

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta stavební
Katedra mapování a kartografie

in Zusammenarbeit mit der

HOCHSCHULE FÜR TECHNIK UND WIRTSCHAFT DRESDEN
Fakultät Geoinformation
Labor Photogrammetrie / Fernerkundung



Mapserver des Projekts Nasca

Masterarbeit

Petr Dobeš

Mai 2010

Erstbetreuer: Ing. Jiří Cajthaml, Ph.D.
Zweitbetreuer: MSc (GIS) Christiane Richter



ZADÁNÍ DIPLOMOVÉ PRÁCE

studijní program: Geodézie a kartografie
studijní obor: Geodézie a kartografie
akademický rok: 2009/2010

Jméno a příjmení diplomanta: Bc. Petr Dobeš
Zadávací katedra: K153 Katedra mapování a kartografie
Vedoucí diplomové práce: Ing. Jiří Cajthaml, Ph.D.
Název diplomové práce: Mapový server projektu Nazca
Název diplomové práce
v anglickém jazyce: Map server of the Nazca project

Rámcový obsah diplomové práce: Cílem práce je příprava a realizace mapového serveru pro projekt Nazca. Dílčí úkoly představují inventarizaci dat projektu na české i německé straně, návrh jednotné struktury dat pro ukládání na server a tvorba ukázkové aplikace mapového serveru.

Datum zadání diplomové práce: 21.9.2009 Termín odevzdání: 18.12.2009
(vyplňte poslední den výuky přísl. semestru)

Diplomovou práci lze zapsat, kromě oboru A, v letním i zimním semestru.

Pokud student neodevzdal diplomovou práci v určeném termínu, tuto skutečnost předem písemně zdůvodnil a omluva byla děkanem uznána, stanoví děkan studentovi náhradní termín odevzdání diplomové práce. Pokud se však student řádně neomluvil nebo omluva nebyla děkanem uznána, může si student zapsat diplomovou práci podruhé. Studentovi, který při opakovaném zápisu diplomovou práci neodevzdal v určeném termínu a tuto skutečnost řádně neomluvil nebo omluva nebyla děkanem uznána, se ukončuje studium podle § 56 zákona o VŠ č. 111/1998 (SZŘ ČVUT čl 21, odst. 4).

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.


vedoucí diplomové práce


vedoucí katedry

Zadání diplomové práce převzal dne: 21.9.2009


diplomant

Formulář nutno vyhotovit ve 3 výtiscích – 1x katedra, 1x diplomant, 1x studijní odd. (zašle katedra)

Nejpozději do konce 2. týdne výuky v semestru odešle katedra 1 kopii zadání DP na studijní oddělení a provede zápis údajů týkajících se DP do databáze KOS.

(Směrnice děkana pro realizaci stud. programů a SZZ na FSv ČVUT čl. 5, odst. 7)



Čj.: 563-01/09/11921/Ma

V Praze, dne 20.10.2009

Bc. Petr Dobeš

Pražská 1151/53

102 00 Praha 15

Věc: Žádost o prodloužení termínu odevzdání DPM

Vzhledem k uvedeným důvodům vyhovuji Vaší žádosti ze dne 19.10.2009 a souhlasím s prodloužením termínu pro odevzdání Diplomové práce a to nejpozději do **14.5.2010**. Neodevzdání Diplomové práce do tohoto termínu bude považováno jako nesplnění předmětu.

Dále souhlasím s vypracováním Diplomové práce v německém jazyce.

prof. Ing. Zdeněk Bittnar, DrSc.
děkan Fakulty stavební ČVUT



v.z. prof. Ing. Jiří Mlčá, CSc.
proděkan pro pedagogickou činnost

Na vědomí: Ing. Jiří Cajthaml, Ph.D. – vedoucí DPM

Vyřizuje: Miroslava Matoušová

Telefon: 224 354 886

Hiermit versichere ich, dass ich die vorliegende Masterarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Ausführungen, die anderen Schriften wörtlich oder sinngemäß entnommen wurden, sind in der Arbeit kenntlich gemacht.

Dresden, den 14. Mai 2010

Mein recht schöner Dank geht an Frau MSc (GIS) Christiane Richter und an Herrn Ing. Jiří Cajthaml, Ph.D. für die Betreuung meiner Diplomarbeit sowie für die fachliche Unterstützung und für mich aufgewandte Zeit.

Ohne Unterstützung der Förderer des Nascaprojekts, der Herren Prof. Dr.-Ing. Bernd Teichert und Prof. Dr. Ing. Karel Pavelka, wäre mein Aufenthalt in Dresden sowie die Arbeit am Projekt nur schwerlich denkbar.

Des Weiteren möchte ich meinen Mitbewohnerinnen Maria und Rahel, einer guten Freundin Antje und einem guten Freund Bert für grammatische und stilistische Korrekturen, sowie meinem Bruder Jan für die grafische Gestaltung und Satzsetzung danken.

Ein besonders herzlicher Dank geht an meine Eltern, die mich während meiner gesamten Studienzeit allseits unterstützt, und mir so diese Ausbildung ermöglichen haben.

Ein liebes Dankeschön geht an meine Freundin Míša, die mich während meiner Studienzeit immer unterstützt und motiviert hat und während meiner Auslandsaufenthalte viel Geduld gehabt hat.

MAPSERVER DES PROJEKTS NASCA

Abstrakt

Die Arbeit beschreibt die Datenhaltung und die WebGIS-Applikation des Forschungsprojekts Nasca. Im ersten analytischen Teil werden die vorhandenen Daten recherchiert und nach bestimmten Regeln gegliedert. Der zweite Teil stellt allgemeine Regeln für die effektive Verwaltung von Daten fest. Der dritte Teil ist rein praktisch und verwendet erworbene Kenntnisse beim Erstellen einer neuen WebGIS-Applikation Nasca Mapserver für das Nascaprojekt

Schlüsselwörter:

Nasca, Datenrecherche, Datenhaltung, WebGIS, Mapserver, Kartenapplikation, WMS, Mapbender

MAP SERVER OF THE NASCA PROJECT

Abstract

The thesis is concerned with data holding and with the WebGIS-Application in the research project Nasca. The first part is analytical, it gathers available data. The second part specifies general rules of efficient data management. The third part is practical, it applies acquired knowledges in the creation of new WebGIS application Nasca Mapserver for the Nasca project.

Keywords:

Nasca, data research, data management, Web GIS, Mapserver, maps application, WMS, Mapbender

MAPOVÝ SERVER PROJEKTU NASCA

Anotace

Práce se zabývá správou dat a tvorbou webové mapové aplikace pro výzkumný projekt Nasca. První část je analytická, shromažďuje dostupná data a třídí je podle různých hledisek. Druhá část práce je syntetická, stanovuje pravidla pro přehledné ukládání dat. Třetí část je praktická a aplikuje získané znalosti při tvorbě webové mapové aplikace Nasca Mapserver

Klíčová slova:

Nasca, řešerše dat, správa dat, WebGIS, Mapserver, mapavá aplikace, WMS, Mapbender

Rozšířená anotace

V Jižní Americe, v Peru, nedaleko města Nasca, se nachází v poušti tajuplná oblast. Na první pohled ze země není vidět téměř nic zvláštního, jen prach, štěrk, možná hromádky kamení. Ale na druhý pohled z vyhlídkové věže, nebo pak na třetí pohled z letadla je vidět, že poušť je poseta podivnými čarami a obrazci. Celé kresby jsou až stovky metrů velké, ale jednotlivé linie jsou jen několik decimetrů široké a jsou vyryty pár centimetrů hluboko. Jen díky místním klimatickým podmínkám se většina z nich dochovala. Obrazce nejsou v oblasti žádnou novinkou, nýbrž jsou ve skutečnosti více než 2000 let staré a pocházejí z kultury Nasca. Nikdo neví, proč byly vytvořeny, avšak existuje mnoho různých teorií.

Maria Reiche, rodačka z Drážďan, se snažila ve 40. až 80. letech 20. století rozluštit tajemství těchto geoglyfů. Její teorie, že se jedná o obrovský astronomický kalendář, ale nebyla doposud prokázána. V roce 1994 byl založen spolek „Dr. Maria Reiche – Linien und Figuren der Nasca-Kultur in Peru, e.V.“, který si klade za cíl chránit životní dílo Dr. Maria Reiche, pokračovat v její výzkumné práci a výsledky všech zúčastněných institucí a jednotlivců zpřístupnit široké veřejnosti. Za tímto účelem vznikl v roce 1995 na HTW Dresden výzkumný projekt Nasca pod vedením Prof. Dr.-Ing. Bernda Teicherta. Od roku 2004 je zde úzká spolupráce s Prof. Dr. Ing. Karlem Pavelkou z ČVUT v Praze. Celé téma výzkumného projektu Nasca zasahuje do mnoha různých oborů a poskytlo by dostatek práce pro archeology, historiky, geology, geografy, astronomy, zeměměřiče, kartografy a další odborníky. Projekt na HTW Dresden a ČVUT v Praze je zaměřen na sběr, správu, vyhodnocení a analýzu dat.

V letech 2004 a 2008 proběhly společné expedice do oblasti Nasca. Celkem bylo doposud v rámci projektu provedeno šest expedic (1994, 1996, 2003, 2004, 2005, 2008) a na léto roku 2010 je plánována sedmá. Sběr dat probíhá podle možností různými způsoby. Zaměření vlíčovacích bodů se provádí především na základě měření GPS přímo na místě. Vlastní tvorba vektorových dat se provádí digitalizací z družicových a leteckých snímků. V posledních letech byly zakoupeny satelitní snímky z družic ASTER, IKONOS, Quickbird a GeoEye, které pokrývají okolí planiny v Nasce a Palpě. ETH Zürich poskytla projektu letecké fotogrammetrické snímky ze snímkování v roce 1998. Další snímky pořídili účastníci expedic v letech 2004 a 2008 profesionální fotogrammetrickou kamerou během vyhlídkového letu z turistického letadla.

V závislosti na typu dat je následně nezbytné jejich zpracování. Jedná se především o metody jako jsou pansharpening, georeferencování pomocí vlíčovacích bodů, převzorkování, mozaikování. Metodami pro vyhodnocení jsou pak klasifikace nebo vektorizace.

Během trvání projektu bylo pořízeno a vyprodukováno velké množství dat a informací, které jsou (nebo smějí být) více či méně k dispozici veřejnosti. Pochopitelně vznikla již na několika místech myšlenka vytvořit k projektu databázi nebo GIS, aby byla data nejen přehledně uložena, ale zároveň aby mohla být prezentována navenek. Geografický informační systém (NascaGIS), ve kterém jsou uložena vektorová a věcná data, již existuje. Společné rozhraní, kde bychom mohli vidět všechny výsledky najednou, však zatím ne. Z této potřeby vychází zadání této práce. Je třeba nejprve shromáždit dostupná data, pak navrhnout pravidla pro jejich přehledné ukládání a nakonec může být vytvořena webová mapová aplikace.

První část práce poskytuje přehled o existujících médiích a datech v projektu Nasca. K dispozici jsou jak surová data, tak i výsledky náročných prací na projektu. Některá data zůstávají aktuální, některá jsou zastaralá nebo byla nahrazena novými. Nicméně všechna patří do projektu a měla by být skladována odpovídajícím způsobem. Vybrané výsledky by měly být prezentovány široké veřejnosti, aby zprostředkovaly poznání tajemné historie pouště v Peru. Seznam souborů je obsažen v `nasca_dateiliste.xls` (viz příloha ANHANG 1)

Po odhalení výhod a nedostatků v současném ukládání dat byla ve druhé části práce stanovena pravidla pro správu dat v projektu s cílem zlepšit spolupráci, interoperabilitu a urychlit výměnu dat. Pravidla jsou založena na jednotné, jasné a intuitivní adresářové struktuře.

V praktické části jsem se při tvorbě webové mapové aplikace Nasca Mapserver dozvěděl, jak se vypořádat s prostorovými daty na internetu. Tato aplikace byla první, kterou jsem kdy vytvořil, proto mi trvalo zpočátku relativně déle, než jsem pochopil potřebné technologické aspekty a než jsem mohl realizovat všechny podklady a návody. Zjistil jsem, že k efektivní tvorbě takovýchto aplikací jsou zapotřebí alespoň základní znalosti fungování internetu a dále znalosti programování (PHP, JavaScript, HTML, atd.). V těchto oblastech jsem narážel na největší potíže. Při práci v prostředí open-source software jsou důležité také dobré znalosti angličtiny, protože většina dokumentací, fór a mailinglistů jsou vedeny v angličtině.

Nakonec se mi podařilo vytvořit fungující aplikaci a nakonfigurovat ji podle zadaných požadavků. Je plná nejnovějších údajů a může být podle potřeby aktualizována a rozšířena. Kromě toho jsem připravil pro správce aplikace krátký manuál, viz příloha ANHANG 6. Aplikace Nasca Mapserver je dostupná jako veřejná verze na adrese <http://maps.fsv.cvut.cz/mapbender/frames/login.php?name=guest&password=guest> nebo jako heslem chráněná plná verze včetně licencovaných dat na adrese <http://maps.fsv.cvut.cz/mapbender/frames/login.php?name=vip>

Bylo mi potěšením podílet se na mezinárodním výzkumném projektu Nasca. Tuto příležitost považuji za opravdu výjimečnou. Rád jsem strávil poslední část svého studia v kruhu přátelských a zároveň kompetentních lidí, kteří jsou v projektu Nasca zainteresováni. Sám jsem během práce získal mnoho nových a užitečných dovedností a věřím, že jsem byl i já projektu prospěšný, když jsem jej rád doprovodil několik kroků ke zdárnému konci. Nakonec každý si může prohlédnout na adrese <http://maps.fsv.cvut.cz/~dobespet> nejen to, čím jsem přispěl k projektu, ale hlavně to, jaké světové dědictví nám všem zanechala před více než 2000 let kultura Nasca.

Inhaltsverzeichnis

Abbildungs- und Tabellenverzeichnis	11
Abkürzungsverzeichnis	12
1 Einleitung	13
2 Datenrecherche	15
2.1 Vorhandene Applikationen	15
2.1.1 GIS-Applikationen	15
2.1.2 WebGIS-Applikationen	16
2.2 Vorhandene Daten	17
2.2.1 Datenarten	18
2.2.2 Datenmerkmale	19
2.2.3 Rasterdaten	21
2.2.4 Vektordaten	23
2.2.5 Sachdaten	23
3 Datenstruktur	25
3.1 Datenspeicherung an der HTW Dresden	25
3.1.1 Rasterdaten	25
3.1.2 Vektor- und Sachdaten	25
3.2 Datenspeicherung an der TU Prag	26
3.3 Vorschläge zur Datenspeicherung	26
4 Erstellung einer WebGIS-Applikation	29
4.1 Prinzipien einer WebGIS-Applikation	29
4.1.1 Webserver	30
4.1.2 Geodaten-Server	30
4.1.3 Mapserver	30
4.1.4 Kartengestützte Online-Auskunftssysteme	32
4.1.5 WebGIS	32
4.1.6 OGC Webservices	32
4.1.7 Auswahl der Hardware- und Softwaremittel	33
4.2 Serverstruktur – Komponenten	34
4.2.1 Apache Webserver	34

4.2.2	UMN MapServer	35
4.2.3	PostgreSQL/PostGIS	35
4.2.4	PHP	35
4.2.5	Mapbender	36
4.2.6	Verzeichnisstruktur	36
4.3	Anforderungen an die Applikation	39
4.4	Installation und Konfiguration	40
4.4.1	MS4W	40
4.4.2	Apache, UMN MapServer und Konfigurationsdateien	40
4.4.3	Mapfile	41
4.4.4	PostgreSQL/PostGIS	42
4.4.5	Mapbender	43
4.5	Aufbau einer Benutzeroberfläche (GUI)	44
4.5.1	Farbgestaltung	46
4.5.2	Elemente	47
4.5.3	Dateneinbindung	52
4.5.4	Abfragen	56
4.5.5	Sicherheit und Lizenzen	57
5	Zusammenfassung und Ausblick	59
5.1	Zusammenfassung	59
5.2	Ausblick	60
	Nachwort	61
	Quellenverzeichnis	62
	Anhangsverzeichnis	65
	Anhänge	

Abbildungs- und Tabellenverzeichnis

ABB. 1	Linie vom Boden aus [WEB: PANORAMIO]	13
ABB. 2	Geoglyph Affe aus der Luft [WEB: WIKIMAPIA]	14
ABB. 3	Nasca WebGIS	16
ABB. 4	Nasca MapServer	16
ABB. 5	Anzeige verschiedener Rasterdaten im Nasca Mapserver	21
ABB. 6	Auszug aus den Vektordaten im Nasca Mapserver	23
ABB. 7	Vereinfachte Darstellung des Datenmodells NascaGIS [NACH RICHTER 2008]	24
ABB. 8	Verzeichnisstruktur zum Nascaprojekt	27
ABB. 9	Architektur eines UMN MapServers [WEB: MAPSERVER]	30
ABB. 10	OGC – Client/Server Beziehungen [WEB: WIKIPEDIA.EN]	33
ABB. 11	phpPgAdmin – Anlegen einer Datenbank	43
ABB. 12	Mapbender – GUI Liste	44
ABB. 13	GUI NascaMapserver_guest	46
ABB. 14	Mapbender – Modul GUI-Elemente editieren	47
ABB. 15	Mapbender – Variable der Elemente editieren	51
ABB. 16	Mapbender – WMS GUI Einstellungen	53
ABB. 17	Kachelung [SIEBOLD 2004]	55
ABB. 18	Bildpyramiden [SIEBOLD 2004]	55
ABB. 19	Mapbender – Datenbaum	56
ABB. 20	Mapbender – WFS konfigurieren	57
ABB. 21	GUI NascaMapserver_vip	58
TAB. 1	EPSG Code	21
TAB. 2	Farben in Nasca Mapserver	46
TAB. 3	Elemente in Nasca Mapserver	50
TAB. 4	Parent-Angabe in Ebenenbaum	53

Abkürzungsverzeichnis

CGI	Common Gateway Interface
CMS	Content Management System
ČVUT	České vysoké učení technické (TU Prag)
DGM	Digitales Geländemodell
EPSG	European Petroleum Survey Group
ESRI	Environmental Systems Research Institute
ETH	Eidgenössische Technische Hochschule (hier ETH Zürich)
GDAL	Geospatial Data Abstraction Library
GDI	Graphics Device Interface
GIS	Geoinformationssystem
GML	Geography Markup Language
GNU	GNU is Not Unix
GUI	Graphic User Interface
HTML	HyperText Markup Language
HTW	Hochschule für Technik und Wirtschaft (hier HTW Dresden)
ISO	International Organization for Standardization
JSON	JavaScript Object Nation
MS4W	MapServer for Windows
NaN	Not A Number
OGC	Open Geospatial Consortium, Inc.®
PHP	PHP Hypertext Preprocessor
PSAD56	Provisional South American Datum 1956
SDE	Spatial Database Engine
TU	Technische Universität
UMN	University of Minnesota
UTF8	Unicode Transformation Format-8
UTM	Universal Transverse Mercator
WCS	Web Coverage Service
WFS	Web Feature Service
WFS-T	WFS mit Transaktionsfähigkeit
WGS84	World Geodetic System 1984
WMS	Web Map Service
WWW	World Wide Web (Weltweites Gewebe)
XML	eXtensible Markup Language

1 Einleitung

In Südamerika, in Peru, in der Nähe der Stadt Nasca, befindet sich in der Wüste ein geheimnisvolles Gebiet. Beim ersten Anblick vom Boden aus sieht man kaum etwas Besonderes, nur Staub, Schotter, vielleicht mal Steinhäufchen. Aber schon beim zweiten Anblick vom Aussichtsturm oder dann beim dritten Anblick vom Flugzeug aus merkt man, dass die Wüste von seltsamen Linien und Figuren geprägt ist. Die ganzen Gebilde sind bis zu Hunderte Meter lang, aber einzelne Linien sind nur einige Dezimeter breit und ein paar Zentimeter tief in Staub und Schotter eingritzelt. Nur dank der lokalen Klimabedingungen sind uns die meisten der Vergangenheitszeugen erhalten geblieben. Die Zeichnungen sind nämlich keine Neuheit, sondern über 2000 Jahre alte Gebilde der damaligen Nasca-Kultur. Keiner weiß sicher, warum sie entstanden sind. Hierzu gibt es eine Vielzahl verschiedener Theorien.

Die Dresdnerin Maria Reiche versuchte in den 40er bis 80er Jahren des 20. Jahrhunderts während ihrer Forschungsarbeit das Geheimnis der Geoglyphen aufzulösen. Ihre Theorie, dass es sich um eine riesige astronomische Kalenderanlage handelt, ist jedoch bis heute nicht bewiesen. 1994 ist ein Verein mit Namen „Dr. Maria Reiche – Linien und Figuren der Nasca-Kultur in Peru, e.V.“ entstanden, um das Lebenswerk von Dr. Maria Reiche zu erhalten, zu würdigen und einer breiten Öffentlichkeit zugänglich zu machen, die Forschungsarbeit von Dr. Maria Reiche fortzuführen und die Ergebnisse allen interessierten Institutionen und Personen weltweit zugänglich zu machen. Zu diesem Zweck wurde 1995 das Forschungsprojekt Nasca an der HTW Dresden, unter der Leitung von Prof. Dr.-Ing. Bernd Teichert, ins Leben gerufen. Seit 2004 besteht eine enge Kooperation mit Prof. Dr. Ing. Karel Pavelka von der TU Prag. Das ganze Thema des Forschungsprojekts Nasca beinhaltet viele verschiedene Themenbereiche und böte genügend Arbeit für Archäologen, Historiker, Geologen, Geographen, Astronomen, Vermesser, Karto-

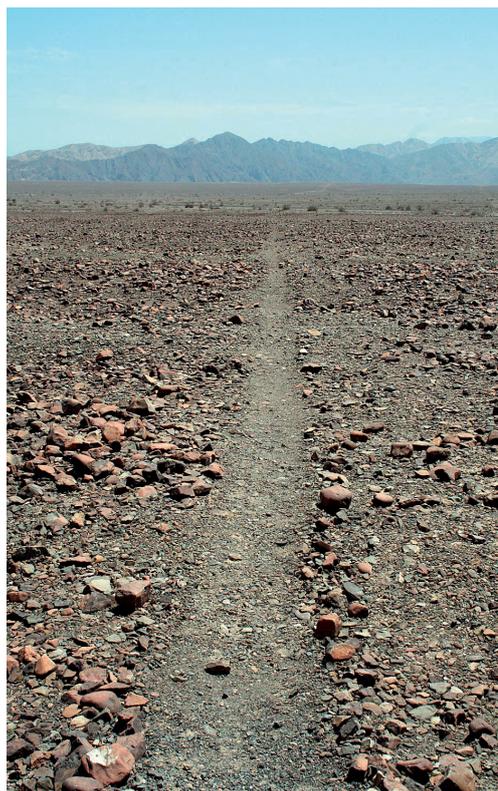


ABB. 1 Linie vom Boden aus [WEB: PANORAMIO]

graphen, sowie andere Fachleute. Der Schwerpunkt des Projektes an der HTW Dresden bzw. der TU Prag besteht in der Erfassung, Verwaltung, Auswertung und Analyse von Daten.

In den Jahren 2004 und 2008 fanden gemeinsame Expeditionen nach Nascagebiet statt. Insgesamt wurden bis dato sechs Expeditionen durchgeführt (1994, 1996, 2003, 2004, 2005, 2008) und im Sommer 2010

plant der Verein die siebte. Die Erfassung von Daten erfolgt nach Möglichkeiten auf verschiedenste Weise. Die Bestimmung von Passpunkten erfolgt vorrangig durch GPS-Messungen vor Ort. Die eigentliche Vektordatenerfassung erfolgt durch Digitalisierung aus Satelliten- oder Luftbildern. Für große Bereiche der Pampa von Nasca und Palpa wurden in den letzten Jahren Daten der Satelliten ASTER, Ikonos, Quickbird und GeoEye gekauft. Photogrammetrische Luftbilder einer Befliegung aus dem Jahr 1998 wurden dem Projekt von der ETH Zürich zur Verfügung gestellt. 2004 und 2008 haben Teilnehmer der Exkursion Aufnahmen mit professioneller Ausrüstung bei einem Rundflug vom touristischen Flugzeug aus aufgenommen.

Je nach Typ der Daten sind im Anschluss verschiedene Bearbeitungsprozesse wie z.B. Pansharpening, Georeferenzierung mit Passpunkten, Resampling, Mosaikierung, sowie Auswertungsmethoden wie Klassifizierung oder Vektorisierung nötig.

