

Visualisierung eines Besucherstrommodells

Bachelorarbeit
Frühjahrssemester 2010

Autor: Marcel Germann
Betreuer: Prof. Dr. Lothar Müller
Co-Betreuung: Mathias Manz
Experte: Markus Flückiger
Gegenleser: Prof. Hansjörg Huser

Technischer Bericht

Erstellt am: 4. März 2010
Abgabedatum: 18. Juni 2010

Erklärung

Ich erkläre hiermit,

- dass ich die vorliegende Arbeit selber und ohne fremde Hilfe durchgeführt habe, ausser derjenigen, welche explizit in der Aufgabenstellung erwähnt ist oder mit dem Betreuer schriftlich vereinbart wurde,
- dass ich sämtliche verwendeten Quellen erwähnt und gemäss gängigen wissenschaftlichen Zitierregeln korrekt angegeben habe.

Ort, Datum:

Name, Unterschrift:

Änderungsnachweis

Version	Änderungsgrund	Kurz-Z.	Datum
1.0	Erstellen des Dokumentes	mg	4.3.10

Inhalt

1	Einleitung	7
1.1	Management Summary	7
1.2	Stand des Visimanprojektes	9
1.3	Abgrenzung	9
2	Problemanalyse	10
2.1	Was ist ein Besucherstrommodell	10
2.2	Zweck des Besucherstrommodells	10
2.3	Datenbestand	11
2.4	Abdeckung.....	13
2.5	Auswertung.....	13
2.6	Erfahrungswerte	17
3	Funktionale Anforderungen	18
3.1	Funktionsliste.....	18
3.2	Szenarien	19
4	Use Cases	21
4.1	Aktor Parkmanager.....	21
4.2	UC 1: Besucherstrom visualisieren.....	21
4.3	UC 2: Daten eingeben	22
5	Nichtfunktionale Anforderungen	23
5.1	Leistungsanforderung.....	23
5.2	Randbedingungen	23
5.3	Qualität	23
5.4	Technologien	23
6	Lösungskonzept	24
6.1	Ansichtskonzept	24
6.2	Darstellungskonzept.....	27
6.3	Filterkonzept	28
7	Realisierungskonzept	30

7.1	Kriterien	30
7.2	GIS Erweiterungen	31
7.3	Desktop Applikation	32
7.4	Webapplikation	32
7.5	Auswertung	36
8	Evaluation Technologien.....	37
8.1	Server	37
8.2	Client	37
8.3	Auswertung Technologien	39
9	Verknüpfung der Mess- und Geodaten	40
9.1	Veränderbare Attributtabelle durch Datenbankview	40
9.2	Verändern der Konfiguration	41
9.3	Konfiguration durch SLD verändern	42
9.4	Berechnung auf Client	42
9.5	Themenlayer WFS – SLD auf Client.....	43
9.6	Auswertung der Verknüpfungsmethoden	44
10	Installation Server	45
10.1	Installation GDAL.....	45
10.2	Installation Proj.4	45
10.3	Installation GEOS	46
10.4	Installation PostgreSQL	46
10.5	Installation PostGIS	47
10.6	Einführung in MapServer	49
10.7	Installation von MapServer auf einem Linux Server	49
10.8	Konfiguration mit dem MapFile	50
10.9	Konvertieren von Shapefiles.....	51
11	Benutzeroberfläche.....	53
11.1	Bedienelemente.....	53
11.2	Layout.....	55
11.3	Entwurf einer Benutzeroberfläche	56
11.4	Realisierte Benutzeroberfläche	57
12	Implementation.....	58
12.1	Webapplikation	58
12.2	Server	65
13	Probleme.....	69
13.1	GeoExt / OpenLayers und Internet Explorer.....	69

13.2	Überlappende Beschriftungen werden von MapServer nicht dargestellt.....	69
13.3	Ladeanzeige nicht möglich	70
13.4	Verschieben der Karte während der Ladephase	70
14	Schlussfolgerungen.....	71
14.1	Was wurde erreicht.....	71
14.2	Was wurde nicht erreicht	71
14.3	Erweiterungen für die Webapplikation	72
14.4	Ausblick	72
Glossar	73
Quellen	75
	Literaturverzeichnis	75
	Abbildungsquellen.....	76
Abbildungsverzeichnis.....		77
A - Aufgabenstellung.....		78
B - Protokoll der Sitzung mit Herr Dominik Siegrist.....		82
C - Protokoll der Sitzung mit Herr Ronald Schmidt.....		84
D - Protokoll der Sitzung mit Herr Stefan Keller		87

