38 seconds)
Succeeded at Tue Feb 24 16:30:56 2015 (Elapsed Time: 7.40 seconds)
Succeeded at Tue Feb 24 16:31:00 2015 (Elapsed Time: 14,30 seconds)
Executing (5_Statistics): Model3 "Feature Set"
Start Time: Tue Feb 24 16:31:00 2015
Executing (5_Statistics): Model3 "Feature Set" c:\arcgisserver\directo-
ries\arcgisjobs\ma_brennecke_30.05.2015\statistics_gps-
erver\ja36697bbdff54079bad32ffa8e9356d7\scratch\scratch.gdb\Statistics
Start Time: Tue Feb 24 16:31:02 2015
Executing (Model3): Model3 "Feature Set" "population SUM;population MEAN;popu-
lation MIN;population MAX" c:\arcgisserver\directories\arcgis-
jobs\ma_brennecke_30.05.2015\statistics_gps-
erver\ja36697bbdff54079bad32ffa8e9356d7\scratch\scratch.gdb\Statistics
Start Time: Tue Feb 24 16:31:02 2015
Executing (Summary Statistics): Statistics "Feature Set" c:\arcgisserver\direc-
tories\arcgisjobs\ma_brennecke_30.05.2015\statistics_gps-
erver\ja36697bbdff54079bad32ffa8e9356d7\scratch\scratch.gdb\Statistics "popula-
tion SUM;population MEAN;population MIN;population MAX" #
Start Time: Tue Feb 24 16:31:03 2015
Succeeded at Tue Feb 24 16:31:04 2015 (Elapsed Time: 1.45 seconds)
Succeeded at Tue Feb 24 16:31:04 2015 (Elapsed Time: 2.70 seconds)
Succeeded at Tue Feb 24 16:31:04 2015 (Elapsed Time: 2.72 seconds)
Succeeded at Tue Feb 24 16:31:07 2015 (Elapsed Time: 7,96 seconds)

```



```

tral_Meridian',9.0],PARAMETER['Scale_Factor',0.9996],PARAMETER['Latitude_Of_Origin',0.0],UNIT['Meter',1.0]] c:\arcgisserver\directories\arcgis-jobs\ma_brennecke_30.05.2015\project_gps-server\j61ca57fcc6dc418bb6a3f2a61f525f69\scratch\scratch.gdb\Input_Proj
Start Time: Tue Feb 24 16:35:20 2015
Executing (Project): Project "Feature Set" PROJCS['WGS_1984_UTM_Zone_32N',GEOGCS['GCS_WGS_1984',DATUM['D_WGS_1984',SPHEROID['WGS_1984',6378137.0,298.257223563]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['Transverse_Mercator'],PARAMETER['False_Easting',500000.0],PARAMETER['False_Northing',0.0],PARAMETER['Central_Meridian',9.0],PARAMETER['Scale_Factor',0.9996],PARAMETER['Latitude_Of_Origin',0.0],UNIT['Meter',1.0]] c:\arcgisserver\directories\arcgis-jobs\ma_brennecke_30.05.2015\project_gps-server\j61ca57fcc6dc418bb6a3f2a61f525f69\scratch\scratch.gdb\Input_Proj
Start Time: Tue Feb 24 16:35:20 2015
Executing (Project): Project "Feature Set" c:\arcgisserver\directories\arcgis-jobs\ma_brennecke_30.05.2015\project_gps-server\j61ca57fcc6dc418bb6a3f2a61f525f69\scratch\scratch.gdb\Input_Proj PROJCS['WGS_1984_UTM_Zone_32N',GEOGCS['GCS_WGS_1984',DATUM['D_WGS_1984',SPHEROID['WGS_1984',6378137.0,298.257223563]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['Transverse_Mercator'],PARAMETER['False_Easting',500000.0],PARAMETER['False_Northing',0.0],PARAMETER['Central_Meridian',9.0],PARAMETER['Scale_Factor',0.9996],PARAMETER['Latitude_Of_Origin',0.0],UNIT['Meter',1.0]] # GEOGCS['GCS_WGS_1984',DATUM['D_WGS_1984',SPHEROID['WGS_1984',6378137.0,298.257223563]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]]
Start Time: Tue Feb 24 16:35:21 2015
Succeeded at Tue Feb 24 16:35:23 2015 (Elapsed Time: 1.87 seconds)
Succeeded at Tue Feb 24 16:35:23 2015 (Elapsed Time: 3.45 seconds)