Während der Dauer des Projekts wurden viele Daten und Informationen erfasst und produziert, die mehr oder weniger der Öffentlichkeit zugänglich sind (und zugänglich werden dürfen). Selbstverständlich ist auch mehrmals und an mehreren Stellen die Idee entstanden, eine Datenbank oder GIS zu erstellen, um die Daten nicht nur griffbereit und übersichtlich zurechtgelegt zu haben, sondern auch sie nach außen präsentieren zu können. Ein Geoinformationssystem (NascaGIS), in dem die Vektor- und Sachdaten gespeichert werden, existiert bereits. Eine gemeinsame Schnittstelle, wo man alle Ergebnisse auf einmal sehen kann, gibt es dennoch nicht. Aus diesem Bedarf stammt auch die Zielsetzung dieser Masterarbeit. Es sollten erstens alle vorhandenen Daten recherchiert werden, dann eine einheitliche Struktur der Datenspeicherung entworfen werden und letztendlich eine WebGIS-Applikation erstellt werden.



ABB. 2 Geoglyph Affe aus der Luft [WEB:WIKIMAPIA]

2 Datenrecherche

Im Rahmen des Nascaprojektes haben sich neben Wissenschaftlern und Projektmitarbeitern auch viele Studenten mit der Thematik beschäftigt. Zum größten Teil sind ihre Ergebnisse in Bachelor-, Diplom- oder Masterarbeiten zusammengefasst. Eine umfangreiche Übersicht der Abschlussarbeiten, sowie verschiedenste Literaturquellen sind auf der Webseite [WEB:HTW NASCA] aufgelistet. Wenige Abschlussarbeiten wurden auch an der TU Prag und an der Beuth Hochschule für Technik Berlin (ehemals TFH Berlin) betreut. Die Arbeiten im Original sind dann entweder in der Fakultätsbibliothek oder am Lehrstuhl erhältlich. Daneben gibt es über das Nascagebiet, die Figuren und Linien sowie über Theorien, wie sie entstanden sind, verschiedene Bücher, Artikel und Internetseiten, die sich mit diesem geheimnisvollen Thema beschäftigen. Es ist fast unmöglich, alle Medien zu finden und zu sammeln und es gehört auch nicht zur Aufgabe dieser Masterarbeit. Trotzdem ist es sinnvoll, einige im Zusammenhang stehende Medien zu erwähnen.

Zu jedem Forschungsprojekt, also das Projekt Nasca nicht ausgenommen, gehört untrennbar die Datensammlung. Für eine entstehende WebGIS-Applikation ist es erforderlich, eine übersichtliche Liste aller relevanten Daten anzufertigen, damit der Entwickler die Möglichkeit hat, die beste Kombination der Daten in die Applikation zu implementieren. Deshalb bestand der erste Teil dieser Arbeit in einer ausführlichen Datenrecherche. Da die sich auf das Nascaprojekt beziehenden Daten an verschiedenen Orten liegen, und damit ich sie während des Arbeitsverlaufs immer griffbereit habe, habe ich mich entschieden, alle Daten auf ein externes Laufwerk zu kopieren und dann sich mit denen schrittweise vertraut zu machen. Für eine im Projekt nicht involvierte Person ist es nicht leicht, die Übersicht über die ganze Menge der Daten zu behalten. Zum Glück waren mir sachkundige Personen behilflich. Da das Ziel der Arbeit im Aufbau einer WebGIS-Applikation besteht, soll sich die Datenrecherche nur auf die im Nascaprojekt vorhandenen Geodaten beziehen.

2.1 VORHANDENE APPLIKATIONEN

2.1.1 GIS-Applikationen

NascaGIS

nur intern an der HTW Dresden zugänglich

Im Rahmen mehrerer Diplomarbeiten an der HTW Dresden entstand ein Datenmodell für das NascaGIS, welches unter Topobase™ der Firma Autodesk (ursprünglich c-plan)

implementiert ist. Die heutzutage laufende Oracle 11g Datenbank ist natürlich auch lange nicht die erste Generation der Datenbank zum Nascaprojekt an der HTW. Angefangen mit dem Entwurf eines Datenmodells wurde schon 1996 und seitdem wurde das Modell mehrmals erweitert, überarbeitet und hat zwischen verschiedenen Versionen gewechselt. Das letzte Datenmodell wurde von Sebastian Baumhekel und Thomas Schuster entwickelt und unter Topobase 2007 implementiert. Das NascaGIS ist aus meiner Sicht das Herzstück für Vektor- und Sachdaten im Projekt. Hier werden sämtliche Daten und Informationen zu den Linien und Figuren erfasst, verwaltet, analysiert und für die Präsentation in Form von thematischen topographischen Karten bzw. in einer Internetapplikation aufbereitet. Der Inhalt der Datenbank wird innerhalb des Kapitels 2.2.4 näher betrachtet.

2.1.2 WebGIS-Applikationen



ABB. 3 Nasca WebGIS

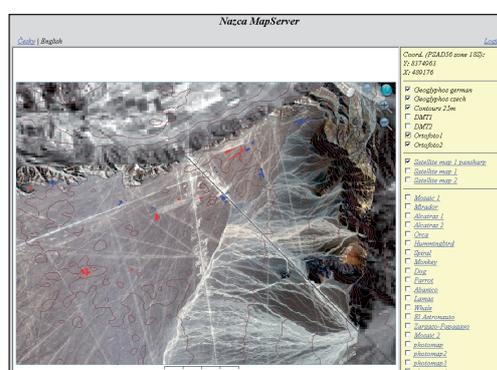


ABB. 4 Nazca MapServer

Nasca WebGIS

<http://141.56.141.2:8080/mapbender2.5/http/frames/login.php?name=Nasca&password=Nasca>

Nasca WebGIS ist das Ergebnis einer Diplomarbeit von Stephan Klemm an der HTW Dresden 2008. Es ermöglicht die Anzeige eines Satellitenbildes, eines digitalen Geländemodells und es wird Wert auf die Darstellung des Datenbankinhalts gelegt. Es sind auch Abfragen an die Sachdaten möglich. Diese Applikation ähnelt am meisten der von mir entwickelten Applikation. Damit es sich bei mir nicht um eine sinnlose Kopie handelt, werden teilweise andere Technologien angewendet und sie wird um neue Funktionalitäten und um zahlreiche neue Ergebnisse erweitert.

Nazca MapServer

<http://maps.fsv.cvut.cz/nazca>

Diese auf einem UMN MapServer, einer nazca.map Datei und einer index.php Datei basierte Applikation wurde aufgrund eines Mangels an Präsentationsmitteln für das Nascaprojekt an der TU Prag erstellt. Es wurde als provisorische Lösung behandelt und deswegen wurde auf jegliche Schönheit verzichtet. Hier sind überwiegend Ergebnisse von Diplomarbeiten implementiert. Diese Applikation sollte durch die innerhalb meiner

Masterarbeit neuentstandene Applikation ersetzt werden. Früher gab es noch eine tschechische Variante `index_cz.php`, die ist jedoch veraltet und wird nicht mehr aktualisiert.

2.2 VORHANDENE DATEN

In diesem Kapitel werden ausschließlich elektronische Daten betrachtet, die zum Zweck der Arbeit am Nascaprojekt angeschafft geworden sind, oder die als Ergebnis der Arbeiten am Projekt entstanden sind. Bevor wir zur Analyse der eigentlichen, vorhandenen Daten kommen, sollen zunächst die möglichen Datenarten und die Datenmerkmale näher betrachtet werden. Es ist prinzipiell möglich sich verschiedene Gliederungsaspekte auszudenken. Als Beispiel werden hier einige Möglichkeiten angeführt. Bei der Betrachtung der einzelnen Daten wird dann eine solche Kombination der Gliederungsaspekte angewendet, dass letztendlich fast alle Möglichkeiten vorkommen.

Gliederung nach Art (Form, Format) der Daten

- Rasterdaten
- Vektordaten
- Sachdaten

Gliederung nach Art der Speicherung

- Datenträger
- Server
- PC
- Datenbank

Gliederung nach Stufe der Verarbeitung

- Rohdaten
- vorbearbeitete Daten
- Daten im Verarbeitungsprozess
- Endergebnisse

Weiter wäre eine Gliederung nach Erfassungs-, Beschaffungs- bzw. Bearbeitungsdatum möglich, oder auch viele andere.

Bei meiner Datenrecherche habe ich mich entschieden, eine Tabelle zu erstellen, wo alle relevanten Dateien aufgelistet sein sollten. Als ich dann die Dateien einzeln untersucht habe, sind Anforderungen an Attribute entstanden und die Gestalt der Tabelle unterlag stetigen Änderungen und Anpassungen, bis die heutige Variante `nasca_dateiliste.xls` (siehe ANHANG 1) sich herauskristallisiert hat. Da sich die Tabelle als übersichtlich und hilfreich erwiesen hat, habe ich sie anschließend im Laufe der ganzen Arbeit verwendet. Die Software MS Excel ermöglicht eine übersichtliche Gestaltung und Verwaltung der Tabellen. Ich habe besonders die Funktion Liste erstellen ausgenutzt, da die Liste verschiedene Auswahlregeln ermöglicht. Man kann so sehr effektiv die Dateien suchen und sortieren und die Attributwerte sind damit schnell griffbereit. So eine Arbeit ähnelt in vielem einer

einfachen Datenbank. Wird künftig eine größere Menge von Dateien im Projekt erwartet, fände ich es sinnvoll, eine Datenbank zu erstellen. Die könnte man besser an verschiedene Dateiformate anpassen und damit leere Zellen vermeiden.

2.2.1 Datenarten

Es wurden bereits viele mögliche Definitionen von Datenarten formuliert, sie weichen allerdings im Grunde genommen nur wenig ab. Im Prinzip wird immer zwischen Raster-, Vektor- und Sachdaten unterschieden. Einzelne Definitionen sind beispielsweise in [WEB:ABC GEODATA], [LINDER 1999] oder [BILL 1999] angeführt. Eine aphoristische Beschreibung der in dieser Arbeit vorkommenden Daten wurde für diese Arbeit von [WEB:UNI STUTT GART] übernommen.

Daten

Quantitative und qualitative Beschreibung von Eigenschaften von regelmäßigen und unregelmäßigen Einheiten oder Objekten des gerade betrachteten Interessengebietes.

Geodaten

Geodaten sind digitale Informationen, mit einem Bezug zum Raum (Lage, Höhe). Sie übermitteln Informationen über Sachverhalte, Eigenschaften, Ereignisse, Merkmale und Objekte auf, unter und über der Erdoberfläche. Im herkömmlichen (analogen) Sinn handelt es sich um die Information, die in Form von Plänen und Landkarten dargestellt werden können. Geodaten werden mit GIS oder CAD-Systemen erzeugt, gespeichert und gepflegt.

Geometriedaten

Die Geometrie von räumlichen Objekten wird durch die Form und relative Lage von Punkten vollständig beschrieben. Festlegung durch Distanzen und Winkel oder durch ein Koordinatensystem. Der Punkt (Knoten) ist der Träger der geometrischen Information.

Vektordaten

Auf Punkten beruhende Beschreibung von raumbezogenen Daten. Grundelemente sind Punkt, Linie und Fläche. Sie haben folgende Eigenschaften:

- Punkt und Linie als graphische Grundstrukturen, Fläche als geschlossener Linienzug
- Daten nach Objektlinien geordnet, dadurch linienhafte Betrachtungsweise
- Logische Datenstrukturierung und Objektbezug leicht möglich
- Punktuelle Datenerfassung durch den Einsatz von bewährten Methoden, jedoch hohe Erfassungszeiten
- Geringe Datenmengen, kurze Rechenzeiten
- Datengewinnung durch geodätische Aufnahme- und Berechnungsverfahren, Digitalisierung von analoger Karteninformation.

Rasterdaten

Die Darstellung bezieht sich direkt auf Flächen statt auf Linien. Geometrisches Grundelement ist das Pixel, welches zeilen- und spaltenweise in einer Matrix gleichförmiger quadratischer oder rechteckiger Elemente angeordnet ist und einheitliche Flächenfüllung aufweist. Eigenschaften:

- Pixel als graphische Grundstruktur
- Flächenhafte Betrachtungsweise
- Ordnung nur nach der Position der Pixel
- Logische Datenstrukturierung und Objektbezug sehr eingeschränkt
- Einfache Datenerfassung, kurze Erfassungszeiten
- Große Datenmengen, dadurch hoher Rechenaufwand
- Datengewinnung durch Scannen der Erdoberfläche mittels satellitengetragener Spezialkameras oder von analogen Vorlagen (Luftbilder, Orthophotos, Karten)

Grafikbeschreibende Daten

Aussagen über die Art und Weise, wie ein räumliches Objekt (Geometrie und Attribute) unter einer bestimmten Thematik dargestellt werden soll. Dazu gehören Farbe, Füllung, Symbol, Linienstil, Flächenstil, Textfonts, Textpositionierung, Grauwerte, etc. Eine Vektorgrafik ergibt sich aus der Ergänzung der geometrischen Grundelemente um die Grafikdaten (leicht veränderbar). Eine Rastergrafik entsteht durch die Manipulation und graphische Ausgestaltung von Rasterdaten durch digitale Bildverarbeitung oder Visualisierungstechniken (unintelligente Grafik).

Sachdaten

Alle nichtgeometrischen Elemente wie Text, Zahlen, Messwerte, Nummern, Namen, Eigenschaften, etc. Sie werden auch als thematische Daten, Attribute, beschreibende Daten genannt.

2.2.2 Datenmerkmale

Jede Datenart hat ihre spezifischen Merkmale, die sie von anderen eindeutig unterscheiden. Bei Rasterdaten könnte das beispielsweise die Pixelgröße sein, bei Vektordaten z.B. die Differenzierung zwischen Punkten, Linien und Flächen. Es gibt dennoch auch Merkmale, die nicht von der Datenart abhängen und somit alle oder fast alle Daten betreffen. Nimmt man eine Gliederung, ist es gleich zu merken, dass nicht alle Merkmale immer sinnvoll sind. Dies ist die Begründung dafür, dass ich nicht alle Felder in `nasca_dateiliste.xls` (siehe ANHANG 1) ausgefüllt habe. Damit man zwischen fehlenden und nicht relevanten Angaben unterscheiden kann, werden die fehlenden leer gelassen, die nicht relevanten indessen mit der Abkürzung *NaN* (Not A Number) erledigt.

Formate

Vorhandene Dateien sind selten einzeln dagewesen, sondern befanden sich überwiegend in Gruppen, zu denen ein Dateiname mit mehreren Dateiendungen gehörte. Bei manchen Formaten war dies erforderlich, bei manchen trat es optional bzw. nur unter be-

stimmten Bedingungen auf. Für eine genauere Beschreibung einzelner Dateieindungen wird hier auf *nasca_dateiliste.xls* (siehe ANHANG 1) verwiesen. Hier werden die wesentlichen Gruppen vorgestellt (immer nur die Erweiterung):

- TIF+TFW** *tif* ist ein Dateiformat zur Speicherung von Bilddaten, während die *tfw*-Datei der *tif*-Datei Georeferenz gibt, zusammen bilden sie das Geotif-Format .
- IMG+RRD+IGE** *img* ist ein Arbeitsformat der Software ERDAS, *rrd* sind dazugehörige Bildpyramiden (nicht mit *rrd* von ESRI kompatibel) und *ige* wird erstellt als Informationsträger bei Dateien größeren als 4GB.
- SHP+DBF+SHX+SBN+SBX** ist ein Shapefile und besteht aus mindestens drei Dateien. *shp* dient zur Speicherung der Geometriedaten, *dbf* enthält Sachdaten im dBase-Format und *shx* dient als Index der Geometrie zur Verknüpfung der Sachdaten (auch Attributdaten genannt). Optional gehören dazu noch *sbx* und *sbn* – Index für Tabellenverbindungen (Joins) und mehrere andere. In einem Shapefile können jeweils nur Elemente eines Typs enthalten sein (Punkte, Linien oder Flächen).

Rasterdaten

Rasterdaten zeichnen sich durch folgende Attribute aus:

- Pixelgröße
- Spektralbereich (Kanäle)
- Bildtiefe (Bittiefe)

Vektordaten

Bei Vektordaten kommen selten spezifische Attribute vor, eher werden sie durch allgemeine Merkmale beschrieben. Man unterscheidet zwischen Punkten, Linien und Flächen.

Sachdaten

Sachdaten an sich haben keinerlei Merkmale, da es sich lediglich um beschreibende Daten handelt. In Zusammenhang mit der Datenbank könnte man eventuell die Wahrscheinlichkeit bei unsicheren Angaben als Attribut nehmen. Sieht man die Sachdaten als Schrift in einer Karte an, würde man dann Schriftart, Schriftgröße, Schnitt usw. als Attribute nehmen.

Allgemeine Merkmale

Merkmale, die alle Dateien verbinden, hängen entweder von der allgemeinen Dateistruktur im Rechner oder vom Raumbezug ab. Optional könnte man auch Herkunft der Daten und Aufnahmedatum erwähnen, wobei näher bestimmt werden müsste, was genau damit gemeint ist, das heißt ob es sich um Erfassungs-, Beschaffungs- oder Bearbeitungsdaten handelt und ob es sich bei der Herkunft um Sensor, Ankäufer oder Bearbeiter handelt. Hier also eindeutige Merkmale:

- Pfad
- Dateiname

- Dateigröße
- Standort, Speicherplatz
- Urheberrecht, Lizenz

Bei Geodaten zusätzlich noch

- Ausdehnung (BoundingBox)
- Räumliche Genauigkeit
- Raumbezug, Koordinatensystem

EPSG Code	Datum	Ellipsoid	Projektion	Einheiten
24878	PSAD56	Hayford International	UTM Zone 18 S	Meter
4326	WGS84	WGS84	UTM Zone 18 S	LatLon
32767	User-Defined	User-Defined	User-Defined	User-Defined

TAB. 1 EPSG Code

Hinweis: Bei Hayford International Ellipsoid tritt manchmal die zusätzliche Beschreibung 1909 oder 1924 auf. Beide Beschreibungen weisen auf die gleichen Ellipsoide hin. Ein Archiv der EPSG Codes in verschiedenen Formaten findet man unter [WEB:SPATIAL REFERENCE] und Dokumentation zur EPSG Codierung ist unter [WEB:EPSG] abrufbar.

2.2.3 Rasterdaten

Rohdaten

Seit Beginn des Nascaprojekts sind viele Rohdaten angeschafft und gesammelt worden. Obgleich die Rohdaten selten gleich verwendbar sind und man viele Bearbeitungen vornehmen muss, sind sie wertvolle Daten, ohne die keine weitere Arbeit möglich wäre. Meistens handelt es sich um gekaufte Satellitenbilder, SRTM Geländemodelle oder Luftbilder. Im Folgenden ist eine Liste der Daten mit ausgewählten Eigenschaften (Aufnahmedatum, Satellit, Kanäle und entsprechende Auflösung) zusammengestellt. Eine erschöpfende Beschreibung ist in `nasca_dateiliste.xls` (siehe ANHANG 1) enthalten. Nur aus Übersichtsgründen wird hier getrennt nach Beschaffungsquelle. Ansonsten gehören alle Daten zum Projekt und werden nach Möglichkeiten ausgetauscht und gemeinsam benutzt.

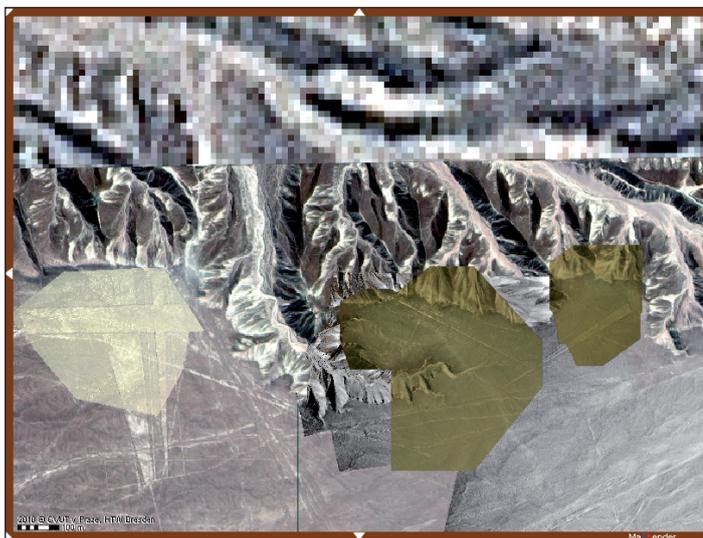


ABB. 5 Anzeige verschiedener Rasterdaten im Nasca Mapserver

- TU Prag**
 - 2002, 2005 – QuickBird – PAN 0,6m – RGBN 2,4m
 - 2004, 2008 – Luftbilder für Fotomosaik
 - 2007 – Aster – VNIR 15m – SWIR 30m – TIR 90m
 - 2009 – GeoEye – PAN 0,5m – RGBN 2,0m
- HTW Dresden**
 - 1998 – photogrammetrische Luftbilder
 - 2000 – Landsat7 – PAN 15m – RGB+IR 30m – TIR 60m
 - 2000 – SRTM-1 und SRTM-3
 - 2000, 2001 – IKONOS – PAN 1,0m – RGBN 4,0m
 - 2002, 2005 – QuickBird – PAN 0,6m – RGBN 2,4m
 - Fotos aufgenommen mit Amateurkameras

Ergebnisse

Während die Rohdaten den Anfangspunkt für Arbeiten an unserem Projekt bilden, fragt die Außenwelt meistens eher nach Ergebnissen. Die Ergebnisse werden im Folgenden auch für die Präsentation in einer WebGIS-Applikation verwendet. Deshalb legte ich Wert darauf, mir einen guten Überblick über die vorhandenen Endergebnisse zu verschaffen. An dieser Stelle wird wieder eine Übersicht aufgelistet und für beschreibende Einzelheiten wird auf `nasca_dateiliste.xls` (siehe ANHANG 1) hingewiesen.

- TU Prag**
 - Gescannte georeferenzierte Karten 1:50 000 mit Passpunkten (60-er Jahre) (Bachelorarbeit Hanzalová 2008)
 - Digitales Geländemodell aus Asterdaten (Bachelor- und Masterarbeit Hanzalová 2008, 2010), ermöglicht u. a. Export von Höhenlinien, Anpassung der Orthophotokarte
 - Fotomosaik (7 Bilder) aus Rundflug (Masterarbeit Hanzalová 2010)
 - Orthophoto aus Asterdaten (Masterarbeit Hanzalová 2010)
 - Fotomosaik der einzelnen Geoglyphen (Bachelorarbeit Kuldanová 2008)
 - Gescannte georeferenzierte topographische Karten von Palpa 1:100 000 (Bachelorarbeit Kuldanová 2008)
 - Gescannte georeferenzierte topographische Karten von Nasca 1:10 000 (Bachelorarbeit Kuldanová 2008)
 - Transformierte und maskierte Satellitenbilder von QuickBird 2009 Daten (Bachelorarbeit Kuldanová 2008)
- HTW Dresden**
 - Gescannte Karten 1:25 000
 - IKONOS, Teile der Mosaik
 - Landsatmosaik, pansharpend, bearbeitet
 - QuickBird+IKONOS Mosaik
 - Luftbildmosaik aus 179 Luftbildern
 - Digitales Geländemodell SRTM-3 und SRTM-1 (Teil)

2.2.4 Vektordaten

Ergebnisse

- TU Prag**
- Vektorisierte Geoglyphen (Bachelorarbeit Kuldanová 2008)
 - Höhenlinien abgeleitet aus DGM (Bachelor- und Masterarbeit Hanzalová 2008, 2010)
- HTW Dresden**
- GPS-Messergebnisse von 1996, 2004, 2008 (Passpunkte, Geoglyphen, Linien)
 - aus topographischen Karten digitalisierte Höhenlinien, Gewässer, Straßen, Gebäude
 - photogrammetrische Auswertung der von der ETH Zürich zur Verfügung gestellten Luftbilder in Hauptgebiet der Pampa von Nasca (Linien, Flächen, Geoglyphen, Straßen)
 - aus Satellitenbildern digitalisierte Linien, Flächen, Geoglyphen, Straßen
 - aus Karte von M. Reindl digitalisierte Linien, Flächen, Geoglyphen, Straßen (Palpa)

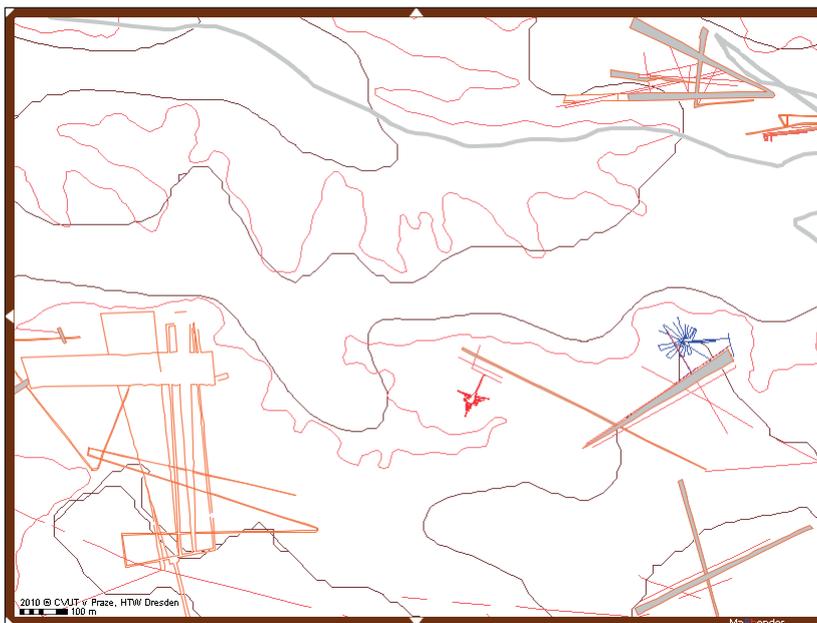


ABB. 6 Auszug aus den Vektordaten im Nasca Mapserver

2.2.5 Sachdaten

Alle relevanten Sachdaten werden im NascaGIS an der HTW gespeichert. Dazu gehören vor allem die Bereiche nach ABB. 7, sowie alle Attribute in restlichen Tabellen, die nicht die Vektorgeometrie an sich beschreiben, sondern zusätzliche Information gewährleisten. Im Laufe der Zeit wurde ein ziemlich umfangreiches Datenmodell entwickelt, dessen Inhalt stets erweitert wird. Deshalb gebe ich an dieser Stelle nur grob die wesentlichen Entitäten an und die nähere Beschreibung wäre für Interessenten in [BAUMHEKEL, SCHUSTER 2007] ausführlich erklärt. Das Datenmodell ist ebenfalls in Digitalform in ANHANG 1 beigefügt.