I Abstract

- Ausgangslage** Die Natur- und Nationalpärke wollen eine Überbelastung der sensiblen Räume durch die Besucher vermeiden. Dies kann mit Besucherlenkungen wie zum Beispiel Absperrungen oder Informationstafeln erreicht werden.
- Besucherlenkungen werden oft eingesetzt, doch bis anhin sind nur wenige Erfahrungen über deren Erfolg vorhanden. Zur Überprüfung der Besucherlenkungsmassnahmen kann ein Besucherstrommodell eingesetzt werden. Zusätzlich können weitere Informationen wie zum Beispiel die Interessen der Besucher und die Entwicklung der Besucherfrequenzen aufgezeigt werden.
- Als Grundlage für das Besucherstrommodell stehen manuelle Zählungen, sowie Messdaten von automatischen Zählungen z.B. durch Zählmatten oder induktive Schleifen zur Verfügung. Ziel dieser Arbeit ist die Entwicklung einer Applikation, die Besucherströme visualisieren kann und somit eine Auswertung vereinfacht.
- Vorgehen / Technologien** Durch Besprechungen mit Mitgliedern des Projektes VISIMAN wurde das Problem analysiert und die Anforderungen an die Visualisierung festgehalten. Anhand dieser Vorgaben wurde ein Lösungskonzept erarbeitet, welches verschiedene Lösungsvarianten aufzeigt.
- In einer Auswertung werden die Vor- und Nachteile der Varianten erklärt. Um ein Lösungskonzept umzusetzen, sind zuerst verschiedene Technologien evaluiert worden. Die Evaluation beschränkte sich auf die Technologien Softwareapplikation, GIS Plugin und Webapplikation.
- Zum Abschluss wurde eine Visualisierung anhand des erstellten Realisierungskonzeptes entwickelt und getestet.
- Ergebnis** Bei den Besprechungen mit den Anwendern hat sich gezeigt, dass es eine möglichst einfache Applikation braucht, welche die Besucherfrequenzen und -ströme auf einer Karte oder in Statistiken visualisiert. Die Messdaten sollen in der Applikation mit bestimmten Attributen gefiltert werden können.
- Für die Berechnung der Gesamtbesucherzahl und das Erstellen eines vollständigen Besucherstrommodells (unter Einbezug der Wege ohne Zählungen) müssten Erfahrungswerte berücksichtigt werden können. Es hat sich gezeigt, dass eine automatisierte Erstellung eines Besucherstrommodells nicht möglich ist, da zu viele Faktoren das Modell beeinflussen. Diese Faktoren sind in jedem Park unterschiedlich.
- Zur einfacheren Auswertung der Besucherdaten und zum Sammeln der benötigten Erfahrungswerte, wurde eine Webapplikation erstellt. Die Grundfunktionen wurden implementiert und die Applikation kann einfach durch weitere Funktionen erweitert und angepasst werden.

1 Einleitung

1.1 Management Summary

1.1.1 Ausgangslage

Die Natur- und Nationalpärke wollen eine Auswertung der Besucherfrequenzen und Besucherströme machen können. Die Auswertung wird benötigt, um zum Beispiel die Entwicklung bei Besucherlenkungsmaßnahmen zu verfolgen oder Erfahrungswerte über die Auslastung der Wege zu sammeln. Aus den visualisierten Besucherdaten können auch die Interessen der Besucher analysiert werden.

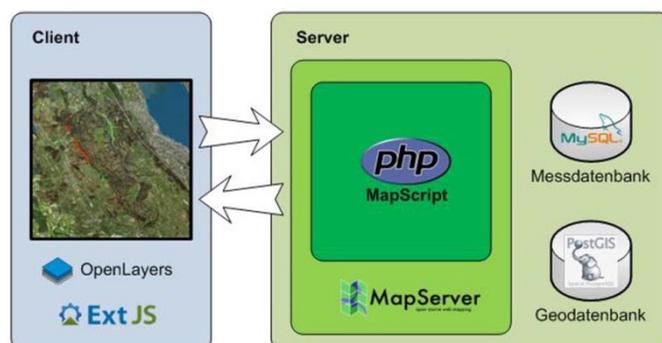
Als Grundlage für die Visualisierung des Besucherstrommodells stehen manuelle Zählungen, sowie Messdaten von automatischen Zählungen z.B. durch Zählmatten oder induktive Schleifen zur Verfügung. Ziel dieser Arbeit ist es Konzepte für die Visualisierung zu entwerfen und eines als Prototyp zu implementieren.