Succeeded at Tue Feb 24 16:35:23 2015 (Elapsed Time: 3.47 seconds)
Succeeded at Tue Feb 24 16:35:24 2015 (Elapsed Time: 6.84 seconds)
Executing (2_Extract_by_Location): ExtractByLocation "Feature Set" "Feature Set" "50 Kilometers"
Start Time: Tue Feb 24 16:35:24 2015
Executing (2_Extract_by_Location): ExtractByLocation "Feature Set" "Feature Set" "50 Kilometers" c:\arcgisserver\directories\arcgis-jobs\ma_brennecke_30.05.2015\extractbylocation_gps-server\jfd7fca5a6a804a4b9a4eafd1c64abffb\scratch\scratch.gdb\Extraction
Start Time: Tue Feb 24 16:35:27 2015
Executing (ExtractByLocation): ExtractByLocation "Feature Set" "Feature Set" "50 Kilometers" c:\arcgisserver\directories\arcgis-jobs\ma_brennecke_30.05.2015\extractbylocation_gps-server\jfd7fca5a6a804a4b9a4eafd1c64abffb\scratch\scratch.gdb\Extraction
Start Time: Tue Feb 24 16:35:27 2015
Executing (Make Feature Layer): MakeFeatureLayer "Feature Set" Input_Proj_Layer # # "OBJECTID OBJECTID VISIBLE NONE;Shape Shape VISIBLE NONE;osm_id osm_id VISIBLE NONE;name name VISIBLE NONE;type type VISIBLE NONE;population population VISIBLE NONE"
Start Time: Tue Feb 24 16:35:29 2015
Succeeded at Tue Feb 24 16:35:29 2015 (Elapsed Time: 0.01 seconds)
Executing (Select Layer By Location): SelectLayerByLocation Input_Proj_Layer WITHIN_A_DISTANCE "Feature Set" "50 Kilometers" NEW_SELECTION
Start Time: Tue Feb 24 16:35:29 2015
Succeeded at Tue Feb 24 16:35:29 2015 (Elapsed Time: 0.04 seconds)
Executing (Copy Features): CopyFeatures Input_Proj_Layer c:\arcgisserver\directories\arcgisjobs\ma_brennecke_30.05.2015\extractbylocation_gps-server\jfd7fca5a6a804a4b9a4eafd1c64abffb\scratch\scratch.gdb\Extraction # 0 0 0
Start Time: Tue Feb 24 16:35:29 2015
Succeeded at Tue Feb 24 16:35:33 2015 (Elapsed Time: 4.17 seconds)
Succeeded at Tue Feb 24 16:35:34 2015 (Elapsed Time: 6.46 seconds)
Succeeded at Tue Feb 24 16:35:34 2015 (Elapsed Time: 6.48 seconds)
Succeeded at Tue Feb 24 16:35:34 2015 (Elapsed Time: 10.09 seconds)
Executing (3_Near): Model "Feature Set" "Feature Set"
Start Time: Tue Feb 24 16:35:34 2015
Executing (3_Near): Model "Feature Set" "Feature Set" c:\arcgisserver\directories\arcgisjobs\ma_brennecke_30.05.2015\near_gps-server\j284e4015cbbc49e3bd0f456d8d0fd7fe\scratch\scratch.gdb\Near
Start Time: Tue Feb 24 16:35:37 2015
Executing (Model): Model "Feature Set" "Feature Set" c:\arcgisserver\directories\arcgisjobs\ma_brennecke_30.05.2015\near_gps-server\j284e4015cbbc49e3bd0f456d8d0fd7fe\scratch\scratch.gdb\Near

```

```

Start Time: Tue Feb 24 16:35:37 2015
Executing (Copy Features): CopyFeatures "Feature Set" c:\arcgisserver\directories\arcgisjobs\ma_brennecke_30.05.2015\near_gps-
erver\j284e4015cbbc49e3bd0f456d8d0fd7fe\scratch\scratch.gdb\Near # 0 0 0
Start Time: Tue Feb 24 16:35:39 2015
Succeeded at Tue Feb 24 16:35:43 2015 (Elapsed Time: 4.00 seconds)
Executing (Near): Near c:\arcgisserver\directories\arcgis-
jobs\ma_brennecke_30.05.2015\near_gps-
erver\j284e4015cbbc49e3bd0f456d8d0fd7fe\scratch\scratch.gdb\Near
in_memory\{9F74CF85-4DB2-49A2-9695-13A28D96B8C8} # NO_LOCATION NO_ANGLE PLANAR
Start Time: Tue Feb 24 16:35:44 2015
Determining data processing extents...