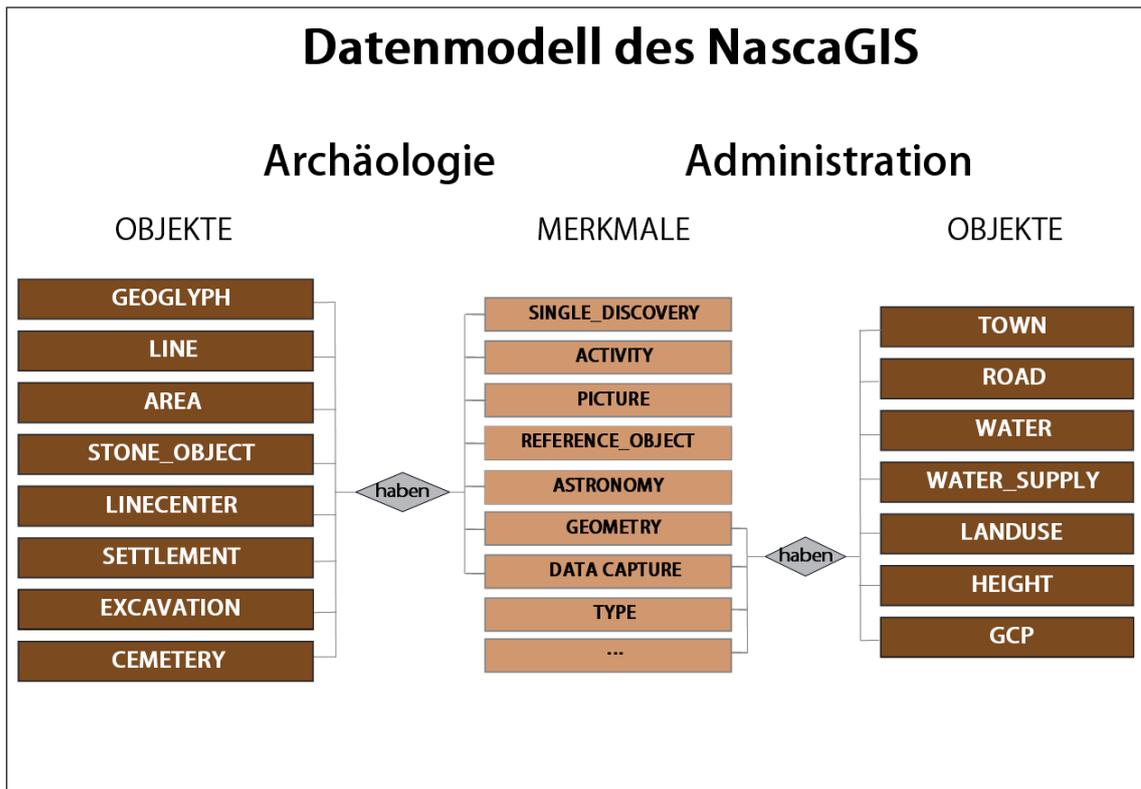


ABB. 7 Vereinfachte Darstellung des Datenmodells NascaGIS [NACH RICHTER 2008]

3 Datenstruktur

Wenn man mit Daten arbeitet, müssen sie irgendwo abgelegt werden. Wenn man mit großen Mengen der Daten arbeitet, ist es wünschenswert, auch eine Übersicht zu haben. Zum Projekt Nasca gibt es eine Menge Daten. Sie haben unterschiedliche Formen, Formate, Bedeutung, Herkunft und unterliegen unterschiedlichen Urheberrechten. Weitere Erschwernisse stammen davon, dass sich verschiedene Daten an verschiedenen Orten befinden. In diesem Kapitel wird zusammengefasst, wo und wie welche Daten abgelegt werden und anschließend werden Vorschläge geäußert, wie man die Datenspeicherung effizienter organisieren könnte.

3.1 DATENSPEICHERUNG AN DER HTW DRESDEN

Die Datenspeicherung an der HTW Dresden ist im Grunde genommen recht gut aufgebaut. Die Daten sind nach ihrer Art geteilt und übersichtlich auf verlässlichen Servern abgelegt. Der Zugriff auf Daten ist geschützt und nur befugten Personen werden die Zugriffsrechte erteilt. Die Überwachung leistet die Laboringenieurin Christiane Richter. Die Server werden regelmäßig gesichert (Backup).

3.1.1 Rasterdaten

Alle relevanten Rasterdaten zum Projekt Nasca sind auf dem Server *KFS1* gespeichert. In einem speziellen Projektverzeichnis *Nasca* befindet sich ein Unterverzeichnis *Bildmosaik*, in dem man Satellitenbilder, Karten, Mosaike, Passpunktskizzen, Terrainmodelle sowie andere graphische Elemente, die zum Projekt Beziehung haben, findet. Meistens sind das *tif*- oder *img*-Dateien mit dazugehörigen Nebendateien. Das ganze Verzeichnis beträgt zur Zeit etwa 42 GB.

3.1.2 Vektor- und Sachdaten

Für eine gute Handhabung der Vektor- und Sachdaten hat sich ein Geoinformationssystem am besten bewährt. Deshalb wird seit langem an der HTW Dresden ein Datenmodell entwickelt. Im Laufe der Zeit unterlag das Modell ständigen Änderungen, bis es heutzutage ein sehr umfangreiches Modell auf der Basis einer Oracle 11g Datenbank gibt, das zahlreiche Abfragen und Funktionalitäten ermöglicht. Der Zugriff auf die Datenbank ist nur beschränkt möglich, damit wirklich nur berechtigte Ergänzungen und Änderungen vorgenommen werden. Die Datenbank liegt auf dem Server *KS2*, der nur intern an der HTW zugänglich ist. Zur Präsentation des *NascaGIS* im Internet wird eine Kopie der Datenbank auf dem Internetserver *KIS2* bereitgestellt.

3.2 DATENSPEICHERUNG AN DER TU PRAG

Die Datenspeicherung an der TU Prag erfolgt heutzutage nur auf der Basis von Datenträgern. Die Rohdaten sowie Ergebnisse einzelner Bachelor- bzw. Masterarbeiten sind auf CDs oder DVDs gespeichert und stehen in Karel Pavelkas Büro zu Verfügung. Ein Teil der Daten, die für bestehende WebGIS-Applikation relevant sind, ist parallel auf dem Server *maps.fsv.cvut.cz* abgelegt. Dort werden alle Dateien (Rasterbilder sowie Shape-Dateien) in ein gemeinsames Verzeichnis kopiert.

3.3 VORSCHLÄGE ZUR DATENSPEICHERUNG

Damit eine effektive Arbeit bzw. Mitarbeit bei einem Projekt möglich ist, erfordert es gute Kommunikation und Synchronisation in vielen Bereichen. Beispielsweise bei Datenaustausch ist es sinnvoll, wenn alle Beteiligten ein Format liefern. Im Nascaprojekt ist das zwar nicht immer der Fall, da mit unterschiedlicher Software gearbeitet wird, nichtsdestotrotz ist ein Datenaustausch nicht so kompliziert, weil es die meisten Programme ermöglichen, andere Formate zu importieren oder zu exportieren und damit die Variabilität und Interoperabilität zu gewährleisten. Manchmal ist es sogar erforderlich, mit mehreren Programmen zu arbeiten, denn jedes hat andere Möglichkeiten und Funktionalitäten und somit können unterschiedliche Wege zur Lösung führen. Insbesondere bei Forschungsprojekten ist es somit sinnvoll, verschiedene Ansätze zu prüfen, um letztendlich den Besten auszuwählen.

In der Datenspeicherung ist es dagegen wünschenswert, eine möglichst einheitliche Struktur beizubehalten (hiermit wird vorzugsweise Verzeichnisstruktur gemeint, oder Datenbankstruktur). Die Struktur sollte übersichtlich und selbsterklärend sein. Das alles hilft besonders dort, wo das Personal oft wechselt. Obwohl im Nascaprojekt nicht viele Leute parallel arbeiten und die Studenten von Fachleuten betreut werden, fände ich es hilfreich, wenn die Verzeichnisstruktur einheitlich wäre. Es sprechen dafür mehrere Gründe. Wenn man nämlich die Daten gleichförmig strukturiert hat, beugt man Missverständnissen und vergeblicher Arbeit vor. Ich hätte beispielsweise bei der Erstellung der Mapserver-Applikation nicht lange prüfen müssen, welche Daten sind doppelt oder welche Ergebnisse von welchen Daten stammen.

Bei der Datenspeicherung muss vermieden werden, doppelte Dateien an mehreren Stellen zu haben, weil es bei Änderungen zur Inkonsistenz von Daten führt. Es gibt selbstverständlich Fälle, in denen eine Datei an mehreren Stellen sinnvoll erscheinen kann, hier wird indessen dringend empfohlen, eine verlinkte Datei zu legen. Dies betrifft nicht die Notwendigkeit, Sicherheitskopien anzulegen.

Bei der Überlegung, wie man die Struktur aufbauen könnte, fielen mir ein paar grundlegende Anforderungen ein:

- In der Verzeichnisstruktur sollte deutlich gekennzeichnet sein, in welchem Zustand sich die Daten befinden, also es ist grundsätzlich zu unterscheiden zwischen Rohdaten und Ergebnissen.
- Speziell beim Nascaprojekt ist es aus Übersichtsgründen nicht vernachlässigbar, welche Institution Daten gekauft bzw. produziert hat. Deswegen ist es praktisch,

dies bei Verzeichnisbenennung zu berücksichtigen. Da es sich allerdings um keine andere Art der Daten, sondern um deren Eigenschaft handelt, ist auf untergeordnete Verzeichnisse zu verzichten. Zu viele Unterebenen machen die Struktur nämlich auch unüberschaubar.

- Wenn möglich, sind in die Verzeichnisse beschreibende *txt*-Dateien zu legen. Auf sie bin ich bei der Durchsuchung der Verzeichnisse an der HTW gestoßen und ich finde es sehr hilfreich, wenn man eine kurze Beschreibung zu den Daten lesen kann.

Unter den oben genannten Bedingungen habe ich folgende Verzeichnisstruktur entworfen:

Eine Grundlage bilden die Verzeichnisse

Endergebnisse/

hier wird unterschieden zwischen

Diplomarbeiten/ eine Kopie beiliegender CD

Fertigprodukte/ kann vorteilhaft verlinkte Dateien enthalten

DGM/ Digitales Geländemodell

Luftbilder/ Photopläne, Mosaik

Satellitenbilder/ Orthophoto, Mosaik

Topokarten/ Topographische Karten

Vektoren/ Geoglyphen, Höhenlinien, Datenbankabdruck,...

Original Rasterdaten/

Hier werden alle gekauften, bzw. erfassten Bilddateien abgelegt, eventuell können noch vorbearbeitete Daten im Unterverzeichnis *Vorbearbeitet* daneben gespeichert werden.

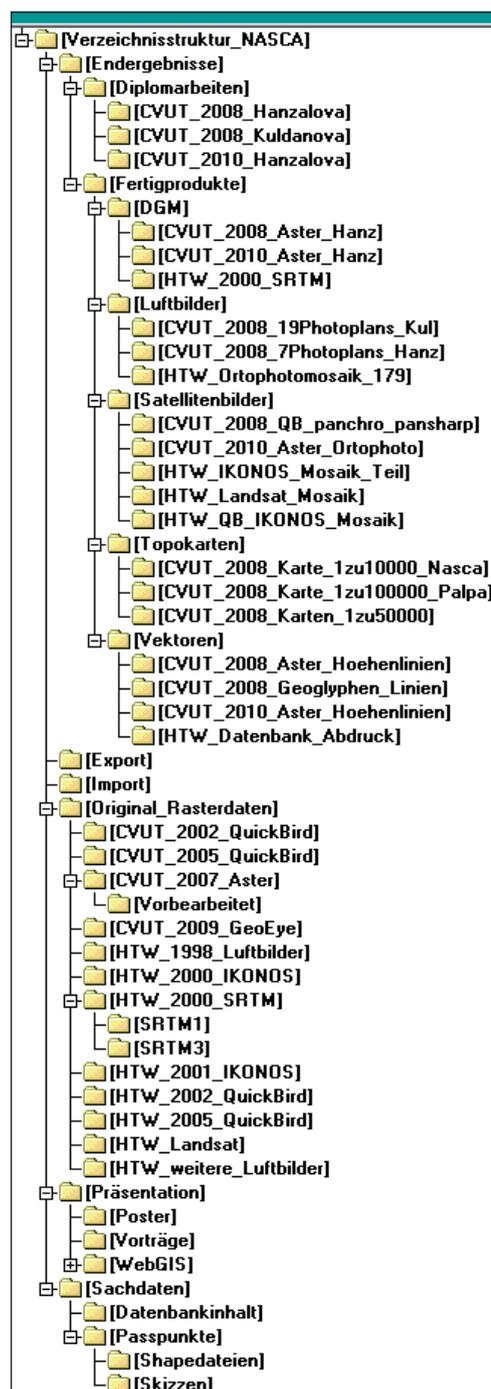


ABB. 8 Verzeichnisstruktur zum Nascaprojekt

Unterstützend habe ich noch weitere Verzeichnisse angelegt:

Export/

Hier könnten optional zweckmäßig erstellte Dateien für Export gespeichert sein.

Import/

Hier sollten vorübergehend die noch nicht eingeordneten Dateien abgelegt werden.

Präsentation/

Poster, Webseiten, Graphik,...

Sachdaten/

Hier würden in unserem Fall Passpunktskizzen oder ähnliche beschreibende Informationen abgelegt, weiter könnte hier eventuell der ganze Inhalt der Datenbank gespeichert werden.

Entsprechend der Bedeutung, der Wichtigkeit und des Wertes sollten regelmäßig Sicherheitskopien von Daten angefertigt werden. Spätestens dann, wenn eine wichtige Änderung oder eine Verbesserung vorgenommen wird.

Ein Vorschlag für die TU Prag wäre, Daten auf einem Server oder auf eine externe Festplatte zu speichern. Durch Speicherung auf einer Stelle und unter oben genannten Bedingungen wäre die Konsistenz des Datenbestands gewährleistet.

Eine weitere Möglichkeit, wie die Handhabung von Daten übersichtlicher gemacht werden kann, ist der Eintrag in einer Datenbank. In unserem Fall stände ein neuer Abschnitt in der bestehenden Datenbank zur Abwägung. Am Anfang eines Projekts, wo viele Daten zu erwarten sind, ist sicherlich mit einer Datenbank zu rechnen. Wenn man jedoch am Anfang nicht weiß, wie viele Dateien gesammelt werden, wird auf die Datenbank logischerweise zuerst entweder verzichtet oder gar nicht gedacht. Und es stimmt auch, dass die Datenbank für Dateien sich erst ab einer bestimmten Menge an Dateien lohnt. Sonst ist es mehr Aufwand als Beitrag. Deshalb ist es kein Wunder, dass eine solche Datenbank, bzw. Datenbankabschnitt bislang fehlt. Ich stelle jetzt die Idee zur Abwägung, die bestehende Datenbank um ein Paar Tabellen zu erweitern. Dabei ist auch die Frage zu beantworten, wie lange das Projekt noch läuft und wie viele Leute noch daran beteiligt werden und ob es sich letztendlich überhaupt lohnt oder eher die Tabelle `nasca_dateiliste.xls` (siehe ANHANG 1) weitergeführt wird.

Für den Fall, dass der Aufbau einer Dateidatenbank erwogen wird, schlage ich hier meine ersten Gedanken vor, wie die Tabellen und deren Attribute aussehen könnten:

- **Datei** Name, Pfad, Größe, Format, angehörige Erweiterungen, Beschreibung, Lizenz
- **Satelliten-/Luftbild** Sensor, Aufnahmedatum, Kanäle, Bildtiefe, Pixelgröße
- **Geometrie** Geodätisches Datum, Ellipsoid, Projektion, EPSG Code, Räumliche Genauigkeit, georeferenziert mit Passpunkten, Boundingbox

Um die Mehrzahl der Informationen zu gewinnen ist die Utility `gdalinfo.exe` aus der GDAL/OGR-Bibliothek sehr gut geeignet. Sie liefert Metadaten wie Format, Pixelanzahl in x/y-Richtung, Koordinatensystem, Pixelgröße, Boundingbox, Kachelung, Anzahl der Bildpyramiden und noch andere. Näher ist diese Utility unter [WEB:GDAL] beschrieben.

4 Erstellung einer WebGIS-Applikation

Vor der tatsächlichen Erstellung einer WebGIS-Applikation ist zu entscheiden, in welcher Hardware- und Software-Umgebung wir uns bewegen werden. In einer komplexen Betrachtung sind proprietäre (kommerzielle) sowie freie und Open-Source-Möglichkeiten zu erwähnen. Proprietäre Programme werden jedoch von Hochschulen nur selten verwendet, da sie manchmal sehr teuer sind und jegliche Anpassungen kaum oder überhaupt nicht möglich sind. Deshalb wird auch beim Nascaprojekt auf proprietäre Software größtenteils verzichtet und nur kostenfreie Produkte verwendet und damit auch hier nur sie betrachtet.

Bei Open-Source-Software handelt es sich um ein Entwicklungsmodell, bei dem der Quelltext frei zu Verfügung steht und jeder Änderungen und Anpassungen vornehmen und auch weitergeben darf. Wer aber nicht selbst Software entwickeln kann, hat dennoch die Möglichkeit, eine kompilierte Version herunterzuladen. Bei manchen Programmen bietet sich auch eine freie Version an. Das heißt, das Produkt wird zunächst vom Entwickler unter das Copyright gestellt, aber allen wird das Recht gegeben, diese Software zu verwenden.

Damit die Entwicklung der Open-Source-Programme eine Ordnung hat und auch die Kompatibilität sichergestellt wird, wurde die OSGeo (Open Source Geospatial Foundation) gegründet. Diese Organisation unterstützt die Entwicklung, Verbreitung und Nutzung von Freier und Open-Source-Software in der räumlichen Datenverarbeitung. OSGeo entwickelt auch die Standards und deren Einhaltung garantiert dann die Kompatibilität aller Projekte. Bevor ein Projekt offiziell von OSGeo angenommen wird, müssen bestimmte Bedingungen erfüllt sein.

4.1 PRINZIPIEN EINER WEBGIS-APPLIKATION

Es muss unterschieden werden, ob man fertig aufbereitete Daten zur Verfügung stellen will oder ob der Benutzer Einfluss auf die Präsentation der Geodaten nehmen kann. Es existieren aus dieser Fragestellung heraus zwei Modelle für den Aufbau eines Mapservers. Im einen Fall steht eine Reihe von Karten zur Verfügung, auf die der Benutzer zugreifen kann. Im anderen Fall liefert eine Mapserver-Software Zugriff auf die Daten, der Benutzer kann über das Layout und die eingebundenen Ebenen entscheiden. In bestimmten Anwendungen ist es von Bedeutung, dass immer aktuelle Daten zur Verfügung stehen, z.B. wenn das aktuelle Verkehrsaufkommen in einem Stadtgebiet dargestellt werden soll. In diesem Fall steht der direkte Zugriff auf die zu Grunde liegenden Daten im Vordergrund.

Andere Applikationen verlangen nach einer exakten kartographischen Aufbereitung, die vom Herausgeber vorgegeben ist. In diesen Fällen steht die Aktualität nicht an vorderster Stelle, vorgefertigte Karten können als Präsentationsform gewählt werden. [PRÜLLER 2006]

Diese Unterschiede ermöglichen uns jetzt, technische Möglichkeiten der Geodatenpräsentation im Internet in Kategorien annähernd [FÜRPAß 2001] einzuteilen.