1.1.2 Vorgehen

Die Arbeit wird in vier Phasen unterteilt. In der ersten Phase wird das Problem analysiert. Dazu werden Gespräche mit den Endbenutzern geführt und die Anforderungen festgehalten. In einer zweiten Phase wird anhand der Anforderungen ein Lösungskonzept mit verschiedenen Varianten erstellt.

In der dritten Phase werden mögliche Technologien evaluiert und ein Realisierungskonzept entworfen. Die Auswertung dieses Realisierungskonzeptes entscheidet, mit welchen Technologien und Methoden der Prototyp entwickelt wird.

Dieser Prototyp wird in der vierten Phase implementiert und hat das Ziel, zu beweisen, dass die im Lösungskonzept und Realisierungskonzept vorgestellten Technologien und Methoden implementiert werden können.

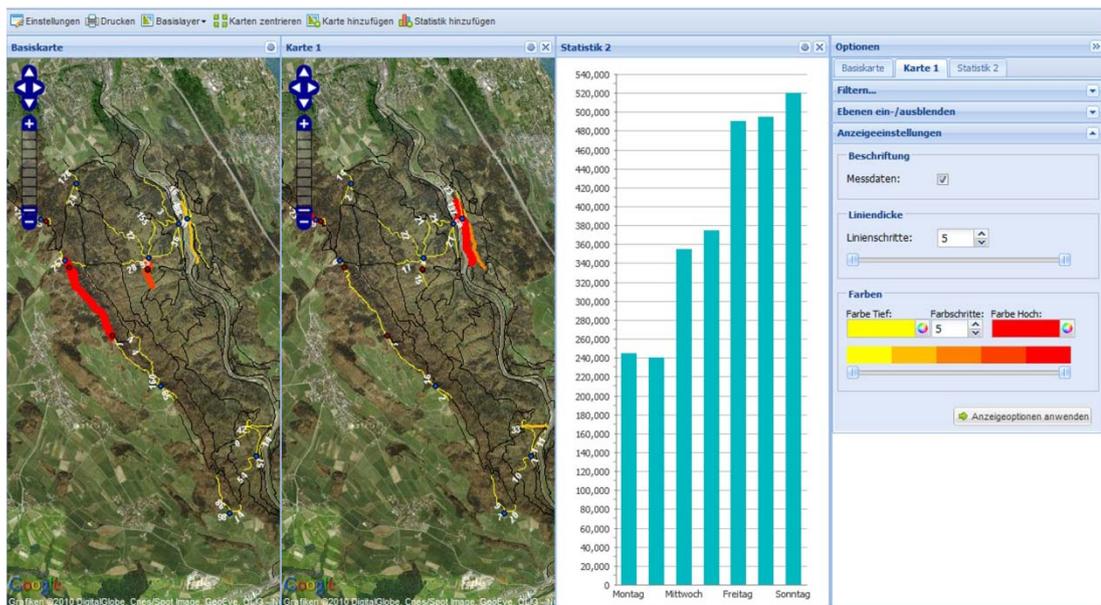


1.1.3 Ergebnisse

Die Wünsche und Anforderungen der Endbenutzer wurden analysiert und in einem Lösungskonzept zusammengetragen. Es wurde erkannt, dass neben der Visualisierung der Besucherfrequenzen und Besucherströme auch das Berechnen der Gesamtbesucherzahl und das Erstellen eines Besucherstrommodells mit Einbezug aller Wege gewünscht werden. Diese Anforderungen hätten den zeitlichen Rahmen der Arbeit gesprengt und man hat sich auf die Visualisierung der Besucherfrequenzen von vorhandenen Messdaten begrenzt.

Das Lösungskonzept wurde in drei Konzepte unterteilt. Das **Ansichtskonzept** zeigt auf, in welcher Form die Besucherzahlen visualisiert werden können. Im **Darstellungskonzept** wird beschrieben, wie die Unterschiede zwischen den Besucherzahlen erkennbar dargestellt werden können. Mit dem **Filterkonzept** wurden Methoden entwickelt, wie die Besucherzahlen sinnvoll gefiltert werden können.

Im Realisierungskonzept wurden Technologien und Methoden evaluiert, die als Hilfsmittel gebraucht wurden, um das Lösungskonzept zu implementieren. Als Resultat der Evaluation wurde beschlossen, eine Webapplikation mit den Technologien ExtJS und OpenLayers für die Visualisierung der Karte und MapServer mit MapScript auf der Serverseite zu erstellen.