Building a neighborhood index from the Near Features...
Generating Near Table...
Found 374 feature(s)
Succeeded at Tue Feb 24 16:35:45 2015 (Elapsed Time: 1.13 seconds)
Succeeded at Tue Feb 24 16:35:45 2015 (Elapsed Time: 7.69 seconds)
Succeeded at Tue Feb 24 16:35:45 2015 (Elapsed Time: 7.70 seconds)
Succeeded at Tue Feb 24 16:35:46 2015 (Elapsed Time: 11.54 seconds)
Executing (5_Statistics): Model3 "Feature Set"
Start Time: Tue Feb 24 16:35:46 2015
Executing (5_Statistics): Model3 "Feature Set" c:\arcgisserver\directo-
ries\arcgisjobs\ma_brennecke_30.05.2015\statistics_gps-
erver\j012312c1b5a84494a74355c681cb17ac\scratch\scratch.gdb\Statistics
Start Time: Tue Feb 24 16:35:48 2015
Executing (Model3): Model3 "Feature Set" "population SUM;population MEAN;popu-
lation MIN;population MAX" c:\arcgisserver\directories\arcgis-
jobs\ma_brennecke_30.05.2015\statistics_gps-
erver\j012312c1b5a84494a74355c681cb17ac\scratch\scratch.gdb\Statistics
Start Time: Tue Feb 24 16:35:48 2015
Executing (Summary Statistics): Statistics "Feature Set" c:\arcgisserver\direc-
tories\arcgisjobs\ma_brennecke_30.05.2015\statistics_gps-
erver\j012312c1b5a84494a74355c681cb17ac\scratch\scratch.gdb\Statistics "popula-
tion SUM;population MEAN;population MIN;population MAX" #
Start Time: Tue Feb 24 16:35:49 2015
Succeeded at Tue Feb 24 16:35:51 2015 (Elapsed Time: 1.60 seconds)
Succeeded at Tue Feb 24 16:35:51 2015 (Elapsed Time: 2.90 seconds)
Succeeded at Tue Feb 24 16:35:51 2015 (Elapsed Time: 2.92 seconds)
Succeeded at Tue Feb 24 16:35:52 2015 (Elapsed Time: 5.87 seconds)
Executing (6_Join): Model5 "Feature Set" "Record Set"
Start Time: Tue Feb 24 16:35:52 2015
Executing (6_Join): Model5 "Feature Set" "Record Set"
Start Time: Tue Feb 24 16:35:52 2015
Executing (Model5): Model5 "Feature Set" "Record Set"
Start Time: Tue Feb 24 16:35:53 2015
Executing (Join Field): JoinField "Feature Set" OBJECTID "Record Set" OBJECTID
SUM_population;MEAN_population;MIN_population;MAX_population
Start Time: Tue Feb 24 16:35:53 2015
Succeeded at Tue Feb 24 16:35:53 2015 (Elapsed Time: 0.01 seconds)
Succeeded at Tue Feb 24 16:35:53 2015 (Elapsed Time: 0.24 seconds)
Succeeded at Tue Feb 24 16:35:53 2015 (Elapsed Time: 0.25 seconds)
Succeeded at Tue Feb 24 16:35:53 2015 (Elapsed Time: 1.23 seconds)
Succeeded at Tue Feb 24 16:35:53 2015 (Elapsed Time: 44.18 seconds)
Succeeded at Tue Feb 24 16:35:53 2015 (Elapsed Time: 44.21 seconds)
Succeeded.

```

Szenario 5: Zeitmessung aus Postman REST Client

TIME 25578 ms