4.1.1 Webserver

Ein Webserver ist ein Computer, der Dokumente an Clients wie z. B. Webbrowser überträgt. Als Webserver bezeichnet man den Computer mit Webserver-Software oder nur die Webserver-Software selbst. Webserver werden lokal, in Firmennetzwerken und überwiegend als WWW-Dienst im Internet eingesetzt. Dokumente können somit dem geforderten Zweck lokal, firmenintern und weltweit zur Verfügung gestellt werden. [WEB:WIKIPEDIA.DE: WEBSERVER]

4.1.2 Geodaten-Server

Geodatenserver sind die unterste Kategorie im Bereich von WebGIS-Applikationen. In diesem Fall besitzt ein Desktop-GIS die Möglichkeit, sich Daten von entfernten Rechnern zu laden und diese lokal weiter zu verarbeiten. Der Rechner, der die Geodaten zum Herunterladen bereitstellt, wird als Geodaten-Server bezeichnet. Für die Auswahl der Geodaten sind Recherchemöglichkeiten notwendig. Die Aufgabe der Kartenerstellung liegt weiterhin beim Client, der ausgehend von seiner Funktionalität somit als „thick client“ bezeichnet werden kann. [WEB:WIKIPEDIA.DE: WEB GIS]

4.1.3 Mapserver

Für den Begriff Mapserver gibt es keine eindeutige Definition. Eine Vorstellung welche Funktion ein Mapserver in einer internetbasierenden Geodatenapplikation ausübt, ist in ABB. 9 zu sehen. Unter einem Mapserver kann man aber in jedem Fall ein Programm verstehen, das Karten nach bestimmten Anforderungen erstellt und diese dann ausliefert. Der klassische Ansatz ist ein Programm in einem Webserver (z.B. Apache) laufen zu lassen, das über URL Parameter angesprochen werden kann. Um die Interaktion mit dem Benutzer zu

gestalten, wird das Programm über ein HTML-Interface angesprochen. Man kann prinzipiell zwischen statischen und dynamischen Mapservern unterscheiden. [PRÜLLER 2006]

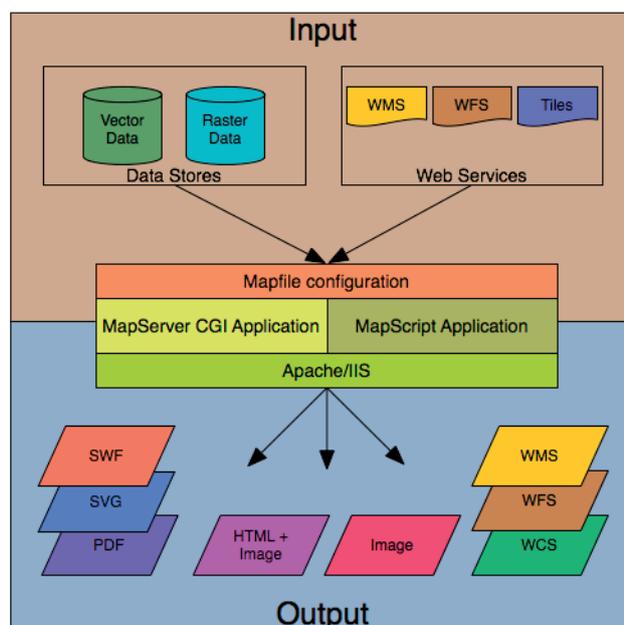


ABB. 9 Architektur eines UMN MapServers [WEB:MAPSERVER]

Statisch (thin client)

Hier liegt die Aufgabe der Kartenerstellung beim Server, genauer gesagt muss darunter aber noch nicht die dynamische Erstellung einer Karte verstanden werden. Weit verbreitet sind die so genannten „static maps“ bei denen das Rasterbild einer Karte angeboten wird, wie beispielsweise bei Anfahrtsskizzen. Auch über die HTML-Technik der „image-maps“ werden zahlreiche Karten im Internet angeboten, die dem Benutzer aber die Möglichkeit geben, einen bestimmten Kartenausschnitt näher zu betrachten. Dabei ist die Karte ebenfalls statisch, der Client benötigt aber nur noch eine eingeschränkte Funktionalität. [WEB:WIKIPEDIA.DE: WEB GIS]

Dynamisch (thick client)

Die Karten werden auf Anfrage vom Client dynamisch erzeugt und zurückgegeben. Der Server wertet die Anfrage aus und übermittelt die relevanten Angaben einem Kartengenerator. Dabei wird sich zum einen nur auf die unterschiedlichen Darstellungsformen (z. B. Farbe, Symbole) beschränkt andererseits können auch themenbezogene Auswahlmöglichkeiten (Layer) vorhanden sein (Online-Auskunftssysteme). Hier wird dann serverseitig die Anfrage bearbeitet, aus den vorhandenen Daten der angefragte Ausschnitt ausgewählt und eine Karte produziert. [WEB:WIKIPEDIA.DE: WEB GIS]

Mögliche Datenquellen eines Mapservers:

- GIS-Daten
- ESRI-Shapedatei (Vektordaten)
- Rasterdaten (meistens *tif*)– Visualisierung in *png*, *jpg*

Ab dieser Untergliederung ist es wichtig, die Begriffe Client/Server und Map-Client/Mapserver voneinander zu unterscheiden. Bisher wurde von der technischen Seite des Internet-Client und des Internet-Servers gesprochen. Dabei wurde unterschieden, ob der Client viel Funktionalität bereitstellen muss (thick client) oder wenig (thin client). Arbeitet man in den Termini von Mapserver und Map-Client muss beachtet werden, dass man bei der oben beschriebenen Arbeit auf einem Mapserver nicht ohne einen geeigneten Map-Client auskommt. Dies ist in der Regel eine Webseite, die z. B. die Funktionen der Layerauswahl bereitstellt, das Kartenbild darstellt und die verschiedenen Navigationsmöglichkeiten (Panning/Bewegung, Zoom) anbietet. Alternativ kann das auch ein vorgefertigter Map-Client, wie z.B. Mapbender, ka-map, chameleon oder andere sein. Der Map-Client ist ein Bindeglied zwischen dem Internet-Client, oder besser Benutzer und dem Mapserver. Er kann entweder vom Betreiber des Mapservers selbst angeboten werden oder ist ein externer Map-Client der mit dem Mapserver kommunizieren kann. Die Funktionalität liegt auf keinen Fall beim Benutzer, dieser ist immer noch als „thin“ zu bezeichnen, da die gesamte Funktionalität über einen normalen Webbrowser erreicht werden kann. [WEB:WIKIPEDIA.DE: WEB GIS]

4.1.4 Kartengestützte Online-Auskunftssysteme

Kartengestützte Online-Auskunftssysteme dienen der Visualisierung vorgefertigter oder interaktiv erstellter Karten, bieten dem Client aber darüber hinaus thematische und einfache raumbezogene Abfragemöglichkeiten. Dies kann beim Erstellen der Karte oder interaktiv auf Basis einer entsprechenden Mapping-Komponente geschehen. [WEB:GISINFO]

Hier sind beliebte Streckenplaner als ein typisches Beispiel zu erwähnen. Es wird sowohl durch Texteingaben wie auch direkt interaktiv in einer Karte eine gewünschte Strecke gefordert, wobei zusätzliche Bedingungen wie z.B. kürzeste oder schnellste Weg in Anspruch genommen werden können.

4.1.5 WebGIS

Unter dem Begriff WebGIS wird im Allgemeinen ein Geoinformationssystem (GIS) verstanden, dessen Funktion teilweise auf der Netzwerktechnologie von Internet und Intranet basiert. Als Synonym wird WebGIS oftmals mit den Termini OnlineGIS, InternetGIS, NetGIS, DistributedGIS oder Webmapping bezeichnet. Unter Webmapping, dem allgemeinen Begriff, kann man theoretisch auch nur den Abruf von Geodaten aus dem Internet in einem DesktopGIS verstehen. Solange mindestens zwei Rechner miteinander kommunizieren und Geodaten austauschen bzw. GIS-Funktionalität bereitstellen (Client-Server Prinzip) kann schon von einem WebGIS gesprochen werden. Daher ist es sinnvoll, verschiedene WebGIS Typen zu unterscheiden. Dabei unterscheiden diese sich in der bereitgestellten Funktionalität des Serverrechners bzw. des Clientrechners. Bei einem typischen DesktopGIS sind die gesamten GIS-Funktionalitäten und die Daten auf einem Clientrechner vorhanden. Bei freien WebGIS Architekturen wie beispielsweise dem UMN MapServer in Verbindung mit einem Webclient ist die Funktionalität des Clients auf die Visualisierung und triviale GIS-Funktionen (Pan, Zoom, Distanzmessung,...) beschränkt, während die Hauptlast der Arbeit bei einem oder mehreren Servern liegt. Für den Client benötigt man hierbei nur noch einen einfachen Webbrowser.

[WEB:WIKIPEDIA.DE: WEB GIS]

4.1.6 OGC Webservice

Das Open Geospatial Consortium (OGC) hat in Zusammenarbeit mit der International Organization for Standardization (ISO) eine Reihe von Standards für den Austausch räumlicher Daten entwickelt. Das Ziel des OGC ist es, die Interoperabilität zwischen verschiedenen Anwendungen mithilfe einer gemeinsamen Sprache zum Datenaustausch über die entsprechenden Standards herzustellen. Das Hauptaugenmerk liegt hierbei nicht auf der Entwicklung eigener Applikationen, sondern auf der Entwicklung von Spezifikationen für den Datenaustausch. Deshalb finden sich in diesem Konsortium alle wichtigen Anbieter von proprietären GIS wie Autodesk, ESRI oder Intergraph wieder. Hierdurch wird die gemeinsame Nutzung von professioneller Open-Source-Software mit proprietären Formaten möglich. [MITCHELL 2008]

Der eingesetzte UMN MapServer implementiert viele der Standards des OGC. Die für unsere Applikation interessantesten Standards sind:

kopierbar. Die ganze Vorgehensweise der Installation und Konfiguration beschreibe ich ausführlich im Kapitel 4.4. Hier werde ich nur die wesentlichen Komponenten nennen, sie werden anschließend in weiteren Abschnitten betrachtet.

Die erste Variante der Konfiguration, mit der ich lange Zeit gearbeitet habe und viele Versuche durchgeführt habe, basierte auf einem MS4W Packet, Version 2.3.1. Dies beinhaltet:

- Apache HTTP Server Version 2.2.14
- PHP Version 5.3.1
- MapServer CGI 5.6.1
- MapScript 5.6.1 (CSharp, Java, PHP, Python)
- Unterstützung für Oracle 11g und SDE Daten
- GDAL/OGR 1.7.0 und Utilities
- PROJ Utilities

Zusätzlich war noch die Installation der Datenbank PostgreSQL, 8.4.2-1 einschließlich PostGIS erforderlich. Letztendlich wurde ein Mapbender 2.5.2 Pack installiert.

Später habe ich auf die Versuche mit externem Laufwerk verzichtet und habe mich entschieden, direkt auf dem Server *maps.fsv.cvut.cz* zu arbeiten. Es hat sich als gute Entscheidung erwiesen, da dort bereits mehrere erforderliche Komponenten vorhanden waren und dadurch musste ich nur kleine Anpassungen vornehmen. Die Konfiguration des Servers ist wie folgt:

- Betriebssystem Linux Ubuntu
- Apache HTTP Server Version 2.2.8
- PHP Version 5.2.4
- MapServer CGI 5.4.2
- MapScript 5.4.2 (CSharp, Java, PHP, Python)
- GDAL/OGR und Utilities
- PROJ 4.6.1
- PostgreSQL 8.1.10/PostGIS 1.3.3

Zusätzlich habe ich den Mapbender 2.5.1 implementiert.

4.2 SERVERSTRUKTUR – KOMPONENTEN

4.2.1 Apache Webserver

Der Apache Webserver ist der am weitesten verbreitete Webserver überhaupt. Er ist ein Open-Source-Projekt, welches das World Wide Web schon seit den frühen Anfängen begleitet. Dieses Alter, verknüpft mit der Erfahrung der Programmierer und einer langen Entwicklungszeit führte mit den aktuellen Versionen zu einem der stabilsten, sichersten und leistungsfähigsten Webserver. Manch Gigant der Softwareindustrie wird wohl diese Position nie auch nur annähernd erreichen. Der Apache läuft unter Unix- und Linux-Derivaten, BeOS, OS2, Netware aber auch unter Windows. Nur sollte er zusammen mit Windows nur zu Testzwecken eingesetzt werden. Zum Beispiel wenn Sie mit Perl oder PHP entwickeln, unter Windows, auch NT, 2000 und XP ist das Sicherheitsniveau laut Aussa-

ge der Entwickler aufgrund der Mängel des Betriebssystems zu niedrig. Ferner stehen unter Windows einige Module und somit Funktionen nicht zur Verfügung. [WEB:FASTIX]

4.2.2 **UMN MapServer**

Der UMN MapServer wird als CGI-Applikation im Webserver betrieben. Die Tatsache, dass MapServer unter fast jedem Betriebssystem (UNIX/Linux/WinNT/Win2K) und fast jeder Webserver-Software eingesetzt werden kann, ist ein großer Vorteil gegenüber anderen Produkten. MapServer kann auf Shapefiles, ArcSDE, Oracle Spatial, ArcInfo, MapInfo, DGN, ArcGrid, Erdas Imagine und viele andere Datenformate zugreifen. Neben out-of-the-box Lösungen kann MapServer durch MapScript über PHP, Python, Perl oder Java angepasst werden. [WEB:GIS-WIKI: UMN MAPSERVER]

4.2.3 **PostgreSQL/PostGIS**

PostgreSQL ist ein objektrelationales Datenbankmanagementsystem, das als Open-Source-Programm auf der Seite [WEB:POSTGRESQL.ORG] frei verfügbar ist und ohne Lizenzierung heruntergeladen und benutzt werden darf. PostgreSQL unterstützt die SQL92 und SQL99 Standards und darüber hinaus bietet es eine Reihe eigener Erweiterungen an. Als objektrelationales Datenbanksystem implementiert PostgreSQL die Speicherung nicht atomarer Daten, Vererbung und Objektidentitäten und erlaubt Benutzern, das System um selbstdefinierte Datentypen, Operatoren und Funktionen zu erweitern. Die Unterstützung der referentiellen Integrität und ein fortschrittliches Transaktionsmanagement gehören ebenfalls zu den Leistungsmerkmalen von PostgreSQL, wie die Definition von Triggern und Regeln, mit denen Zugriffe auf Datenbankobjekte gesteuert werden können. [WEB:POSTGRESQL.DE]

PostGIS erweitert das objektrelationale Datenbankmanagementsystem PostgreSQL um GIS-Funktionalitäten, die der OpenGIS Spezifikation „OpenGIS Simple Features Specification For SQL“ folgen. PostGIS wurde von der kanadischen Firma Refrations Research entwickelt und unterliegt der GNU General Public License. Es gibt über 200 Funktionen, mit denen man z.B. räumliche Analysen durchführen oder Geometrie-Objekte bearbeiten und manipulieren kann. [WEB:GIS-WIKI: POSTGIS_TUTORIAL]

Soweit ich am lokalen Windows-Rechner gearbeitet habe, wurde pgAdmin III als Verwaltungsplattform verwendet. Mit dem kann man unter anderem eine Verbindung zum Datenbankserver aufbauen, Datenbanken verwalten, Tabellen darstellen oder auch SQL-Anweisungen ausführen. Auf dem Server *maps.fsv.cvut.cz* wird PostgreSQL mit phpPgAdmin mittels Web-Browser verwaltet.

4.2.4 **PHP**

PHP ist eine serverseitige Skriptsprache mit zahlreichen Anwendungsmöglichkeiten, die speziell für Webanwendungen gedacht ist. Zur weiten Verbreitung von PHP trägt bei, dass es unter einer freien Lizenz steht. Es wird nicht nur von einer Gemeinschaft aus Anwendern und Entwicklern ständig weiter entwickelt, bei der Anwendung für das Erstellen eigener Skripte fallen auch keine Lizenzgebühren an. PHP-Skripte werden auf dem Webserver ausgeführt. Sobald eine PHP-Datei mit einem Browser aufgerufen wird, wird die

Anfrage an den Webserver weitergeleitet. Auf dem Webserver ist ein Parser (Sprachanalytiker) installiert, welcher den PHP-Code interpretiert und nach den Vorgaben dieses Codes eine HTML-Seite auswirft, die an den Client geschickt wird. Im Seiten Quelltext, der sich dem Browser präsentiert, ist kein PHP-Code mehr erkennbar. Die Dateiendung (meist *php*) zeigt dem Webserver, dass es sich um eine PHP-Datei handelt. [WEB:AKADEMIE.DE]

4.2.5 **Mapbender**

Die Software Mapbender ist eine in PHP und JavaScript implementierte Entwicklungsumgebung für Kartenanwendungen. Alle Daten werden zur Laufzeit dynamisch aus einer Datenbank gelesen, so wie das bei Content Management Systemen (CMS) üblich ist. Mapbender ist also im Grunde genommen ein Geodaten CMS und wird deshalb häufig als Geoportalsoftware eingesetzt. Die Software beinhaltet vorkonfigurierte Oberflächen für die Anzeige, Navigation und Abfrage von OGC standardisierten Diensten (z.B. WMS, WFS-T, GML). Zusätzlich stehen in der Administration Module und Oberflächen für die Verwaltung der Kartenwerke (Server), Benutzer, Benutzergruppen und Berechtigungen zur Verfügung. Die Mapbender-Datenbank ermöglicht eine exakte Benutzer- und projektspezifische Protokollierung aller Aktionen, Abfragen und Navigation. Die Software ist kompatibel mit Kartendiensten und Datendiensten, die entsprechende OGC-Schnittstellenspezifikationen implementieren und kann deshalb als Rahmen für eine breite Palette von GIS- und GDI-Softwareprodukten unterschiedlicher Hersteller genutzt werden. Die Palette der unterstützten Software kann bei Bedarf erweitert werden und ermöglicht es, auch herstellerspezifische Besonderheiten zu berücksichtigen. [WEB:MB-WIKI]

Mapbender benötigt für eine fehlerfreie Arbeit folgende Komponenten:

- Webserver, in unserem Fall wird Apache verwendet
- PHP ab Version 5.1.x für PostgreSQL-prepared Statements und PHP 5.2.1 für natives JSON (JavaScript Object Nation)
- SQL-Datenbank für Oberflächen-, WMS- und Benutzerverwaltung. Hierfür kommt PostgreSQL zum Einsatz.

Ein Mapbender an sich beinhaltet weder raumbezogene- noch Sachdaten. Alle Inhalte werden durch WMS, WFS oder andere OGC standardisierte Dienste eingebunden. Deswegen ist es erforderlich, eine WMS fähige Mapdatei zu erstellen.

4.2.6 **Verzeichnisstruktur**

Auf dem Server *maps.fsv.cvut.cz* sind verschiedene Daten abgelegt und dort laufen mehrere Mapserver-Applikationen. Manche Software wird von mehreren Benutzern im Anspruch genommen oder ist für andere Applikationen erforderlich, deswegen sollte der Zugriff geschützt werden. Ein Linux-Server ermöglicht die genaue Behandlung der Zugriffsrechte, was auch beim Erstellen unserer Applikation ausgenutzt worden ist. Grundsätzlich sind die Datenverzeichnisse, die u. a. die lizenzierten Daten enthalten, von den zum Veröffentlichen geeigneten zu trennen. Im unseren Fall geschah es anhand der Konfiguration des Apache Servers. Die Verzeichnisstruktur des Servers ist nach Empfehlungen von [WEB:ČEPIČKÝ] aufgebaut und verfolgt die allgemeinen Empfehlungen für das Linux-

Dateisystem, wie unter [WEB:FREEOS] beschrieben. Im Folgenden sind die Verzeichnisse aufgelistet, die sich auf Nasca Mapserver und seine Funktionalität beziehen.

Linux-Server root-Struktur

/sbin/

Dieses Verzeichnis enthält alle Binärdateien, die wesentlich für das Funktionieren des Systems sind. Dazu gehören Systemadministration sowie Wartung und Konfiguration der Hardware-Programme. Sie finden hier die wesentlichen Programme (lilo, fdisk, init, ifconfig), die für alle Nutzer erforderlich sind. Ein weiteres Verzeichnis, das System-Binärdateien enthält, ist /usr/sbin/. Dieses Verzeichnis enthält andere Binärdateien, deren Nutzungsrechte dem Systemadministrator gehören.

/bin/

Im Gegensatz zu /sbin/, enthält das /bin/ Verzeichnis mehrere nützliche Befehle, die sowohl vom System-Administrator als auch von nicht privilegierten Benutzern verwendet werden. Dieses Verzeichnis enthält in der Regel die Schalen wie bash, csh etc. Es gibt auch ein Verzeichnis /usr/bin/, das Binärdateien von anderen Benutzern enthält.

/data/

_projekty/nasca/

In diesem Verzeichnis wird die Mapdatei nasca_PrD.map gespeichert.

nasca/

Hier werden entsprechend der entworfenen Verzeichnisstruktur die Raster- und Vektordateien gespeichert – siehe ABB. 8.

/etc/

apache2/

Hier befindet sich die zentrale Konfigurationsdatei apache2.conf und weitere conf-Dateien. Diese Dateien sollten nicht geändert werden.

apache2/sites-enabled/

Die verlinkte Datei 000-default verweist auf /etc/apache2/sites-available/default, wo die Benutzer-Anpassungen enthalten sind.

apache2/mods-enabled/

In diesem Verzeichnis werden ausgewählte Module aus etc/apache2/mods-available/ verlinkt.

/home/

dobespet/public_html/

Hier befindet sich eine einfache HTML-Seite index.html von der aus die Verlinkungen auf eine öffentliche und auf eine geschützte Applikation führen. Diese Seite erscheint nach Eingabe der Adresse *http://maps.fsv.cvut.cz/~dobespet* in Adressenzeile.

Linux ist ein Multi-User-System, so dass jedem Nutzer auch ein bestimmtes Verzeichnis zugeordnet ist, das nur ihm und dem System Administrator zugänglich ist. Dies sind die Benutzer-Home-Verzeichnisse, die unter /home/username/ gefunden werden können. Dieses Verzeichnis enthält auch die benutzerspezifischen Einstellungen für Programme wie IRC, X usw.

/lib/

Dieses Verzeichnis enthält alle gemeinsamen Bibliotheken, die vom Systemprogramm benötigt werden. Windows Äquivalent zu einer gemeinsamen Bibliothek wäre eine *dll*-Datei.

/usr/**lib/cgi-bin/**

Hier liegt das UMN MapServer-Programm *mapserv* und seine zweckbedingte Kopie *nasca*

Das Verzeichnis */usr/* ist eines der wichtigsten Verzeichnisse im System, da es alle Benutzer-Binärdateien enthält. X und seine unterstützenden Bibliotheken können hier gefunden werden. User-Programme wie Telnet, FTP usw. sind auch hier untergebracht. */usr/doc/* enthält nützliche Unterlagen. */usr/src/linux/* enthält den Quellcode für den Linux-Kernel. Das Verzeichnis */usr/lib/* enthält die Benutzer-Bibliotheken.

/var/**www/**

Dies ist ein Server-root-Verzeichnis

www/mapbender/

Mapbender-Verzeichnis

Das Verzeichnis */var/* enthält die vermittelten Daten wie Emails oder den Ausgang aus dem Drucker-Daemon. Die System-Protokolle werden in */var/log/messages/* gehalten. Dort befindet sich auch die Datenbank für BIND in */var/named/* und für NIS in */var/yp/*. Im Unterverzeichnis */var/www/mapbender/* werden alle für den Mapbender benötigten Dateien bereitgestellt. Die dafür erforderliche Struktur ist im Folgenden näher erläutert.

Mapbender-Struktur

Ein Mapbender-Verzeichnis beinhaltet eine Vielzahl an Unterverzeichnissen, in denen wiederum Dateien vorliegen, die er für einen vollen Funktionsumfang benötigt. Die Wichtigsten sollen hier erläutert werden. Erläuterungen erfolgen nach [KLEMM 2008].

conf/

Hier befindet sich die zentrale Konfigurationsdatei *mapbender.conf*. An dieser Stelle können bestimmte Einstellungen vorgenommen werden, wie der Zugriff auf die Administrations-Datenbank PostgreSQL erfolgt, wie Fehlermeldungen ausgegeben werden sollen oder wie die Verbindung zu Mapbender aufgebaut wird.

http/

Anwendungsverzeichnis, auf das Mapbender zugreift, um verschiedene Module laden zu können. Diese sind nach entsprechendem Typ (stylesheets, JavaScripts, PHP u.v.m.) getrennt und in Unterverzeichnisse abgelegt.

log/

Hier erfolgt die Speicherung von aufgetretenen Fehlern. Dieses Verzeichnis muss über Schreibrechte verfügen.

owsproxy/

Darüber läuft der Mapbender im gesicherten Modus.

resources/

Hier befinden sich jene *sql*-Dateien, welche die Erstellung der Tabellen in der Administrations-Datenbank ermöglichen und darin Daten einzufügen. Auf diese Tabellen greift Mapbender zu, um die jeweiligen Einstellungen der Nutzer bzw. Administrationsoberflächen zu laden oder Änderungen zu speichern.

tools/

enthält eine Vielzahl von PHP-Skripten, wodurch z.B. die Konfiguration des Mapbenders und einiger Systemkomponenten überprüft werden kann.

MS4W-Struktur

Obwohl der MS4W Pack für die auf dem Server endgültig laufende Applikation nicht verwendet wurde, habe ich mit ihm während der Versuchsphase hauptsächlich gearbeitet. Im Fall des Einsatzes eines Windows-Servers wäre das MS4W Pack nützlich. Daher ist eine Beschreibung der Struktur im ANHANG 2 beigelegt.

4.3 ANFORDERUNGEN AN DIE APPLIKATION

Bevor man überhaupt eine Mapserver-Applikation zu entwickeln anfängt, sind mehrere Aspekte zu berücksichtigen. Es gibt drei wesentliche Seiten. Auf der einen Seite ist der Auftraggeber, seine Vorstellungen, Ideen, Wünsche, Bedingungen und Möglichkeiten. Auf der zweiten Seite ist die Benutzergruppe mit ihren Erwartungen, Bedürfnissen und Kenntnissen. Dann bleibt noch der Gestalter und seine Vorkenntnisse, Invention und technische oder andere Begrenzungen übrig. Im Fall unserer Applikation ist der Kreis der Beteiligten, bzw. das oben genannte Dreieck nicht so breit. Durch eine gelungene Applikation könnte allerdings die Zielgruppe wachsen.

Elementare Funktionen, wie die Anzeige vorhandener Daten, Zoom oder Pan werden stillschweigend vorausgesetzt. Daneben gibt es aber eine Menge Zusatzfunktionen, um die die Applikation erweitert sein kann. Es ist aber immer abzuwägen, welche noch nützlich sind und welche schon überflüssig und damit störend sind. Aus den Vorgesprächen bei der Aufgabenstellung stammen folgende grundlegende Anforderungen, die aber immer nach Bedarf oder nach kommenden Ideen erweitert oder abgeändert werden können.

Die Applikation sollte möglichst alle Ergebnisse des Projekts Nasca darstellen können, wobei beschrieben werden sollte, wie man neue Daten einfügen kann. Als Grundlage sollten ebenfalls möglichst viele verschiedenen Karten und Bilder zur Verfügung stehen, damit man eine Übersicht hat, welche Daten vorhanden sind. Diesbezüglich sind immer die Lizenzbedingungen und Beschränkungen einzuhalten. Die Daten, deren Lizenz keine Veröffentlichung ermöglicht, sollten unter geschütztem Zugang nur für befugte Nutzer stehen. Die Datenstruktur sollte an der HTW Dresden und TU Prag ähnlich sein, damit ein Austausch kompatibel ist. Die Oberfläche sollte möglichst mehrsprachig sein. Da die Applikation letztendlich auf einem tschechischen Server läuft, sollte eine der Sprachen Tschechisch sein. Die elementaren Funktionalitäten werden um brauchbare erweitert. Dazu gehören unter anderem Längen- und Flächenermittlung, Koordinatenanzeige. Die Applikation sollte fähig sein, Rasterdaten, Vektordaten sowie Sachdaten

zu zeigen. Dafür ist auch die Anbindung an eine Datenbank erforderlich und zur Datenbank gehört auch die Suchfunktionalität. Da die Daten unterschiedlichen Umfang, Qualität, Auflösung und Ausführung aufweisen, ist es praktisch unmöglich, alle auf einmal darzustellen. Infolge dessen sind sie in Kategorien (Ebenen) aufzuteilen. Dem Nutzer sollte dann die Möglichkeit zur Verfügung stehen, einzelne Ebenen ein- und auszuschalten und auch nach oben oder nach unten zu verschieben. Damit entsteht eigentlich immer eine originale Darstellung, die den individuellen Erfordernissen möglichst gut angepasst wird. Eine wichtige Frage ist die Ladegeschwindigkeit der Applikation. Deshalb sind alle möglichen Maßnahmen zu gewährleisten, damit die Zeit des Ladevorgangs so weit wie möglich verkürzt wird.