Mit dem implementierten Prototyp wurde aufgezeigt, dass es möglich ist, das erstellte Lösungskonzept zu realisieren. Die Webapplikation wurde mit Grundfunktionen ausgestattet, die es erlauben, mehrere Karten und Diagramme anzuzeigen, die individuell gefiltert und in der Darstellung angepasst werden können.

1.1.4 Ausblick

Mit dem Prototyp wird gezeigt, dass die Implementierung des Lösungskonzeptes wie auch des Realisierungskonzeptes möglich ist. Die im Prototyp erstellte Webapplikation kann ausgebaut und erweitert werden, damit ein Produktives System entsteht.

1.2 Stand des Visimanprojektes

Die Bachelorarbeit „Visualisierung eines Besucherstrommodells“ ist die letzte der geplanten Arbeiten. Vorgängig wurden folgende Arbeiten durchgeführt

Technologiestudie VISIMAN

In der Technologiestudie wurden Szenarien für die Überwachung der Natur- und Nationalpärke entwickelt. Für die Implementierung wurde ein Dispatchersystem evaluiert. Das Ergebnis der Bachelorarbeit war, Nagios als Dispatchersystem einzusetzen.

Entwicklung eines Besuchermonitoring-Systems

In der Studienarbeit wurde eine Java Mobile Applikation zur manuellen Besucherzählung implementiert. Das in der Technologiestudie als Prototyp entwickelte Datenmodell wurde weiterentwickelt und für die Speicher von verschiedenen Zählungen und Sensoren angepasst.

1.3 Abgrenzung

Die Bachelorarbeit wird als Nachfolgearbeit der Studienarbeit „Entwicklung eines Besuchermonitoring-Systems“ und der Bachelorarbeit „Technologiestudie VISIMAN“ durchgeführt. Die Infrastruktur des Servers wird von diesen Projekten übernommen und daher keine Anleitungen zur Installation und Konfiguration des Servers dokumentiert.

2 Problemanalyse

2.1 Was ist ein Besucherstrommodell

Natur- und Nationalpärke sind in einem grösseren Gebiet verteilt. Um im Park die Tiere und Sehenswürdigkeiten anschauen zu können, gibt es viele Wege, die zu einem grossen Wegenetz zusammengefasst werden können. Die Besucher können sich im ganzen Park frei bewegen und benutzen daher ganz unterschiedliche Routen und Wege.

Ein Besucherstrommodell zeigt auf, wieviele Besucher über eine bestimmte Zeit hinweg einen Weg benutzt haben und wohin sie gegangen sind. Damit ist es möglich, Engpässe auf den Wegen und Kreuzungen zu erkennen, Aussagen über die Interessen der Besucher zu machen und eine Besucherstatistik zu führen.

2.2 Zweck des Besucherstrommodells

Ein Hauptziel eines Naturparks ist die Besucherzufriedenheit. Die Besucher sollten sich wohl fühlen, sich erholen können und Ihr Ziel finden, ohne lange zu suchen. Damit dies möglich ist, müssen an den richtigen Standorten Informationstafeln und Wegweiser aufgestellt werden. Um den Park aber nicht mit Informationstafeln zu überschwemmen, müssen die Interessen der Besucher, die sich auch je nach Jahreszeit ändern können, erfasst werden und genau dies kann aus einem Besucherstrommodell gelesen werden.

Neben der Besucherzufriedenheit wird auch der Naturschutz sehr hoch geschätzt. Um sensible Bereiche zu schützen und Konflikte mit zum Beispiel ängstlichen Tieren zu vermeiden, ist eine aktive Besucherlenkung notwendig. Das Besucherstrommodell hilft dem Parkmanager, Probleme in kritischen Gebieten zu erkennen und nach den Lenkungsmassnahmen zu überprüfen, wie sich die Situation entwickelt hat. In den Schweizer Naturpärken sind bisher wenige Erfahrungen über die Auswirkungen von Besucherlenkungsmassnahmen vorhanden. Diese Erfahrung erhofft man sich mit Hilfe des Besucherstrommodells und langjährigen Statistiken zu sammeln.

Ein weiterer Vorteil des Besucherstrommodells ist die Anzeige der räumlichen Verteilung. Damit kann erkannt werden, wo sich die meisten Besucher aufhalten oder ob zum Beispiel einzelne Wege sehr oft oder fast nicht benutzt werden.

Bei neuen Attraktionen oder Änderungen von Standorten will man Informationen bekommen, ob diese von den Besuchern auch vermehrt besucht werden.