4.4 INSTALLATION UND KONFIGURATION

Meine ersten Schritte beim Erstellen einer Mapserver Applikation sind häufig vom Tutorial [KLEMM 2008] begleitet worden. Dort sind alle relevanten Anleitungen zur Installation und Konfiguration von Apache Webserver, UMN MapServer, PostgreSQL, pgAdmin III, PostGIS und Mapbender schrittweise sehr übersichtlich zusammengefasst. Wenige Anpassungen waren trotzdem erforderlich, aber im Wesentlichen war dieses Tutorial tatsächlich hilfreich. Die wichtigsten Schritte, die ich vorgenommen habe, werden im ANHANG 3, bzw. ANHANG 5 aufgeführt, dabei sind einige Vorgehensweisen und Beschreibungen ohne nochmalige Quellenangabe vom Tutorial [KLEMM 2008] übernommen. Was im Anhang nicht zu finden ist, wird größtenteils im Tutorial beschrieben oder dann auf entsprechenden Produkt-Webseiten. Da es sich um Open-Source-Produkte handelt, werden meistens Dokumentationen beigelegt. Daneben sind Onlinediskussionen oder Mailinglisten eingerichtet worden, wo man mit anderen Benutzern bzw. mit Entwicklern die Probleme ziemlich effektiv und schnell lösen kann. Diese Beratung sollte allerdings erst dann genutzt werden, wenn man sich vergewissert hat, dass es nicht innerhalb der Dokumentation beschrieben wurde. Sonst werden die Diskussionen überbelastet und die Leute, die hier sonst freiwillig und kostenlos beraten, werden unnötig beansprucht.

Hinweis: Die jeweils aktuellste Version der angegebenen Software kann in der Bezeichnung abweichen. Es ist empfehlenswert, immer die aktuellste Version anzuwenden, da sie die meisten Funktionalitäten unterstützt.

4.4.1 MS4W

Obwohl der MS4W Pack für die auf dem Server endgültig laufende Applikation nicht verwendet wurde, habe ich mit ihm während der Versuchsphase hauptsächlich gearbeitet. Im Fall des Einsatzes eines Windows-Servers wäre das MS4W Pack nützlich. Daher ist eine Beschreibung der Installation im ANHANG 3 beigelegt.

4.4.2 Apache, UMN MapServer und Konfigurationsdateien

Eine Installation von Grund auf war auf dem Server *maps.fsv.cvut.cz* nicht erforderlich, da dort manche Komponenten bereits vorhanden gewesen sind. Konkret konnte ich den laufenden Apache Webserver, UMN MapServer und PostgreSQL/PostGIS nutzen. Aller-

dings waren zahlreiche Anpassungen in den Konfigurationsdateien notwendig. Alle Angaben sind im ANHANG 5 enthalten.

Die Anpassungen für einen Linux-Server können von MS4W leicht abweichen, dies ist aber im ANHANG 5 deutlich gemacht. Der größte Unterschied besteht darin, dass MS4W nur eine Konfigurationsdatei `httpd.conf` verwendet, während auf dem Linux-Server verschiedene Bereiche einzeln gespeichert und durch eine relativ schlichte Datei `apache2.conf` mittels *Include*-Angaben verbunden werden. Eine dieser zwei Konfigurationsdateien des Apache wird immer beim Starten des Webservers einmal gelesen. Hierin sind die Einstellungen zur Verwaltung von Apache festgelegt. Deshalb ist es nötig, nach jeder Änderung Apache neu zu starten. Vor jeder Änderung empfiehlt es sich, eine Kopie der funktionierenden Version zu speichern. Die Pfadangaben dürfen entweder absolut oder relativ angegeben werden. Relative Angaben sind im Allgemeinen günstiger, wobei immer aufgepasst werden muss, worauf sich der relative Pfad bezieht. In der Regel ist es entweder der Speicherort der Konfigurationsdatei oder der gegebene Server-root-Pfad. Die Änderungen betreffen folgende Dateien, bzw. Verzeichnisse:

- `/etc/apache2/apache2.conf`
- `/etc/apache2/sites-enabled/000-default` (verweist auf `etc/apache2/sites-available/default`)
- `/etc/apache2/mods-enabled/` (verweist auf `etc/apache2/mods-available/`)
- `/etc/apache2/ports.conf`
- `/ms4w/Apache/conf/httpd.conf`

Beispielsweise erfolgt die Einbindung von zusätzlichen Modulen in MS4W durch das Entfernen von Kommentarzeichen in `httpd.conf` bei der entsprechenden *LoadModule*-Zeile.

```
LoadModule deflate_module modules/mod_deflate.so
LoadModule headers_module modules/mod_headers.so
```

Auf der anderen Seite, in Linux werden in `apache2.conf` die Zeilen

```
Include /etc/apache2/mods-enabled/*.load
Include /etc/apache2/mods-enabled/*.conf
```

eingefügt und des Weiteren sind Verlinkungen auf `mods-available/mod_deflate` und `mods-available/mod_headers` in `mods-enabled/` anzulegen.

Die nächste Anpassung erfolgt identisch sowohl auf dem Linux-Server (in der Datei `default`) als auch in MS4W (in der Datei `httpd.conf`) und hat zwei Gründe. Erstens wird bei jedem *GetCapabilities*-Request ein verkürzter Request verwendet und zweitens ist der Speicherort der Mapdatei vom Browser aus nicht mehr erkennbar.

```
SetEnvIf Request_URI "/cgi-bin/nasca" MS_MAPFILE=/data/_projekty/nasca/nasca_
PrD.map
```

Der verkürzte Request heißt dann:

```
http://maps.fsv.cvut.cz/cgi-bin/nasca?REQUEST=GetCapabilities&SERVICE=WMS&
VERSION=1.1.1
```

4.4.3 Mapfile

Das Mapfile ist die Zentrale vom MapServer. Hier werden die Beziehungen zwischen den Objekten definiert, und dem MapServer gesagt wo er welche Daten findet und wie diese gezeichnet werden. Das Mapfile ist in mehrere Bereiche gegliedert

HEADER

Am Anfang der Datei stehen allgemeine Eigenschaften wie Ausdehnung, Einheiten, Pfad zu Daten, Schriftart- und Symboldateibestimmung, Projektion, Ausgabeformat.

WEB

Der WEB-Bereich ist erforderlich, wenn die Mapdatei als WMS Dienst verwendet werden soll. Hier sind Metadaten des WMS Dienstes enthalten.

REFERENCE, SCALE, LEGEND, QUERYMAP

Diese Bereiche erstellen der Reihe nach eine Übersichtskarte, Maßstabsleiste, Legende und Ergebniskarte auf Nachfrage. Auf alle diese Bereiche könnte in einer Mapdatei für Nasca Mapserver verzichtet werden, da die dementsprechenden Elemente durch den Mapbender selbst dynamisch erstellt werden.

LAYER

Es ist meistens der umfangreichste, sowie der wichtigste Bereich der Mapdatei. Jede Datei, die angezeigt werden soll, bekommt hier eine eigene <LAYER> Angabe, wo die dazugehörigen Sachverhalte definiert werden. Unter anderem sind dies Name der Ebene, Dateiname, Datenart, Name der Ebene in einem WMS Dienst und Projektion.

Es gibt einige wichtige Sachverhalte die für eine zuverlässige Konfiguration des MapServers durch Mapfiles verstanden werden müssen. Der Sachverhalt wird durch Layer beschrieben. Ein Layer ist die Kombination von Daten und ihrer Ausprägung (Styling). Daten, in der Form von Attributen und Geometrie, können durch die Benutzung von <CLASS> und <STYLE> Direktiven dargestellt werden. [WEB:MAPSERVER]

Auf eine genauere Betrachtung, wie den Aufbau und die Möglichkeiten des Mapfiles, wird hier verzichtet und es wird dabei entweder aufs Tutorial [KLEMM 2008] oder auf [WEB:MAPSERVER] verwiesen.

Das für die Applikation verwandte Mapfile `nasca_PrD.map` basierte erstmal auf dem ursprünglichen Mapfile von Nazca MapServer (LAYER-Bereich) und die Struktur der WEB-Sektion wurde von Nasca WebGIS übernommen. Im Endeffekt waren aber so viele Änderungen nötig, dass ein vollkommen neues Mapfile entstanden ist. Das Mapfile `nasca_PrD.map` ist im ANHANG 1 enthalten.

4.4.4 PostgreSQL/PostGIS

Auf dem Server `maps.fsv.cvut.cz`, wo PostgreSQL bereits vorhanden ist, erfolgt die Verwaltung über einen Web-Browser (<http://maps.fsv.cvut.cz/pgadmin>). Eine neue Datenbank lässt sich in phpPgAdmin ziemlich einfach anlegen. Für Mapbender wurde eine Datenbank mit folgenden Angaben angelegt:

```
rolename   nasca
user       mapbender
password   *****
port       5432
host       localhost
Codierung  UTF-8
```

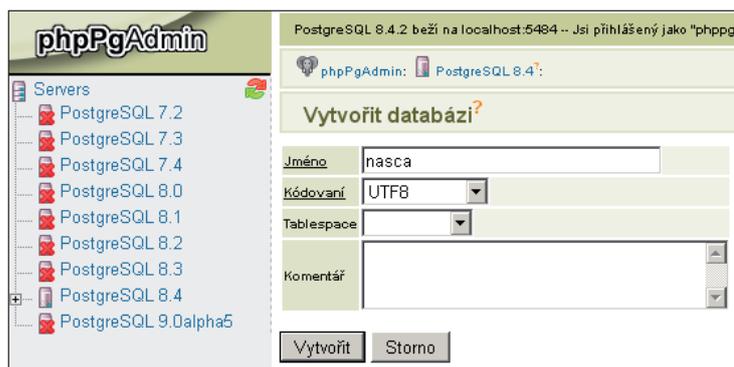


ABB. 11
phpPgAdmin – Anlegen
einer Datenbank

Die Installation, wie ich sie am lokalen Windows-Rechner durchgeführt habe, ist im ANHANG 4 beschrieben.

4.4.5 Mapbender

In Kleinigkeiten unterscheiden sich die Varianten für MS4W und für den Linux-Server. Eine aktuelle Version ist in der Regel unter http://www.mapbender.org/Version_History zu finden. Hier wurde für den Linux-Server die Datei `mapbender_2.5.2.zip` verwendet. Dieses Archiv ist am besten in `ms4w/Apache/htdocs` bzw. in `var/www` zu entpacken, um den Zugriff über das WWW zu ermöglichen. Eine vorbereitete Version für MS4W wäre unter <http://www.maptools.org/ms4w/index.phtml?page=downloads.html>, Datei `mapbender_2.5.2_ms4w.zip`, herunterzuladen

Die zahlreichen Einstellungen für den Apache Webserver, die Mapbender-Datenbank und Angaben für den Mapbender selbst werden, dem Betriebssystem entsprechend, im ANHANG 5 aufgeführt. Die Änderungen betreffen folgende Dateien, bzw. Verzeichnisse:

- `/etc/apache2/apache2.conf`
- `/etc/apache2/sites-enabled/000-default` (verweist auf `etc/apache2/sites-available/default`)
- `/etc/apache2/mods-enabled/` (verweist auf `etc/apache2/mods-available/`)
- `/ms4w/Apache/conf/httpd.conf`
- `/ms4w/Apache/cgi-bin/php.ini`
- `mapbender/conf/mapbender.conf`

In der `php.ini` werden Einstellungen bezüglich Fehlermeldungen, Mapbender-Authorisierung oder benötigter Module vorgenommen. Die `mapbender.conf` ist eine Konfigurationsdatei des Mapbenders befindet sich in dem Ordner `conf` unterhalb des Mapbender-Verzeichnisses.

Anlegen der Mapbender-Datenbank

In die Datenbank `nasca`, die in PostgreSQL mithilfe phpPgAdmin angelegt wurde, muss jetzt Schema (Tabellen, die eindeutigen Keys sowie Indexes) und die Daten eingespielt werden. Es erfolgt anhand vorgefertigter SQL-Skripte. Die Skripte befinden sich in dem Ordner `resources/db/postgresql/` unterhalb des Mapbender-Verzeichnisses.

- | | |
|-----------------------------|---|
| Laden des Schemas | <code>pgsql_schema.sql</code> |
| Laden der Daten | <code>utf8/pgsql_data.sql</code> |
| Setzen der Sequenzen | <code>pgsql_serial_set_sequences.sql</code> |

Prüfung der Konfiguration

Mapbender unterstützt die Anpassungen in der `mapbender.conf`, indem es mit einem Skript `mapbender_setup.php` geprüft wird. Das Skript befindet sich in dem Ordner `tools` und ist über den Browser mit der URL `http://localhost/mapbender/tools/mapbender_setup.php` ausführbar. Die jeweiligen Hinweise sind zu beachten. Die Angabe der Adressen kann bei entsprechend definierten Aliassen abweichen.

4.5 AUFBAU EINER BENUTZEROBERFLÄCHE (GUI)

Der Mapbender lässt sich über einen Web-Browser steuern (`http://maps.fsv.cvut.cz/mapbender`). Die Erstanmeldung nach der Installation erfolgt mit Namen `root` und Passwort `root`. Der Benutzer `root` besitzt alle Rechte eines Administrators und kann andere Benutzer anlegen und organisieren. Dazu ist er als einziger berechtigt, soweit er nicht die Admin-Rechte auf andere Benutzer überträgt. Grundsätzlich sind in Mapbender diese drei in Beziehung stehenden Objekte zu unterscheiden:

- Benutzer und Gruppen
- Oberflächen (GUI, wird neu als Applikation bezeichnet)
- Karten- / Geometriedienste OGC (WMS/WFS)

Die Beziehungen können auch nur vom Admin definiert und geändert werden.

In unserer Applikation ist ein Admin definiert (`root`) und zwei andere Benutzer (`vip`, `guest`). Diesen zwei Benutzern werden dann einzelne GUIs und WMS Dienste zugeteilt. Diesen Benutzern wird auch keine Verwaltung ermöglicht, sondern es ist ihnen lediglich gestattet die zugeordneten GUIs anzuschauen und zu durchsuchen.

Nach Einloggen in Mapbender sieht man eine Liste von vordefinierten Oberflächen. Ihre Bedeutungen sind an dieser Stelle allerdings nicht gleich und man könnte sie in drei Gruppen untergliedern.

Mapbender GUI list	
GUI name	GUI description
<code>admin_de_services</code>	erweiterte admin GUI - WMS, WFS, Metadaten (deutsch)
<code>admin_en_services</code>	extended admin GUI - WMS, WFS, metadata-handling (english)
<code>admin1</code>	admin GUI containing all administrative modules (use only as a template)
<code>admin2_de</code>	admin GUI in deutscher Sprache
<code>admin2_en</code>	admin GUI in english
<code>gui</code>	GUI with tab, search modules
<code>gui_digitize</code>	GUI with WFS search and digitizing using WFS-T
<code>gui1</code>	GUI combining most of the Mapbender functionality
<code>gui2</code>	GUI with focus on layout
<code>NascaMapserver_guest</code>	Nasca Mapserver - GUI für Gäste ohne Login
<code>NascaMapserver_vip</code>	Nasca Mapserver - GUI für Gäste mit VIP Login
<code>wms_africa</code>	Container for Africa WMS
<code>wms_australia</code>	Container for Australia WMS
<code>wms_europe</code>	Container for european WMS
<code>wms_gdi_de</code>	Container for german WMS from GDI-DE
<code>wms_germany</code>	Container for german WMS
<code>wms_north_america</code>	Container for north America WMS
<code>wms_worldwide</code>	Container for world wide WMS

ABB. 12 Mapbender – GUI Liste

GUIs für Verwaltung (Admin GUIs)

- admin_de_services
- admin_en_services
- admin1
- admin2_de
- admin2_en

Die GUI admin1 enthält nur die einzelnen Administrationsmodule und ist für uns nicht von Bedeutung. Eine der letzten zwei GUIs (deutsch- oder englischsprachig) sind dagegen für normale Gestalter einer Applikation die Wichtigsten. Sie beinhalten die Modulgruppen WMS Verwaltung, WMS Zuordnung, GUI-Verwaltung, Benutzerverwaltung, Benutzerzugriff erteilen. Ich habe ausschließlich den admin2_de benutzt und darauf beziehen sich auch alle folgenden Erläuterungen und Vorgehensweisen. Die ersten zwei (in deutscher oder englischer Sprache) sind erweiterte Admin GUIs, sie ermöglichen zusätzlich die WFS Verwaltung und WMS Metadaten zu bearbeiten. Diese wurden nur beim Einbauen der Suchfunktion in Anspruch genommen.

Tatsächliche GUIs für die Benutzer

- gui
- gui_digitize
- gui1
- gui2

In Mapbender liegen bereits vier vorgefertigte GUIs vor. Die erste vordefinierte Oberfläche gui enthält mehrere mögliche Suchfunktionen, die Oberfläche gui_digitize ist mit WFS Suchfunktion und Digitalisierungswerkzeug mithilfe WFS-T (WFS mit Transaktionsfähigkeit) ausgestattet. Die Nächste Oberfläche gui1 beherrscht die meisten Mapbender-Funktionen und die letzte gui2 ist auf Layout orientiert. Zu diesen vier GUIs kommen dann die vom Benutzer neu definierten Oberflächen hinzu.

Unterstützende WMS GUIs

- wms_africa
- wms_australia
- wms_europe
- wms_gdi_de
- wms_germany
- wms_north_america
- wms_worldwide

Der Restlichen sind an sich keine Oberflächen, sondern nur vordefinierte WMS Dienste, die in tatsächlichen GUIs eingebaut werden können.

Es ist praktisch nicht nötig, eine neue GUI von Anfang an zu erstellen, sondern es ist grundsätzlich eine vorhandene zu kopieren und zu bearbeiten. Es ist nicht so wichtig, welche Oberfläche man am Anfang nimmt, da die fehlenden Elemente von den anderen übernommen werden und die übrigen wiederum gelöscht bzw. deaktiviert werden können. Obwohl es im Endeffekt gar nicht zu merken ist, habe ich als Grundlage die gui2 ver-

wendet, die ich jedoch später so geändert habe dass ich eher die gui wegen Tabs und Buttons oder gui1 wegen Funktionen hätte nehmen können.

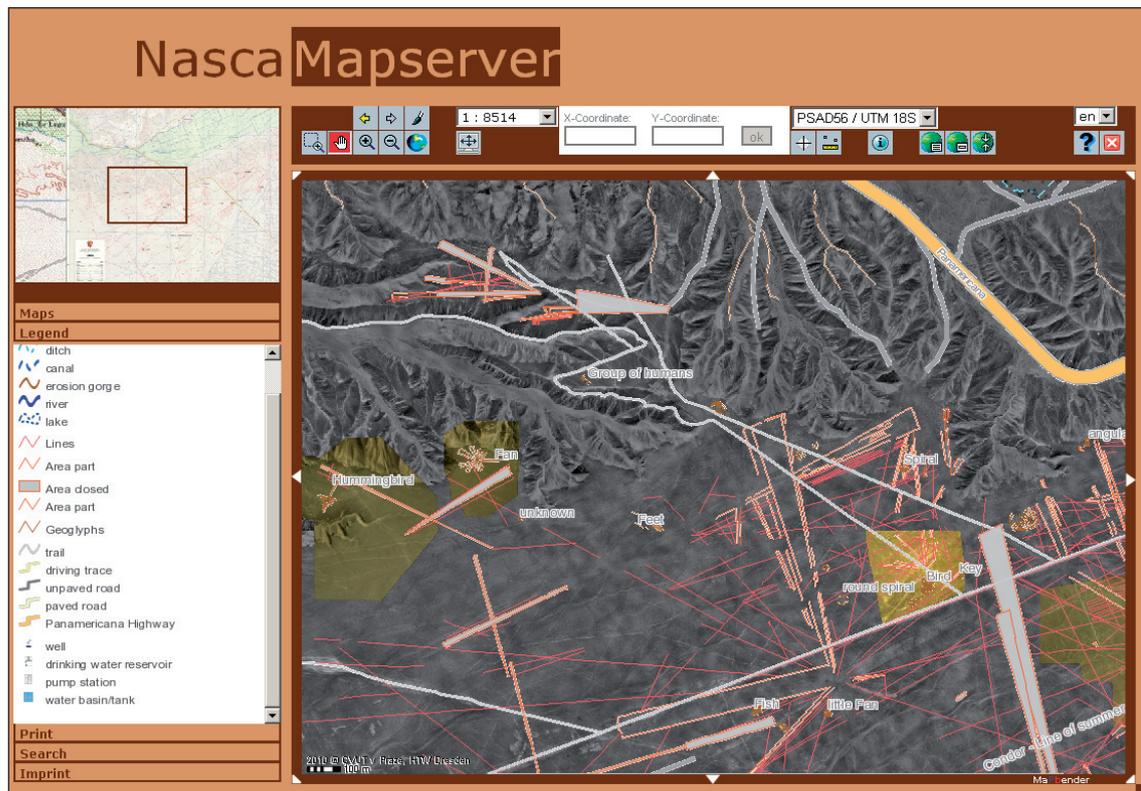


ABB. 13 GUI NascaMapserver_guest

4.5.1 Farbgestaltung

Was vielleicht nebensächlich scheint, ist die Farbgestaltung der GUI. Ich halte es doch für wichtig, weil der Benutzer sie als erstes sieht und im Unterbewusstsein bewertet. Wenn die Umgebung angenehm ist, stört das auch den Benutzer nicht und er besucht die Seiten wieder. Auch wenn man einzelne Elemente platziert, müssen sie dem Hintergrund entsprechen.

Aus oben genannten Gründen habe ich gleich am Anfang die essentielle Farbe gewählt. Bei der Wahl spielte die bestehende Applikation der HTW die größte Rolle. Und es ist auch sinnvoll, wenn in der Regel zu Nasca die braune bis braun-graue Farbe gehört, diese einheitliche Farbgebung beizubehalten. Deshalb habe ich die Farben, wie in (TAB02-FARBEN) aufgeführt, als tragende gewählt. Einzelne Farben sind für Web in RGB Modus und Hexadezimal bezeichnet. Die Farben sind so gewählt, dass sie in jedem Browser gleich angezeigt werden, da sie aus der Netscape Palette (auch als Websichere Farben bekannt) herkommen. Nähere Betrachtungen unter [WEB:WEBTVORBA].

#FFFFFF (weiß)	#CC9966 (helles braun)	#663300 (dunkles braun)	#000000 (schwarz)
-------------------	---------------------------	----------------------------	----------------------

TAB. 2 Farben in Nasca Mapserver

Die Farbangabe erfolgt in Mapbender an vielen Stellen. Teilweise in GUI-Elemente editieren als Style Angabe oder Content Angabe, weiter in Variablen der Funktionen und manchmal direkt in Dateien der Funktionen. Deshalb ist es ziemlich aufwendig, die Farbgestaltung im Ganzen zu ändern. Es ist zu empfehlen, einmal die Farben gut zu durchdenken und sie anschließend überall wo sie vorkommen, auch einzuhalten.

4.5.2 Elemente

Zunächst ist mittels GUI umbenennen/kopieren eine Basisoberfläche unter neuer Benennung zu duplizieren und folgendes Modul GUI-Elemente editieren auszuwählen. Wählen wir jetzt aus der Liste die neu entstandene Oberfläche, erscheint rechts die Liste der bereits eingebundenen Elemente. Wählt man jetzt ein Element (durch Anklicken der Schaltfläche), werden links Parameter in entsprechenden Feldern eingeblendet. Über den Parameterfeldern befinden sich vier Schaltflächen mit folgenden Funktionen:

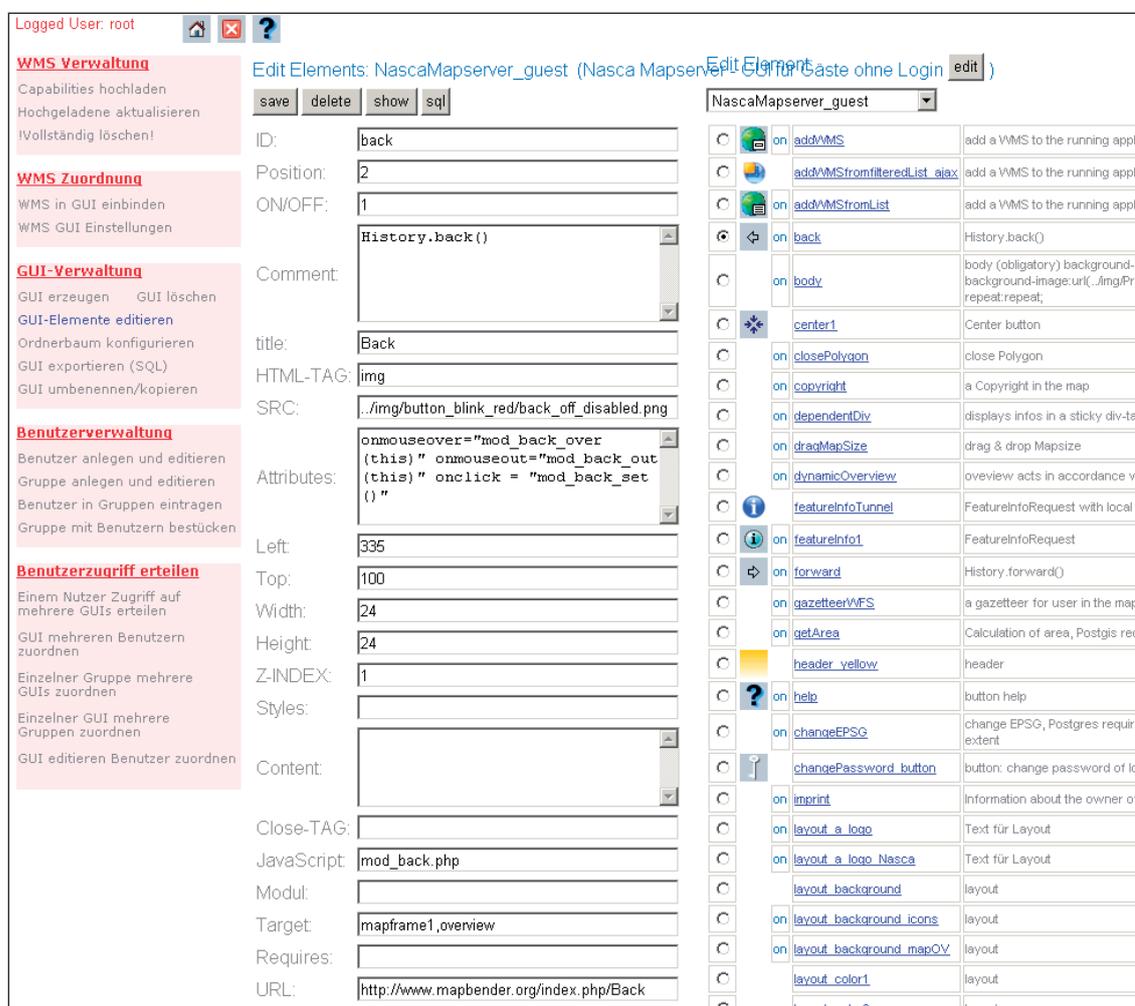


ABB. 14 Mapbender – Modul GUI-Elemente editieren

Schaltflächen

SAVE Speichert alle Änderungen an Parametern des ausgewählten Elements; wird die ID geändert, wird die ganze Angabe als ein neues Element gespeichert und ein altes bleibt unverändert, die ID muss eindeutig sein.

DELETE Löscht das ausgewählte Element.

SHOW Zeigt im neuen Fenster die gesamte GUI (vorher müssen Änderungen gespeichert werden, damit sie wirksam sind).

SQL Hiermit wird ein SQL-Befehl angezeigt, der bei Eingabe in der Datenbank ein entsprechendes Element erzeugt.

Die in der Applikation Nasca Mapserver verwendeten Elemente

Elementname	Beschreibung
 addWMS	Fügt einen WMS Dienst in die laufende Applikation ein.
 addWMSfromList	Fügt einen WMS Dienst aus einer Liste in die laufende Applikation ein.
 back	Damit lassen sich die letzten Kartenausschnitte wieder aufrufen unter Berücksichtigung der Ebenen und Kartengröße. Das Mapframe und der Overview werden angesprochen.
body	Ist die Hintergrundfläche des Mapbenders. Die Farbe kann über einen Mausklick auf den Namen in der Klasse <code>css_class_bg</code> geändert werden.
closePolygon	Dieses Element wird mit zwei weiteren Elementen im Zusammenhang über die Schaltfläche <code>measure</code> aktiviert und im Kartenfenster des Mapbenders angezeigt. Es lässt sich nun eine Fläche digitalisieren, die mit <code>closePolygon</code> den letzten Stützpunkt mit dem Startpunkt verbindet und schließt.
copyright	Das dahinter liegende Modul <code>mod_copyright.php</code> kann mit diesem Element editiert werden um Copyright-Hinweise zu integrieren. Auch hier ist die Einstellung wieder über einen Mausklick auf den Namen veränderbar.
dependentDiv	Hierüber wird die Position eines Textes in Abhängigkeit von einem anderen Element bestimmt.
dragMapSize	Rechts unterhalb des Kartenfensters ist ein Quadrat sichtbar. Damit kann durch einen Klick und ziehen des Quadrates die Fenstergröße verändert werden.
dynamicOverview	overview richtet sich nach dem Kartenfenster.
 featureInfo1	Damit wird ein <code>getFeatureInfo</code> -Request mit den jeweiligen Abfrageergebnissen an den entsprechenden Server geschickt.
 forward	Ähnlich der Browserfunktion können mit diesem Button Kartenausschnitte wieder hergestellt werden.
gazetteerWFS	Es implementiert einen Suchdialog für einen WFS gazetteer. Es ist möglich, mehrere WFS-Suchen einzubinden und diese aus einer Auswahllbox zu selektieren.

Elementname	Beschreibung
getArea	Diese Schaltfläche wird in Verbindung mit <code>measure</code> sichtbar. Nach der Digitalisierung einer Fläche kann mittels <code>getArea</code> die Größe und der Umfang der Fläche berechnet werden.
 help	Hilfe-Button
changeEPSG	Wechselt zwischen Koordinatensystemen.
imprint	Enthält Informationen über den Besitzer einer GUI.
layout	Mit diesem Element ist es möglich, grafische Gestaltungen vorzunehmen (Einfügen von Bildern, farbigen Flächen u.v.m.).
legend	Angaben zur Legende.
loadData	Das ist der Container um Daten in die GUI zu laden.
 logout	Abmelden des aktiven Benutzers.
mapbender	Im Rahmen des Kartenfensters befindet sich ein Schriftzug des Mapbenders. Damit wird darauf hingewiesen, um welche Software es sich handelt. Der Text ist mit einem Link zur Mapbender-Seite versehen.
mapframe1	Es ist ein Zielobjekt zahlreicher Module und bildet den Rahmen für den Kartenausschnitt.
 measure	Über diesen Button kann in der Karte eine Linie, ein Punkt oder Polygon abgesteckt und anschließend mit <code>getArea</code> berechnet werden.
navFrame_grey	Das ist das Navigationsfenster und gehört zu <code>mapframe</code> . Damit ist es möglich die Karte mit einem Klick auf die Pfeile in alle Himmelsrichtungen zu verschieben. Die Farbe ist über den Parameter <code>content</code> konfigurierbar.
overview	Der <code>overview</code> ist eine kleine Übersichtskarte in der GUI. In ihr ist es möglich, in oder aus einer Karte zu zoomen. Das Overview-Fenster verweist immer auf den eingebundenen Dienst und wird als Übersichtskarte verwendet.
 pan1	Mittels <code>pan</code> kann der Kartenausschnitt manuell per Mausklick in die Karte und Drag & Drop verschoben werden.
printPDF	Ermöglicht den Druck in <i>pdf</i> .
 repaint	Mit dieser Funktion wird bei Mausklick auf die Schaltfläche der Kartenausschnitt neu gezeichnet. Es ist ein optionaler Parameter, da einige Module (Ebenen an-/ausschalten) das Kartenfenster selbst immer neu zeichnen.
 resizeMapsize	Dieser Button erlaubt eine automatische Anpassung des Kartenfensters an die Bildschirmgröße.
rubber	Die Funktion <code>measure</code> wird über den <code>rubber</code> zurückgesetzt und löscht die gemessene Strecke im Kartenfenster.
sandclock	Hier erfolgt die Anzeige des Ladezustandes. Es kann ein animiertes GIF eingebunden werden.
scalebar	Betrifft die Anzeige der Maßstabsleiste und passt sich der gewählten Zoomstufe an.

Elementname	Beschreibung
scaleSelect	In dieser Auswahlliste stehen verschiedene Maßstäbe zur Verfügung. Nach der Wahl eines Maßstabes wird automatisch auf diese Stufe gezoomt.
 selArea1	Über die Schaltfläche selArea kann mit der linken Maustaste ein Viereck in der Karte aufgezoogen werden. Anschließend erfolgt das Zoomen auf den gewählten Bereich.
setBBOX	Hier werden Einstellungen zur Ausdehnung des Kartenfensters und der Übersichtskarte vorgenommen.
 showCoords_div	Damit lassen sich die Koordinaten eines gewählten Punktes in der Karte anzeigen.
switchLocale_noreload	Wechselt zwischen Sprachen für die Oberfläche. Beim Wechsel wird die GUI nicht neu geladen.
tabs	Über dieses Modul können weitere Module in Form eines vertikalen Karteireiters aufgerufen und angezeigt werden.
treeGDE	Hierbei lässt sich die Baumstruktur des WMS hinsichtlich Größe, Farbe und Verhalten ändern.
wfs	Ermöglicht eine WFS Abfrage und behandelt das Ergebnis.
 WMS_preferences	Hierdurch können Voreinstellungen für einzelne eingebundene WMS Dienste konfiguriert werden.
zoomCoords	Der Zoomausschnitt wird um die, durch Koordinaten definierte Position zentriert.
 zoomFull	Der Kartenausschnitt wird wieder auf die ursprüngliche, im WMS definierte Ausdehnung gezoomt.
 zoomIn1	Ermöglicht das Hineinzoomen in den Kartenausschnitt.
 zoomOut1	Ermöglicht das Herauszoomen aus der Karte.

TAB. 3 Elemente in Nasca Mapserver

Die Parameter eines Moduls

ID	Hier wird der Name des Moduls eingetragen und sollte eindeutig sein. Keine Verwendung von Sonderzeichen.
POSITION	Mit der Position wird das Ladeverhalten des Mapbenders beeinflusst. Sie muss höher sein, als die des Moduls body.
ON/OFF	0 Das Modul ist deaktiviert 1 Das Modul ist aktiv
COMMENT	eine kurze Beschreibung des Moduls
TITLE	Tool-Tips für einzelne Schaltflächen
HTML-TAG	Ein öffnender HTML-Tag, worüber das Modul in die Anwendung integriert werden kann. z.B. <i>img</i> , <i>div</i>
SRC	Kann ein Pfad zu einem Bild sein. (z.B. Bilder, die als Buttons eingebunden werden)
ATTRIBUTES	Hier sind die Attribute des HTML-Tag's enthalten.

LEFT	Pixelangabe für die linke Position des Elementes in der GUI
TOP	Pixelangabe für die obere Position des Elementes in der GUI
WIDTH	Breite des Elementes in Pixel
HEIGHT	Höhe des Elementes in Pixel
Z-INDEX	Darstellungsreihenfolge; Elemente mit höherem Z-Index befinden sich über Elementen mit niedrigerem Z-Index
STYLES	zusätzliche Parameter für Stylanaben
CONTENT	der Inhalt eines Elementes, welches in der GUI angezeigt wird
CLOSE-TAG	ein schließender Tag des HTML-Elementes
JAVASCRIPT	Referenz auf einer JavaScript-Datei, mit entsprechender Funktionalität
MODUL	Dies ist ein JavaScript, das mehrere Module benutzen aber nur einmal geladen werden soll.
TARGET	das entsprechende Ziel eines Moduls wo die Parameter gelten sollen. Beispielsweise measure hat als Ziel das mapframe1
REQUIRES	Nennt das Modul, welches von diesem Modul abhängig ist.
URL	Link zu der Beschreibung des Moduls auf der Internetseite des Mapbenders Erläuterung der Elemente sowie der Parameter erfolgte nach [KLEMM 2008].

Variable der Elemente

Liegen irgendwelche Variablen jeweiliger Elemente vor, sind diese durch Mausklicks auf die Elementnamen (Verlinkung) abrufbar und einstellbar.

The screenshot shows the 'Edit Element' interface in Mapbender. At the top, there is a dropdown menu for the element name, currently set to 'NascaMapserver_guest'. Below this, there are three rows of JavaScript modules, each with a radio button, a name, and a description: 'showCoords_div' (displays coordinates by onmouseover), 'switchLocale_noreload' (changes the locale without reloading the client), and 'tabs' (vertical tabs to handle iframes). The middle section is titled 'Edit Element Vars: NascaMapserver_guest / tabs' and contains 'save', 'delete', and 'return' buttons. Below the buttons, there are four input fields: 'Name' (containing 'tab_frameHeight[1]'), 'Value' (containing '385'), 'Context' (empty), and 'Type' (containing 'var'). On the right side, there is a list of variables with radio buttons next to them: 'open_tab', 'tab_frameHeight[0]', 'tab_frameHeight[1]' (selected), 'tab_frameHeight[2]', 'tab_frameHeight[3]', 'tab_frameHeight[4]', 'tab_ids[0]', 'tab_ids[1]', 'tab_ids[2]', and 'tab_ids[3]'. Each variable has a type indicator to its right, such as 'var' or 'php_var'.

ABB. 15 Mapbender –
Variable der Elemente editieren

Platzierung der Elemente

Die Platzierung einzelner sichtbarer Elemente erfolgt, wie aus der Beschreibung der Parameter erkennbar ist, durch Pixelangabe der linken oberen Ecke des Elements (Parameter LEFT und TOP). Damit mehrere Elemente auf einmal verschoben werden können, ist ein SQL-Befehl in der Datenbank auszuführen. Folgender SQL-Befehl verschiebt alle Ele-

mente, die 215 Pixel unterhalb der Oberkante des Fensters positioniert sind um weitere 60 Pixel nach unten.

```
UPDATE gui_element SET e_top=e_top+60
WHERE fkey_gui_id='name_der_gui' AND e_top >= 215;
```

Sprachauswahl

Mittels des Elements `SwitchLocale_noload` können dem Benutzer mehrere Sprachversionen angeboten werden. Bisher wurde Mapbender in ungefähr 6 Sprachen übersetzt, wovon Englisch als Hauptsprache, Deutsch und Spanisch für die Applikation implementiert wurde. Sobald die tschechische Version veröffentlicht wird, kann sie zusätzlich auch implementiert werden. Durch Sprachänderung wird nur die Beschreibung der Elemente beeinflusst, nicht die Inhalte.

4.5.3 Dateneinbindung

Um Inhalt in einen Kartenrahmen zu bringen, sind erstens WMS Capabilities hochzuladen, eventuell später nur zu aktualisieren (unter WMS Verwaltung). Beim Hochladen habe ich insgesamt zwei Adressen eingegeben.

Mittels dieser Anfrage habe ich alle in der `nasca_PrD.map` Datei aufgeführten Ebenen eingeladen: `http://maps.fsv.cvut.cz/cgi-bin/nasca?REQUEST=GetCapabilities&SERVICE=WMS&VERSION=1.1.1`

Hiermit wurden Ebenen aus dem bestehenden WMS Kartendienst Pampa of Nasca hochgeladen: `http://141.56.141.2:8080/cgi-bin/mapserv?REQUEST=GetCapabilities&service=WMS&version=1.1.1`

Schalter

Folgend sind Eigenschaften der WMS für einzelne GUIs zu definieren und zuzuordnen. Dies erfolgt über WMS GUI Einstellungen. Jeweilige Schalter und Spalten mit ihren Bedeutungen sind umfangreich in [CHRISTL, EMDE 2008] beschrieben. Hier wird nur auf die für die Applikation Wichtigen verkürzt eingegangen.

- | | |
|------------------------|---|
| EPSG | Über diese Auswahl kann das gewünschte Koordinatensystem, mit dem die GUI startet, selektiert werden. |
| Mapformat | Hier bestimmt man das Ausgabeformat, mit dem der Dienst seine Kartenbilder generieren soll. Bevorzugt werden die Formate <i>png</i> und <i>gif</i> , da sie auch Transparenz unterstützen. |
| Infoformat | Das Infoformat definiert, wie der Dienst alphanumerische Daten an den Client zurückliefern soll. Es sollte die <i>text/html</i> ausgewählt sein. |
| Exceptionformat | Hier wird definiert, in welchem Format Fehlermeldungen zurückgeliefert werden sollen. Allerdings erwartet Mapbender ein Bild, sodass es sinnvoll ist, das Format <i>inimage</i> oder <i>application/vnd.ogc.se_inimage</i> auszuwählen. Wenn der Dienst fehlerhaft ist, so erfolgen die Fehlermeldungen als Bild in der angeforderten Größe im Mapbender-Kartenfenster. |
| Visibility | Mit Visibility ist die Sichtbarkeit des Dienstes auf sichtbar (<i>visible</i>) oder versteckt (<i>hidden</i>) einzustellen. |

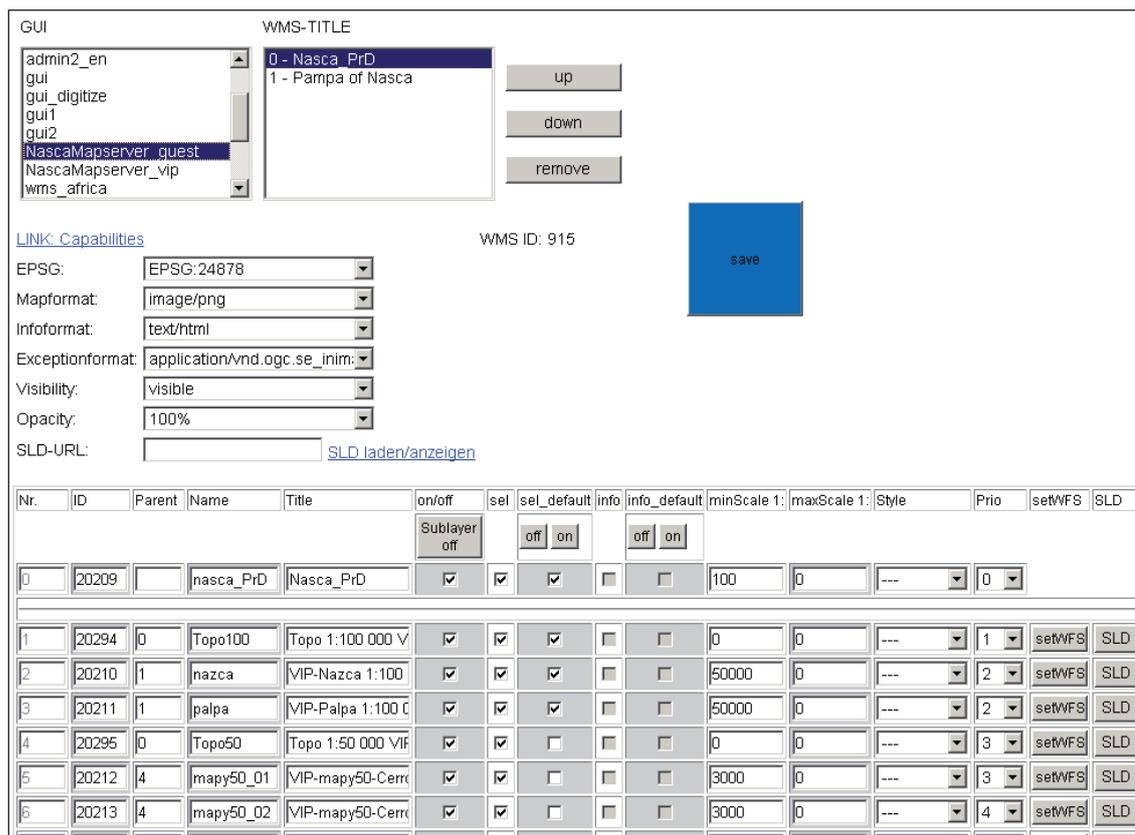


ABB. 16 Mapbender – WMS GUI Einstellungen

Ebentabelle

Allgemein sind sämtliche Ebenen des Dienstes durchnummeriert. Bei Neuladen (Aktualisieren) des Datendienstes kann die Nummerierung eigenmächtig geändert werden. Die Zahl in der Spalte Nr. ist aufsteigend und beginnt mit dem Wert 0. Die erste Tabellenzeile entspricht dem gesamten Kartendienst (Nr. = 0) und sollte nicht verändert werden.

Nr.	Parent	Ebene
0		E 0
1	0	E 11
2	1	E 111
3	1	E 112
4	0	E 12
5	4	E 121
6	0	E 13
7	6	E 131
8	7	E 1311
9	6	E 132

TAB. 4 Parent-Angabe in Ebenenbaum

Parent	Die Zahl in dieser Spalte verweist auf die Nr. der übergeordneten Ebene. Ist die Zahl gleich 0 so bedeutet dies, dass es eine eigenständige Ebene ist, die nur dem gesamten Kartendienst untergeordnet ist. Wenn der Wert größer 0 ist, deutet das auf einen gruppierten Layer hin. Die Zahl verweist in diesem Fall wieder auf jene Ebene, der die gruppierte Ebene unterliegt. Änderungen sind lediglich direkt in der Datenbank (z.B. über phpPgAdmin) möglich. Ein Beispiel in der TAB. 4.
Name	In dieser Spalte ist der Name zu sehen, der im LAYER-Bereich unter <TITLE> in dem <i>GetCapabilities</i> -Dokument des Dienstes eingetragen ist.
on/off	Mit einem Check-Button lässt sich individuell einstellen, ob die Ebenen des Dienstes sichtbar sein sollen oder nicht.
sel	Mit einem Häkchen kann voreingestellt werden, dass der Benutzer den Layer an- und ausschalten darf. Ohne ein Häkchen ist der entsprechende Layer fixiert.
sel_default	Die Check-Buttons in dieser Spalte definieren, welche Ebenen zum Start des Mapbenders aktiviert sein sollen.
info, info_default	In der Spalte info lässt sich angeben, welche Layer für die <i>FeatureInfo</i> -Requests an- bzw. ausgestellt werden sollen. info_default gibt nur die Ebenen an, die bereits beim Start des Mapbenders für die Abfragen aktiv sein sollen.
minScale, maxScale	Diese Spalten erlauben es, den größten und kleinsten Maßstab einzutragen in denen die entsprechenden Ebenen sichtbar sein sollen. Wenn die Sichtbarkeit auf keinen bestimmten Maßstabsbereich eingeschränkt werden soll, muss hier jeweils eine 0 stehen. Dennoch haben die Einstellungen des Mapfiles höhere Priorität. Diese Werte sollten taktisch sinnvoll eingestellt werden. Dabei spielen die Pixelgröße und die Genauigkeit eine Rolle. Die nicht genau georeferenzierten Bilder sollten nicht im Detail angezeigt werden.
Prio	Die Einstellungsmöglichkeiten in der Spalte Prio beziehen sich auf die Reihenfolge der einzelnen Layer. Der Layer mit dem kleinsten Wert wird zuerst angefordert und liegt in der Karte auf der untersten Ebene. Doppelte Werte müssen vermieden werden, da die Reihenfolge bei gleichen Werten, abhängig von der verwendeten Datenbank, alphabetisch oder in der Reihenfolge des WMS Dienstes vergeben wird.
Style	In dieser Auswahlliste ist es möglich die erstellten SLD-Styles (Styled Layer Descriptor) für den jeweiligen Layer zu laden. Die entsprechenden Styles sind mit dem SLD-Editor herstellbar und zu speichern. Wenn mehrere Styles für einen Layer erstellt wurden, sind diese in der Auswahl verfügbar.
setWFS, SLD	Über die Schaltfläche setWFS können die Feature Types eines WFS Dienstes einem WMS Layer zugeordnet werden. Über die Schaltfläche SLD wird ein Editor in einem neuen Browser-Tab geöffnet. Jetzt können die bisher definierten Style-Angaben in dem Mapfile neu definiert werden.

Rasterdaten

Die Einbindung der Rasterdaten erfolgt in Mapbender ausschließlich durch eine Mapdatei, die als WMS Dienst hochgeladen wird. Damit das Bild in einer Applikation richtig erscheint, muss das Koordinatensystem (Angabe unter <PROJECTION> in Mapdatei) und die Georeferenz (*tfw*-Datei für *tif*-Bilder) angegeben werden.

Wie in *nasca_dateiliste.xls* (siehe ANHANG 1) zu sehen ist, sind die meisten Rasterdateien ziemlich groß (manchmal bis mehrere GB), daneben ist die Übertragungsgeschwindigkeit über Internet ziemlich begrenzt. Das Herunterladen von einer ganzen Datei wäre unglaublich zeitraubend und unsinnig, da die Daten nie alle auf einmal gezeigt werden, sondern immer nur ein solcher Ausschnitt, der vom Benutzer erwünscht ist. Dies geschieht jedoch in dementsprechender Zoomstufe, die die Größe des Kartenfensters ermöglicht. Aus oben genannten Gründen ist es empfehlenswert, an den Dateien Optimierungen durchzuführen. Es geht grundsätzlich um zwei Operationen, die jeweils in unumkehrbarer Reihenfolge durchgeführt werden müssen. Und zwar Kachelung und Bildpyramiden. Die dazu geeigneten GDAL-Utilities findet man entweder unter [WEB:GDAL] zum Herunterladen, oft werden sie aber auch gleich mit der Mapserver-Applikation mitgeliefert.

Bei Kachelung (Funktion `gdal_translate`) wird die Struktur der *tif*-Datei in einzelne Kacheln (meistens 256×256 Pixel) zerlegt. Beim Laden der Bilder wird somit die zu übertragene Datenmenge minimiert und die Bilder können durch Kachelung und bedarfsgerechtes Laden eine beliebige Größe haben. Mittels dieses Befehls wird folgende Funktion aufgerufen:

```
gdal_translate -of GTiff -co "TILED=YES" datei.tif datei_tiled.tif
```

Beim Berechnen der Bildpyramiden (Funktion `gdaladdo`) geht es um eine Anpassung der geometrischen Auflösung an die verwendete Zoomstufe. Es werden der gegebenen Anzahl entsprechend Ansichtsbilder in der Datei zugeschrieben. Damit wächst zwar die Dateigröße um etwa ein Drittel, aber dafür können nur wirklich anzeigbare Pixel übertragen werden. Im Befehl wird die Zoomstufe der hinzugefügten Ansichtsebenen angegeben:

```
gdaladdo -r average datei.tif 2 4 8 16 32 64 128 256 512
```

Diese zwei einfach durchführbaren Funktionen beschleunigen erheblich den Ladevorgang der ansonsten ziemlich umfangreichen Dateien. `gdal_translate` darf nicht noch mal nach `gdaladdo` ausgeführt werden, da es die Pyramiden aufhebt.

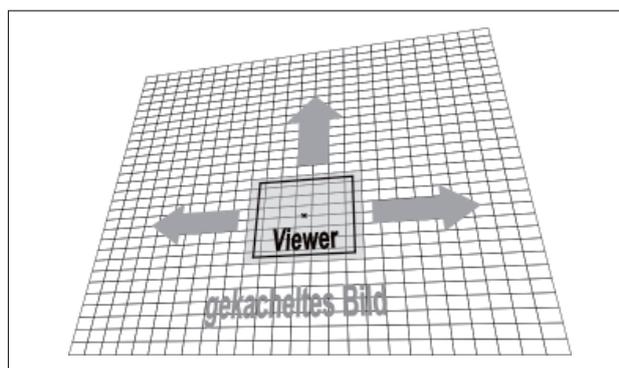


ABB. 17 Kachelung [SIEBOLD 2004]

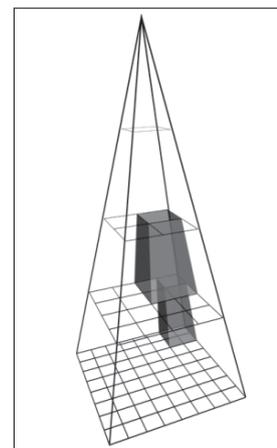


ABB. 18 Bildpyramiden
[SIEBOLD 2004]

Datenbank

Im Fall der Applikation Nasca Mapserver war es nicht nötig, Daten aus einer Datenbank einzubinden. Die Vektordaten aus dem NascaGIS sind nämlich im WMS Kartendienst Pampa of Nasca enthalten. Die Einbindung erfolgt durch die Angabe in einer Mapdatei. Eine genaue Vorgehensweise der Einbindung von Datenbanken kann man in [KLEMM 2008], bzw. im dazugehörigen Tutorial nachlesen.

Um Abfragen zu ermöglichen, ist es jedoch nötig einen WFS Dienst anzuknüpfen. Die Einstellungen werden im Kapitel 4.5.4 näher betrachtet.

Datenbaum

Im linken Bereich der Benutzeroberfläche, im Reiter Maps, werden die verfügbaren Ebenen aufgelistet. Die Reihenfolge der einzelnen Layer wird anhand des Parameters Prio in WMS GUI Einstellungen bestimmt. Der Layer mit dem kleinsten Wert wird als erstes geladen und dargestellt, steht in der Liste ganz oben und liegt in der Karte auf der untersten Ebene. Die Layer sind bezüglich des WMS Dienstes untergeordnet. Mittels Anklicken der blauen Pfeile kann der Benutzer die Ebene nach oben bzw. unten schieben, die Informationen anschauen oder auf sie zoomen. Die linke Auswahlfläche dient zum Ein- oder Ausblenden, die rechte lässt eine Abfrage der Ebene zu.

Leider war es bis zum Ende meiner Arbeit nicht möglich, Ebenen verschiedener WMS Dienste durcheinander zu mischen und deswegen überdecken sich manchmal einige Ebenen aus fremden Kartendiensten ungünstig. In Zukunft sollte es ermöglicht werden, mittels des Moduls treeconfGDE, einen benutzerdefinierten Ordnerbaum zu konfigurieren. Diese Funktionalität befindet sich leider zur Zeit erst in der Entwicklungsphase.

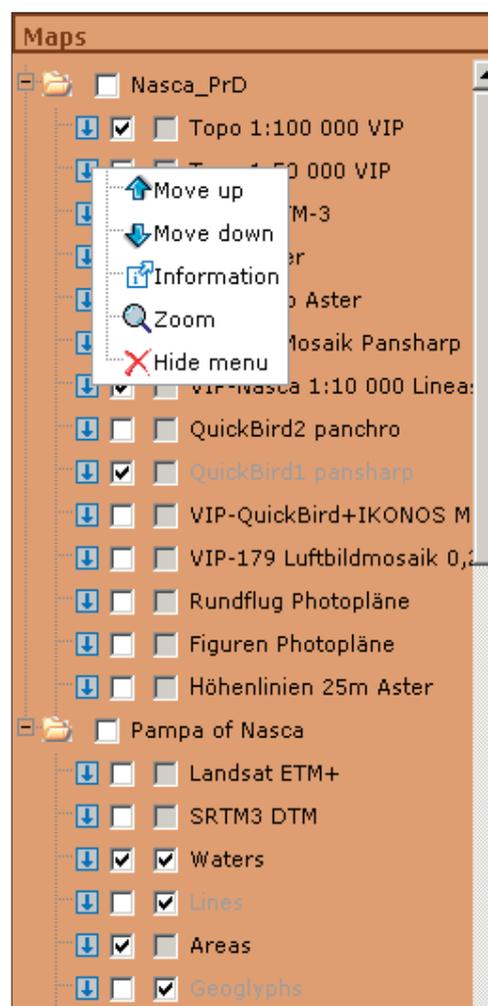


ABB. 19 Mapbender – Datenbaum

4.5.4 Abfragen

Um Abfragen zu den Sachdaten aus einer Datenbank vornehmen zu können, ist es nötig einen WFS Dienst zu erstellen und anzubinden. Dieser Dienst existiert bereits an der HTW unter dem Namen Pampa of Nasca-WFS. Dies ist eine Mapdatei, die nur allgemeine Angaben im HEADER-Bereich enthält, in der WEB-Sektion sind nur Metadaten über den WFS Dienst angegeben und die einzelnen Tabellen im Sinne einzelner Ebenen zur Verfügung gestellt. Die Angaben in LAYER-Bereich beziehen sich auf die Verbindung mit der

Datenbank, je auf eine extra für die WebGIS-Applikation erstellte Tabelle (Materialisierte View). Die Implementierung in einer GUI setzt eine umfangreiche Einstellungen für jede einzelne Ebene voraus. Die Einstellungen habe ich anhand des Tutorials [KLEMM 2008] durchgeführt. Hier nur allgemeine Beschreibung der Vorgehensweise.

Zunächst sind WFS Capabilities hochzuladen (unter WFS Verwaltung). Dafür muss die erweiterte Admin Oberfläche `admin_de_services` verwendet werden. Mittels dieser Anfrage wurden Ebenen aus dem bestehenden WFS Dienst Pampa of Nasca-WFS hochgeladen: `http://141.56.141.2:8080/cgi-bin/mapserv?map=C:/ms4w/Mapfile/Nazca_WFS.map&REQUEST=GetCapabilities&SERVICE=WFS&VERSION=1.0.0`



Im Folgenden sind Attribute, nach denen gesucht werden soll, auszuwählen. Dies macht man im Modul WFS konfigurieren. Bei den neu geladenen Diensten ist die Auswahlliste zu benutzen, während für das Editieren einer bereits konfigurierten Tabelle auf edit WFS Configuration zu klicken ist. Die jeweiligen Schalter und Spalten mit ihren Bedeutungen sind umfangreich in [CHRISTL, EMDE 2008] beschrieben. Daher wird an dieser Stelle nicht auf sie eingegangen.

ABB. 20 Mapbender – WFS konfigurieren

4.5.5 Sicherheit und Lizenzen

Wenn die Daten im Internet zur Verfügung gestellt werden sollen, sind sie unbedingt entsprechend zu schützen. Die Applikation Nasca Mapserver, soll dem Besucher eine Karte bereitstellen, der Benutzer darf jedoch keinesfalls die Möglichkeit haben, einige Änderungen an den Daten vorzunehmen. Außerdem ist es nicht gewünscht, dem Benutzer die Verzeichnisstruktur, die Mapdatei (diese kann Passwörter enthalten) oder andere empfindliche Datenbestände zur Verfügung zu stellen.

Der Apache Webserver in Verbindung mit einem Linux-Server bieten ein großes Potenzial für den Datenschutz. In der Konfigurationsdatei des Apache (`httpd.conf`) lassen sich nur bestimmte Verzeichnisse im Internet zur Verfügung stellen. Darüber hinaus kann man die Zugriffsrechte nur auf ausgewählte IP Adressen beschränken. Eine Linux-Dateistruktur ermöglicht daneben, für einzelne Verzeichnisse, bzw. Dateien genaue Lese-, Schreib- und Ausführrechte festzulegen, und zwar getrennt für den Inhaber, die Benutzergruppe und alle Sonstigen.

Manche Daten sind vom Hersteller unter Lizenz gestellt oder es bestehen entsprechende Urheberrechte. Die jeweiligen Lizenzdateien, bzw. Rechte zu einzelnen Dateien sind in `nasca_dateiliste.xls` (siehe ANHANG 1) deutlich gemacht. Die Einhaltung der Lizenzen und das Respektieren von Urheberrechten stellt eine wichtige Voraussetzung nicht nur für die wissenschaftliche Tätigkeit, sondern auch für alle anderen Anwendungen dar. Da

es im Nascaprojekt viele rechtlich geschützte Daten gibt, habe ich zwei im Prinzip identische GUI erstellt, die sich nur dadurch unterscheiden, dass in der der Öffentlichkeit zur Verfügung gestellten GUI NascaMapserver_guest die lizenzierten Ebenen deaktiviert sind. Zur Anmeldung in die Vollversion (GUI NascaMapserver_vip) sind Name und Passwort gefordert.

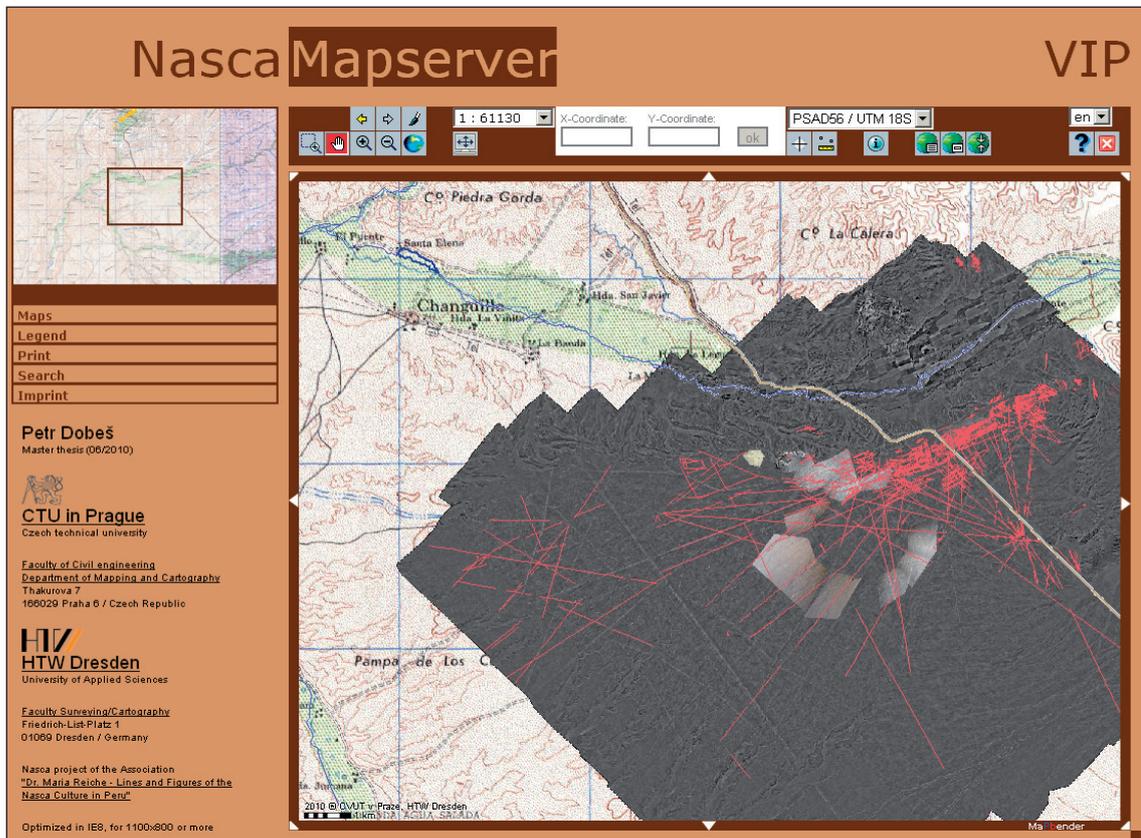


ABB. 21 GUI NascaMapserver_vip

5 Zusammenfassung und Ausblick

5.1 ZUSAMMENFASSUNG

Während der Laufzeit des Nascaprojekts wurden zahlreiche Daten gesammelt und auch große Summen investiert. Es gibt sowohl Rohdaten als auch Ergebnisse von anspruchsvollen Projektarbeiten. Manche Daten bleiben wertvoll und manche sind veraltet oder durch neue ersetzt worden. Alle gehören allerdings zum Projekt und sollten entsprechend gespeichert werden. Die ausgewählten Ergebnisse sind dann einer breiten Öffentlichkeit zu präsentieren, um Kenntnis von einer geheimnisvollen Geschichte der Wüste in Peru zu vermitteln.

Diese Arbeit bietet eine Übersicht über vorhandene Medien und Daten zum Nascaprojekt. Nach der Aufdeckung von Vorzügen und Mängeln der Datenspeicherung wurden einige Regeln aufgestellt mit dem Ziel, die Zusammenarbeit, Kompatibilität und den problemlosen Datenaustausch zu sichern. Sie basieren auf einer einheitlichen, übersichtlichen und selbsterklärenden Verzeichnisstruktur.

Beim Erstellen der WebGIS-Applikation Nasca Mapserver habe ich gelernt, wie man mit Geodaten im Internet umgehen kann. Diese Applikation war die allererste, die ich erstellt habe, deshalb hat es bei mir vielleicht länger gedauert, bis ich die technologischen Aspekte begriffen habe und anschließend alle möglichen Unterlagen und Anleitungen verwirklichen konnte. Ich habe festgestellt, dass eine solche Arbeit wenigstens grundlegende Internet- und folglich Programmierungskennntnisse (PHP, JavaScript, HTML, usw.) erfordert, um eine derartige Applikation effizient erstellen zu können, denn auf diesen Gebieten tauchten die meisten Schwierigkeiten während der Arbeit auf. Bei der Arbeit in einer Umwelt von Open-Source-Software sind gute Englischkenntnisse ebenfalls unerlässlich, da die meisten Dokumentationen, Foren und Mailinglisten in Englisch geführt werden.

Letztendlich habe ich es geschafft, eine funktionierende Applikation zu erstellen und nach gegebenen Anforderungen zu konfigurieren. Sie ist mit aktuellsten Daten angefüllt und kann nach Bedarf aktualisiert und erweitert werden. Dazu habe ich eine kurze Anleitung (siehe ANHANG 6) angefertigt. Die Applikation Nasca Mapserver ist als öffentliche Version aufrufbar unter <http://maps.fsv.cvut.cz/mapbender/frames/login.php?name=guest&password=guest> bzw. als passwortgeschützte Vollversion, inklusive der lizenzierten Daten, unter <http://maps.fsv.cvut.cz/mapbender/frames/login.php?name=vip>

5.2 AUSBLICK

Ich fühle mich nicht in der Lage, die Richtung des gesamten Projektes anzugeben, da ich mich nur mit einem kleinen Teil des Projektes beschäftigt habe. Die Themen *Präsentation in WebGIS* und *Datenhaltung* sind zwar wichtig aber immer nur unterstützend, deshalb möchte ich nur die sich auf diese zwei Themen beziehenden Vorschläge äußern.

Falls es als lohnend bewertet wird, fände ich es hilfreich, wenn die Datenbank um eine Sektion für die Dateienübersicht erweitert würde. Es sollte den kommenden Beteiligten viel Zeit bei verschiedenen Recherchen und Auskünften von beschreibenden Daten ersparen.

Es wäre eine Freude für mich, wenn die von mir erstellte Applikation nicht konserviert, sondern stetig verwendet, erweitert und immer auf aktuellem Stand erhalten bleiben würde. Ich kann mir noch weitere Funktionen vorstellen, die implementiert werden könnten. Die erste Funktion ist bedingt durch die Erstellung einer Dateidatenbank. Wenn die Umrissrahmen von vorhandenen Daten bekannt sind, könnten sie in der Applikation angezeigt werden. Falls es eine neue Version des Mapbenders ermöglicht, wäre es schön, eine übersichtlichere benutzerdefinierte Ordnerbaumstruktur zu machen und die Oberfläche in tschechischer Sprache zu übersetzen. Wenn das Projekt soweit ist, dass von den Geoglyphen oder anderen Datenbankeinträgen die Bilder vorhanden sind, könnten diese durch einen Mausklick auf das Bild aufgerufen werden.

Am wichtigsten finde ich – nicht nur beim Nascaprojekt – eine laufende Kommunikation. So wären einige elementare Regeln für den Informationsaustausch von Vorteil. Damit ist beispielsweise gemeint, dass wenn an der HTW die Datenbank aktualisiert wird oder an der TU neue Daten gekauft werden, dies bekannt gegeben werden sollte, damit alle auf einem aktuellen Stand bleiben.

Nachwort

Es ist mir eine Freude gewesen, an dem internationalen Forschungsprojekt Nasca mitarbeiten zu können. Diese Gelegenheit fand ich recht außergewöhnlich. Den letzten Teil meines Studiums im Kreis der netten und befugten Leute, die beim Nascaprojekt sind, verbringen zu können, war mir ein Vergnügen. Ich habe bei der Arbeit viele neue Kenntnisse erworben. Hoffentlich bin ich dem Projekt auch behilflich gewesen, denn ich habe es gern einige Schritte zum Ende begleitet. Letztendlich kann jeder unter <http://maps.fsv.cvut.cz/~dobespet> sehen, was ich zum Projekt beigetragen habe und darüber hinaus, welches Weltkulturerbe uns allen die Nasca-Kultur vor mehr als 2000 Jahren hintergelassen hat.

Quellenverzeichnis

Literatur

- [BAUMHEKEL, SCHUSTER 2007] Baumhekel, S., Schuster, T.: *Migration von Topobase 2.15 nach Topobase 2007*. Diplomarbeit. HTW Dresden, Dresden 2007
- [BILL 1999] Bill, R.: *Grundlagen der Geo-Informationssysteme, Band 1 Hardware, Software und Daten*. Wichman Verlag, Heidelberg 1999
- [FÜRPAß 2001] Fürpaß, Ch.: *Mapserver als Hilfsmittel zur Datenvisualisierung im Internet*. Diplomarbeit. TU Wien, Wien 2001. Aufrufbar unter <http://www.carto.net/papers/christian_fuerpass/diplomarbeit-fuerpass.pdf> [14.05.2010]
- [CHRISTL, EMDE 2008] Christl, A., Emde, A.: *Mapbender Dokumentation*. Für Version 2.5 vom 29.09.2008, Stand 31.10.2008. Aufrufbar unter <http://svnmirror.osgeo.org/mapbender/trunk/documents/documentation/Dokumentation_mapbender25_de.pdf> [14.05.2010]
- [KLEMM 2008] Klemm, Stephan: *Erstellung einer WebGIS-Applikation des NascaGIS unter Verwendung eines Open Source Produktes*. Diplomarbeit. HTW Dresden, Dresden 2008
- [LINDER 1999] Linder, W.: *Geo-Informationssysteme: ein Studien- und Arbeitsbuch*. Springer, Berlin 1999
- [MITCHELL 2008] Mitchell, T.: *Web-Mapping mit Open Source GIS-Tools*. O'Reilly Verlag GmbH & Co.KG, Köln 2008.
- [PRÜLLER 2006] Prüller, R.: *Visualisierung von Geodaten mit dem UMN Mapserver*. Masterarbeit. TU Graz, Graz 2006. Aufrufbar unter <http://www.geomatics.tugraz.at/download/thesis/Thesis_Prueller_2006.pdf> [14.05.2010]
- [RICHTER 2008] Richter, Ch.: *NascaGIS - An application for cultural heritage conservation*. Dresdener Kartographische Schriften, Band 7, Dresden 2008
- [SCHÖLER 2008] Schöler, A.: *Erstellung einer Internetapplikation zu den Petroglyphen in Palpa/Peru unter Verwendung von Open Source Produkten*. Diplomarbeit. HTW Dresden, Dresden 2008
- [SIEBOLD 2004] Siebold, M.: *Entwurf und Realisierung eines Konzeptes zur pyramidengestützten Bildarstellung*. Masterarbeit, FH Mainz, Mainz 2004. Aufrufbar unter <<http://apps.geoinform.fh-mainz.de/abschlussarbeiten/dokumente/M003/index.html>> [14.05.2010]

Vorlesungsskripte

[SKRIPT:GIS] Schwarzbach, F.: *Geoinformationssysteme*. HTW Dresden 2007/2008

[SKRIPT:FE] Teichert, B.: *Fernerkundung*. HTW Dresden 2007/2008

[SKRIPT:INKA] Cajthaml, J.: *Interaktivní kartografie*. ČVUT v Praze 2008/2009

Internetquellen

[WEB:ABC GEODATA] Axmann Geoinformation: ABC Geodata. Aufrufbar unter <<http://www.abc-geodata.com>> [14.05.2010]

[WEB:AKADEMIE.DE] Online lernen bei akademie.de: *PHP-Einstieg*. Aufrufbar unter <<http://www.akademie.de/programmierung-administration/html-und-css/tipps/skripte-formulare/einstieg-in-php.html>> [14.05.2010]

[WEB:ČEPICKÝ] Čepický, J.: *Seriál Mapový server snadno a rychle*. Aufrufbar unter <<http://gisak.vsb.cz/livecd/tutor/umn/cepik/clanky>> [14.05.2010]

[WEB:ČVUT NASCA] TU Prag: *Projekt Peru*. Aufrufbar unter <<http://lfgm.fsv.cvut.cz/main.php?cap=4&zal=416&lang=cz>> [14.05.2010]

[WEB:EPSG] OGP Surveying & Positioning Committee. Aufrufbar unter <<http://www.epsg.org>> [14.05.2010]

[WEB:FASTIX] Fastix: *Apache Webserver Seminar*. Aufrufbar unter <http://seminare.fastix.de/kurs_apache.html> [14.05.2010]

[WEB:FREEOS] FreeOS: *The Linux filesystem explained*. Aufrufbar unter <<http://www.freeos.com/articles/3102>> [14.05.2010]

[WEB:GDAL] GDAL – *Geospatial Data Abstraction Library*. Aufrufbar unter <<http://www.gdal.org>> [14.05.2010]

[WEB:GISINFO] GISinfo *Geoinformationssystem zum Bestpreis*. Aufrufbar unter <<http://www.gisinfo.de>> [14.05.2010]

[WEB:GIS-WIKI] GISWiki. Aufrufbar unter <<http://www.giswiki.org>> [mehrfach]

[WEB:HTW NASCA] Verein „*Dr. Maria Reiche – Linien und Figuren der Nasca-Kultur in Peru e.V.*“ Aufrufbar unter <<http://www.htw-dresden.de/nazca>> [14.05.2010]

[WEB:MAPSERVER] *MapServer open source webmapping*. Aufrufbar unter <<http://mapserver.org>> [14.05.2010]

[WEB:MB-WIKI] *MapbenderWiki*. Aufrufbar unter <<http://www.mapbender.org>> [mehrfach]

[WEB:PANORAMIO] *Panoramio – Fotos aus aller Welt*. Aufrufbar unter <<http://www.panoramio.com>> [mehrfach]

[WEB:POSTGRESQL.DE] *PostgreSQL, das fortschrittlichste Open Source Datenbanksystem*. Aufrufbar unter <<http://www.postgresql.de>> [14.05.2010]

[WEB:POSTGRESQL.ORG] *PostgreSQL, the world's most advanced open source database*. Aufrufbar unter <<http://www.postgresql.org>> [14.05.2010]

[WEB:SPATIAL REFERENCE] *Spatial Reference*. Aufrufbar unter <<http://spatialreference.org>> [14.05.2010]

[WEB:UNI STUTTGART] Uni Stuttgart: *Arten von Daten*. Aufrufbar unter <http://www.geographie.uni-stuttgart.de/mitarbeiterseiten/Jakob/PcMap/Arten_von_Daten.html> [14.05.2010]

- [WEB:WEBTVORBA] Webtvorba: Barvy, CSS. Aufrufbar unter <<http://www.webtvorba.cz/css/barvy.html#zapis>> [14.05.2010]
- [WEB:WIKIMAPIA] Wikimapia. Aufrufbar unter <<http://wikimapia.org>> [mehrfach]
- [WEB:WIKIPEDIA.DE] Wikipedia – Die freie Enzyklopädie. Aufrufbar unter <<http://de.wikipedia.org>> [mehrfach]
- [WEB:WIKIPEDIA.EN] Wikipedia, the free encyclopedia. Aufrufbar unter <<http://en.wikipedia.org>> [mehrfach]

Weitere Internetseiten

- <<http://download.osgeo.org/osgeo4w>>
- <<http://geo.arc.nasa.gov/sge/landsat/l7.html>>
- <<http://knihovny.cvut.cz/studium/vskp.html>>
- <<http://mapserver.org/de/introduction.html#introduction>>
- <<http://public.beuth-hochschule.de/~photo/projektsat.html>>
- <<http://www.freeos.com/articles/3102/>>
- <<http://www.mysteryperu.com>>
- <<http://www.remotesensing.org/gdal/ogr/index.html>>
- <<ftp://ftp.remotesensing.org/mapserver/docs/mapserver-users-manual.pdf>>

Anhangsverzeichnis

ANHANG 1	CD
ANHANG 2	Verzeichnisstruktur MS4W
ANHANG 3	Installation von MS4W
ANHANG 4	Installation von PostgreSQL/PostGIS
ANHANG 5	Anpassung der Konfiguration
ANHANG 6	Verwaltungsanleitung Nasca Mapserver

ANHANG 1 CD

- PDF Diplomarbeit
- Datenmodell NascaGIS von [BAUMHEKEL, SCHUSTER 2007]
- Mapfile nasca_PrD.map
- SQL Quellcode von einzelnen GUIs
- Tabelle nasca_dateiliste.xls (Blätter Dateien, Formate)
- Verwaltungsanleitung Nasca Mapserver

ANHANG 2 Verzeichnisstruktur MS4W

Das Verzeichnis `ms4w` muss sich direkt im `root`-Verzeichnis auf beliebiger Festplatte des Rechners bzw. Servers befinden. Es beinhaltet mehrere Unterverzeichnisse, auf deren Inhalt jetzt eingegangen wird. Erläuterungen erfolgen nach [KLEMM 2008]

Apache/

bin/	In diesem Ordner befindet sich der Apache Monitor für eine nutzerfreundliche Steuerung des Apache.
cgi-bin/	Hier liegen das UMN MapServer-Programm, die zugehörigen Bibliotheken sowie die Skriptsprache PHP.
conf/	Der Ordner enthält die zentrale Konfigurationsdatei des Apache. Des Weiteren beinhaltet der Ordner extra zusätzliche <code>conf</code> -Dateien für beispielsweise die Verschlüsselung des Servers oder der Einrichtung eines virtuellen Hosts.
error/	Das Unterverzeichnis listet die möglichen Fehlermeldungen, die vom Server ausgegeben werden können.
htdocs/	Das Verzeichnis gibt der Webserver für die Zugriffe über das WWW frei. Hierin liegen beispielsweise die Startseiten, die über das Internet zu erreichen sind. Im Fall der Anwendung liege darin das Mapbender-Verzeichnis.
logs/	Hier werden die Zugriffe auf den Webserver sowie die Fehlermeldungen gespeichert.
manual/	In diesem Verzeichnis sind HTML-Seiten mit weiterführenden Informationen über die Installation oder Konfiguration aufgeführt.
modules/	In diesem Ordner befinden sich alle Module, die sich über die zentrale <code>conf</code> -Datei des Apache installieren lassen.
php/	Der Unterordner enthält für die Skriptsprache PHP die notwendigen Bibliotheken. In dem Unterverzeichnis <code>/php/session/</code> werden die Sitzungen des Mapbenders protokolliert. Zum Beispiel wird gespeichert, wie lange der Map-Client mit dem Webserver verbunden ist. Anhand des Sessionmanagements ist ein Server in der Lage, die Anfragen über HTTP in Bezug zu setzen und die Clients zu unterscheiden.
specialplugins/	Der letzte Ordner führt die Zugriffsbibliotheken auf ArcSDE.

apps/

Es enthält die bereits vorkonfigurierten Web-Dateien.

gdaldata/

Das Verzeichnis führt alle notwendigen Dateien, die im Zusammenhang mit der Rasterdaten-Bibliothek stehen.

gdalplugins/

Das Verzeichnis enthält Bibliotheken, die einen Zugriff über OGR z.B. auf eine Oracle-Datenbank realisieren.

proj/

Es ist ein sehr wichtiges Verzeichnis. Darin sind alle Projektionen aufgeführt, die der MapServer verarbeiten kann.

python/

Diesem Verzeichnis obliegt die Speicherung von Mapscripten unter Verwendung der Programmiersprache Python.

tmp/

Mit Hilfe dieses Unterordners werden die generierten Bilder des MapServers zwischengespeichert.

tools/

Im letzten Ordner des MS4W-Verzeichnisses sind die verschiedenen Bibliotheken wie z.B. GDAL und OGR aufgeführt.

ANHANG 3 Installation von MS4W

Aus der Seite <http://www.maptools.org/ms4w/index.phtml?page=downloads.html> ist entweder `ms4w-2.3.1-setup.exe` auszuführen oder die Datei `ms4w_2.3.1.zip` herunterzuladen und im `root`-Verzeichnis auszupacken. Bei direkter Installation durch die `setup` Datei sind anschließend wenige Installationsfenster auszufüllen, dabei ist darauf zu achten, dass Installationspfad auf einem beliebigen Laufwerk liegt und keine Unterverzeichnisse beinhaltet. Ich habe lieber die `zip` Datei heruntergeladen, dabei ist im Folgenden nur die Datei `/ms4w/Apache/bin/httpd.exe` auszuführen, um den Apache Webserver zu starten. Damit ist MS4W technisch installiert und kann durch `http://localhost` oder `http://127.0.0.1` Eingabe in Adressenzeile eines Browsers getestet werden. Hat alles gut geklappt, erscheint die Startseite des MS4W. Weiter kann noch die fehlerfreie Funktion des Mapservers durch `http://localhost/cgi-bin/mapserv?` geprüft werden. Erscheint die Fehlermeldung „No query information to decode. QUERY_STRING is set, but empty“, ist die Installation erfolgreich durchgelaufen.

ANHANG 4 Installation von PostgreSQL/PostGIS

Aus der Seite <http://www.postgresql.org/download> ist die Datei `postgresql-8.4.2-1-windows.exe` (oder neuere Version) herunterzuladen oder direkt auszuführen. Seit Version 8 ist PostGIS in PostgreSQL mit enthalten. Es kann jedoch immer von <http://postgis.refrains.net/download> heruntergeladet werden.

Während des Installationsverlaufs wird man wie üblich begleitet. Es ist darauf zu achten, dass die vorgewählte Auswahl der installierten Komponenten um Unterstützung für Sprachen (für Status und Fehlermeldungen in einer anderen Sprache) und PostGIS (um räumliche Operationen zu ermöglichen) erweitert werden sollte. Im nächsten Schritt erfolgt die Dienst-Konfiguration in der Windows-Systemumgebung. Ebenso soll ein Datenbank-Cluster initialisiert werden. Mit der Initialisierung eines Datenbank-Clusters entsteht ein Verzeichnis, in dem alle Dateien der Datenbankinstallation abgelegt und alle Datenbanken gespeichert werden. Der Port `5432` ist der Standard-Port und sollte deshalb freigegeben werden. Bei der Option Adressen lässt sich über ein Häkchen auch Verbindungen von anderen Rechnern auf die Datenbank herstellen. Ohne das Häkchen sind nur Verbindungen von `localhost` zulässig. Die Einstellung `Locale C` weist darauf hin, dass der Computer die lokalen Ländereinstellungen übernimmt. Encoding definiert den zu verwendenden Zeichensatz, den die Datenbank nutzt. Die Voreinstellung `WIN1252` ist auf den Standard für Unicode-Zeichen `UTF-8` gesetzt. Anschließend ist ein Benutzername mit Passwort für den Datenbank-Administrator (Superuser) zu definieren. Weitere Fenster sind für unsere Zwecke nicht von Bedeutung, und sind deshalb mit dem Button **Weiter** durchzugehen um so die Installation zu beenden. In Windows erfolgt die Verwaltung einer Datenbank durch pgAdmin III (Start-Programme-PostgreSQL8.3-pgAdmin).

ANHANG 5 Anpassung der Konfiguration

Die Änderungen betreffen folgende Dateien, bzw. Verzeichnisse:

- /etc/apache2/apache2.conf
- /etc/apache2/sites-enabled/000-default (verweist auf etc/apache2/sites-available/default)
- /etc/apache2/mods-enabled/ (verweist auf etc/apache2/mods-available/)
- /etc/apache2/ports.conf
- /ms4w/Apache/conf/httpd.conf
- /ms4w/Apache/cgi-bin/php.ini
- mapbender/conf/mapbender.conf

Apache Webserver

Pfad der Konfigurationsdatei

```
ServerRoot „/etc/apache2“  
ServerRoot „/ms4w/Apache“
```

Portnummer

```
# Include ports listing  
Include /etc/apache2/ports.conf  
Listen 80  
Listen 80
```

Virtual host

```
# Include the virtual host configurations:  
Include /etc/apache2/sites-enabled/
```

Die Datei 000-default ist nur eine Verlinkung auf etc/apache2/sites-available/default. Diese Datei beginnt und endet mit dem Tag <VirtualHost *>.

```
NameVirtualHost *  
<VirtualHost *>  
...  
</VirtualHost *>  
  
Document root-Verzeichnis  
DocumentRoot /var/www/  
DocumentRoot “/ms4w/Apache/htdocs/”
```

Rechte für die Verzeichnisse

Hier erfolgt die Angabe der Rechte für das root-Verzeichnis.

```
<Directory />  
Options FollowSymLinks  
AllowOverride FileInfo  
</Directory>  
<Directory />  
Options none  
AllowOverride None  
Order allow,deny  
Allow from all  
</Directory>
```

Weitere Angaben bestimmen die Rechte für andere Verzeichnisse.

```
<Directory /var/www/>  
Options FollowSymLinks MultiViews  
Order allow,deny  
allow from all  
# This directive allows us to have apache2's default start page  
# in /apache2-default/, but still have / go to the right place  
#RedirectMatch ^/$ /apache2-default/  
</Directory>
```

```
<Directory /data>  
Options FollowSymLinks MultiViews  
Order allow,deny  
allow from all  
# This directive allows us to have apache2's default start page  
# in /apache2-default/, but still have / go to the right place  
#RedirectMatch ^/$ /apache2-default/  
</Directory>
```

```
<Directory "/ms4w/Apache/htdocs/">  
Options Indexes FollowSymLinks  
AllowOverride None  
Order allow,deny  
Allow from all  
</Directory>
```

```
<Directory "/Programy/PostgreSQL/8.4/bin/">  
Options none  
AllowOverride None  
Order allow,deny
```

```
Allow from all  
</Directory>
```

```
<Directory "/ms4w/Mapfile/">  
Options none  
AllowOverride None  
Order allow,deny  
Allow from all  
</Directory>
```

```
Das Script (CGI-) Verzeichnis  
ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/  
<Directory „/usr/lib/cgi-bin“>  
AllowOverride None  
Options ExecCGI -MultiViews +SymLinksIfOwnerMatch  
Order allow,deny  
Allow from all  
</Directory>
```

```
ScriptAlias /cgi-bin/ "/ms4w/Apache/cgi-bin/"  
ScriptAlias /fcgi-bin/ "/ms4w/Apache/cgi-bin/"
```

```
Error Log Warnungen  
ErrorLog /var/log/apache2/error.log  
ErrorLog „logs/error.log“
```

UMN MapServer

MapServer Mapfile Abkürzung

```
SetEnvIf Request_URI „/cgi-bin/nasca“ MS_MAPFILE=/data/_projekty/nasca/  
nasca_PrD.map  
SetEnvIf Request_URI "/cgi-bin/nasca" MS_MAPFILE= G:/ms4w/Mapfile/nazca_  
PrD.map
```

```
Projektionsordner proj  
SetEnv PROJ_LIB „/usr/share/proj/nad/“  
SetEnv PROJ_LIB „/ms4w/proj/nad/“
```

```
# set GDAL_DRIVER_PATH environment variable for gdal plugins  
SetEnv GDAL_DRIVER_PATH "/usr/share/gdalplugins"  
# set GDAL_DRIVER_PATH environment variable for gdal plugins  
SetEnv GDAL_DRIVER_PATH „/ms4w/gdalplugins“
```

Zusätzliche Module

```
# Include module configuration:  
Include /etc/apache2/mods-enabled/*.load  
Include /etc/apache2/mods-enabled/*.conf
```

Auf dem Linux-Server sind weiterhin Verlinkungen auf `mods-available/mod_deflate` und `mods-available/mod_headers` in `mods-enabled` anzulegen.

```
LoadModule deflate_module modules/mod_deflate.so  
LoadModule headers_module modules/mod_headers.so
```

Mapbender

Virtuelle Verzeichnisse

```
Alias /mapbender/ „/var/www/mapbender/http/“  
<Directory „/var/www/mapbender/http/“>  
Options Indexes  
AllowOverride None  
Order allow, deny  
Allow from all  
</Directory>
```

```
Alias /mapbender/ “/ms4w/Apache/htdocs/mapbender/http/”  
<Directory “/ms4w/Apache/htdocs/mapbender/http/”>  
Options Indexes MultiViews FollowSymlinks  
AllowOverride None  
DirectoryIndex index.php  
Order allow,deny  
Allow from all  
</Directory>
```

Apache Modul mod-deflate

Man muss den folgenden Text an den `<Directory>` Eintrag des Mapbenders anhängen.

```
# Insert filter  
SetOutputFilter DEFLATE  
# Netscape 4.x has some problems...  
BrowserMatch ^Mozilla/4 gzip-only-text/html  
# Netscape 4.06-4.08 have some more problems  
BrowserMatch ^Mozilla/4\.[0678] no-gzip  
# MSIE masquerades as Netscape, but it is fine  
# BrowserMatch \bMSIE !no-gzip !gzip-only-text/html  
# NOTE: Due to a bug in mod_setenvif up to Apache 2.0.48  
# the above regex won't work. You can use the following  
# workaround to get the desired effect:
```

```
BrowserMatch \bMSI[E] !no-gzip !gzip-only-text/html
# Don't compress images
SetEnvIfNoCase Request_URI \
\.(?:gif|jpe?g|png)$ no-gzip dont-vary
# Make sure proxies don't deliver the wrong content
Header append Vary User-Agent env=!dont-vary
```

Zugriff auf entfernte Dateien

```
extension_dir = „/ms4w/Apache/php/ext/“
allow_url_fopen = On
```

Session

Auf dem angegebenen Verzeichnis müssen Schreibrechte vorliegen.

```
session.save_handler = files
session.save_path = „/ms4w/Apache/php/session/“
```

Erweiterungen

Das Kommentarzeichen ist zu entfernen, damit Mapbender zusätzliche Erweiterungen lädt.

```
extension=php_gd2.dll
extension=php_gettext.dll
extension=php_mbstring.dll
extension=php_pgsql.dll
```

Auswahl der Administrationsdatenbank

Je nach dem, welche Datenbank betrieben werden soll, ist das Kommentarzeichen zu entfernen.

```
#define („SYS_DBTYPE“, „mysql“);
define („SYS_DBTYPE“, „pgsql“);
```

Einstellungen für den Datenbankzugriff

```
else {
define(„DBSERVER“, „localhost“);
define(„PORT“, „5432“);
define(„DB“, „nasca“);
define(„OWNER“, „mapbender“);
define(„PW“, „password“); # Angabe eine Passwortes
include_once(„../mapbender/http/php/databasepgsql.php“);
}
```

Nutzung von Prepared Statements

```
define („PREPAREDSTATEMENTS“,true); # Wert „true“ oder „false“
```

Encoding der Datenbank

```
define („CHARSET“, „UTF-8“);
```

Temp-Verzeichnis

```
define („TMPDIR“, „../tmp“);
```

Die Login-URL

```
define („LOGIN“, „http://maps.fsv.cvut.cz/mapbender/frames/login.php“);  
define („MAXLOGIN“, 5);  
$login = LOGIN;
```

Server-Connection-Typ

```
#define(„CONNECTION“, „curl“);  
define(„CONNECTION“, „http“);  
#define(„CONNECTION“, „socket“);
```

ANHANG 6 Verwaltungsanleitung Nasca Mapserver

VERWALTUNGSANLEITUNG NASCA MAPSERVER

Neue Raster- oder Vektordatei hinzufügen

maps.fsv.cvut.cz

/data/nasca – Dateien einladen (mittels WinSCP)

/data/_projekty/nasca – Mapfile editieren

- neue Ebene einfügen, Reihenfolge durchdenken und nach Bedarf ändern
- alle Parameter entsprechend ändern und ausreichend beschreiben (# kommentieren)

maps.fsv.cvut.cz/mapbender/frames/login.php

als root einloggen

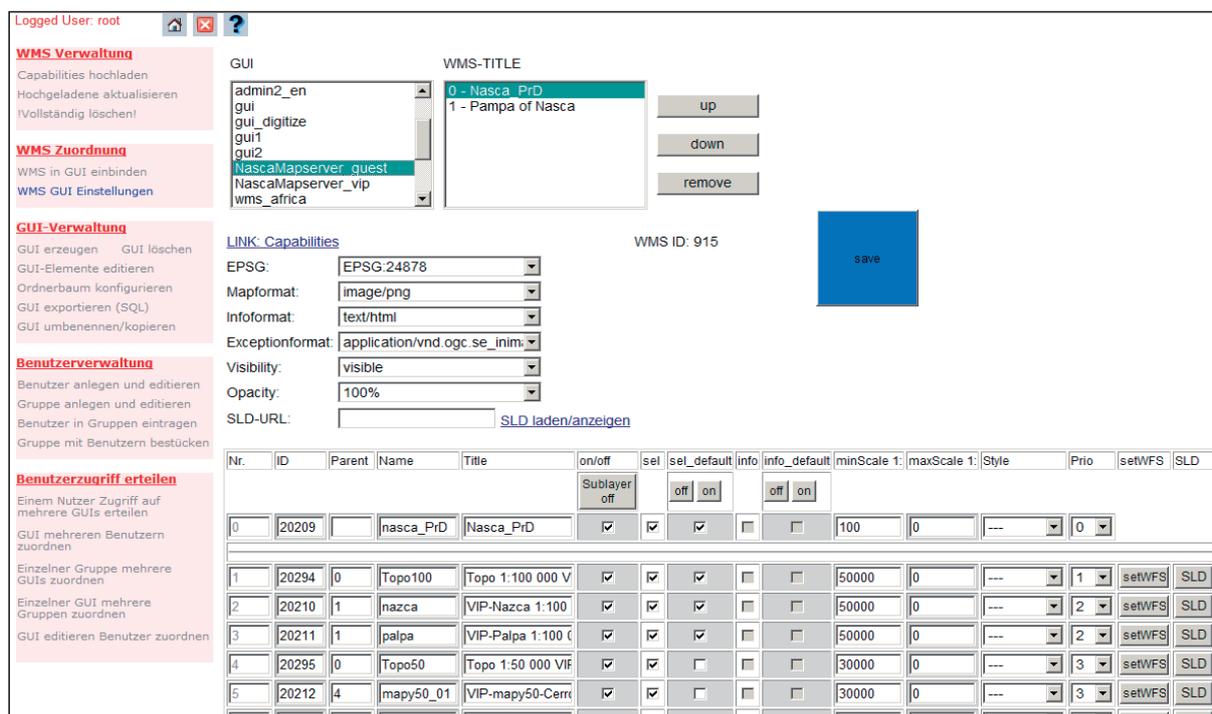
Admin GUI admin2_de öffnen

WMS Verwaltung – Hochgeladene aktualisieren

- Nasca_PrD auswählen
- Link von erste Zeile (last uploaded) in zweite Zeile (new WMS) kopieren
- durch Preview Capabilities prüfen
- Upload Capabilities Button drücken

WMS Zuordnung – WMS GUI Einstellungen

- GUI NascaMapserver_guest oder NascaMapserver_vip auswählen
- WMS Nasca_PrD auswählen
- bei der neuen Ebene Parameter entsprechend einstellen
 - on/off – Ausgangstatus (beim Starten der Applikation)
 - sel, info – anzeigbar, abfragbar
 - sel_default, info_default - sel, info Ausgangstatus
 - minScale, maxScale
 - prio – Reihenfolge sinngemäß anpassen
- mittels save Schaltfläche bestätigen



POKYNY KE SPRÁVĚ NASCA MAPSERVERU

Přidání nové datové vrstvy

maps.fsv.cvut.cz

/data/nasca – nahrát soubor (pomocí WinSCP)

/data/_projekty/nasca – editovat Mapfile

- přidat novou vrstvu, promyslet a podle potřeby změnit pořadí
- doplnit a přehledně popsat parametry nové vrstvy (# komentářem)

maps.fsv.cvut.cz/mapbender/frames/login.php

přihlásit se jako uživatel *root*

otevřít Admin GUI *admin2_en*

WMS Management – Update WMS

- zvolit *Nasca_PrD*
- odkaz z prvního řádku (last uploaded) zkopírovat do druhého řádku (new WMS)
- ověřit správnost pomocí Preview Capabilities
- stisknout Upload Capabilities

Configure WMS Access – WMS GUI settings

- zvolit GUI *NascaMapserver_guest* nebo *NascaMapserver_vip*
- zvolit WMS *Nasca_PrD*
- nastavit parametry u nové vrstvy
 - on/off – výchozí stav vrstvy (při startu aplikace)
 - sel, info – možnost zobrazit/vypnout vrstvu, možnost dotazu na vrstvu
 - sel_default, info_default – výchozí stav parametrů sel, info
 - minScale, maxScale
 - prio – zkontrolovat a přizpůsobit pořadí zobrazení vrstev
- potvrdit pomocí tlačítka *save*

Nr.	ID	Parent	Name	Title	on/off	sel	sel_default	info	info_default	minScale 1:	maxScale 1:	Style	Prio	setWFS	SLD
0	20209		nasca_PrD	Nasca_PrD	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	100	0	---	0		
1	20294	0	Topo100	Topo 1:100 000 V	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	5000	0	---	1	setWFS	SLD
2	20210	1	nazca	VIP-Nazca 1:100	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	5000	0	---	2	setWFS	SLD
3	20211	1	palpa	VIP-Palpa 1:100	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	5000	0	---	2	setWFS	SLD
4	20295	0	Topo50	Topo 1:50 000 VIF	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	3000	0	---	3	setWFS	SLD
5	20212	4	mapy50_01	VIP-mapy50-Cerr	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	3000	0	---	3	setWFS	SLD