2.3 Datenbestand

In den Natur- und Nationalparks werden mit verschiedenen Messsystemen die Besucher automatisch gezählt. Die meisten automatischen Zählungen werden mit sogenannten Zählmatte durchgeführt, welche im Boden vergraben sind und daher täglich die Besucher erfassen. Die automatischen Zählungen können nur auf Wegen durchgeführt werden. Je nach Modell der Zählmatte sind auch Zählungen mit Richtungsangaben möglich.

Im Wildnispark Sihlwald sind 4 Zählmatte im Gebiet verteilt (siehe Abbildung 1: automatische Zählstellen Sihlwald). Im Jahr 2009 sind ab der Kalenderwoche 30 Daten von 3 Zählmatte vorhanden. Während 23 Wochen, in denen die Zählmatte aktiv waren, wurden von allen drei Zählmatte insgesamt 19827 Besucher erfasst. Am meisten Besucher hatte die Zählmatte am Gratweg, mit 12278 Besucher, gezählt. Im Gegensatz dazu gab die Zählmatte am Albishornweg während der gleichen Zeit nur 1187 Besucher an.

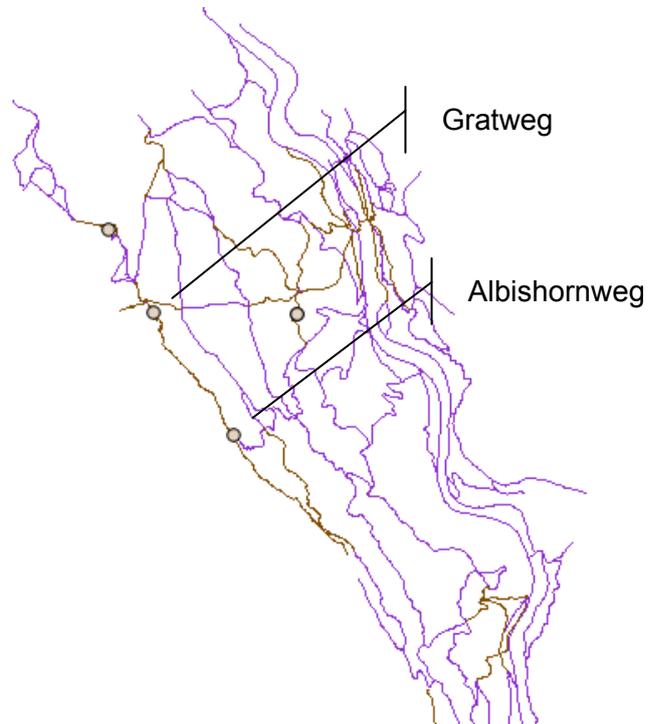


Abbildung 1: automatische Zählstellen Sihlwald

Im Vergleich der Wochentage wird ersichtlich, dass am Wochenende (Samstag und Sonntag) mit 10942 Zählungen mehr Besucher unterwegs sind als unter der Woche mit 8991 Zählungen. Beim Vergleich dieser Zahlen sollte beachtet werden, dass die Wochenendzählung nur 2 Tage beinhaltet. Das Verhältnis der Besucher zwischen Wochentagen und Wochenende beträgt also durchschnittlich 1 zu 3.

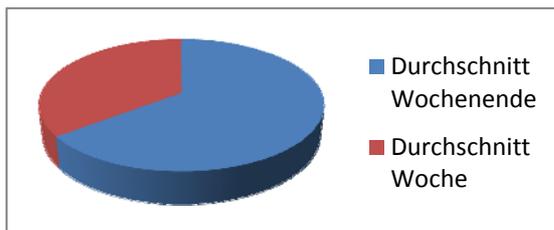


Abbildung 2: Vergleich der Besucherzahlen nach Wochentagen

Zusätzlich zu den Zählmatten wurden manuelle Zählungen durchgeführt. Im Wildnispark Sihlwald wurden diese verteilt während 8 Wochen jeweils Dienstag, Donnerstag, Freitag, Samstag und Sonntag durchgeführt. Die erste Zählung beginnt in der Kalenderwoche 26 und die letzte Zählung stammt von der Kalenderwoche 42. Dabei sind insgesamt 11033 Zählungen erfasst worden. Auch bei der manuellen Zählung sieht man den Unterschied zwischen Wochentagen und Wochenende. Während vom Wochenende 7283 Zählungen vorhanden sind, wurden während den drei Zähltagen unter der Woche nur 3742 Besucher erfasst. Dieser grosse Unterschied darf jedoch nur mit Vorsicht gewertet werden, da am Wochenende mehr Zählungen stattgefunden haben.

Abbildung 4: manuelle Zählstellen Sihlwald

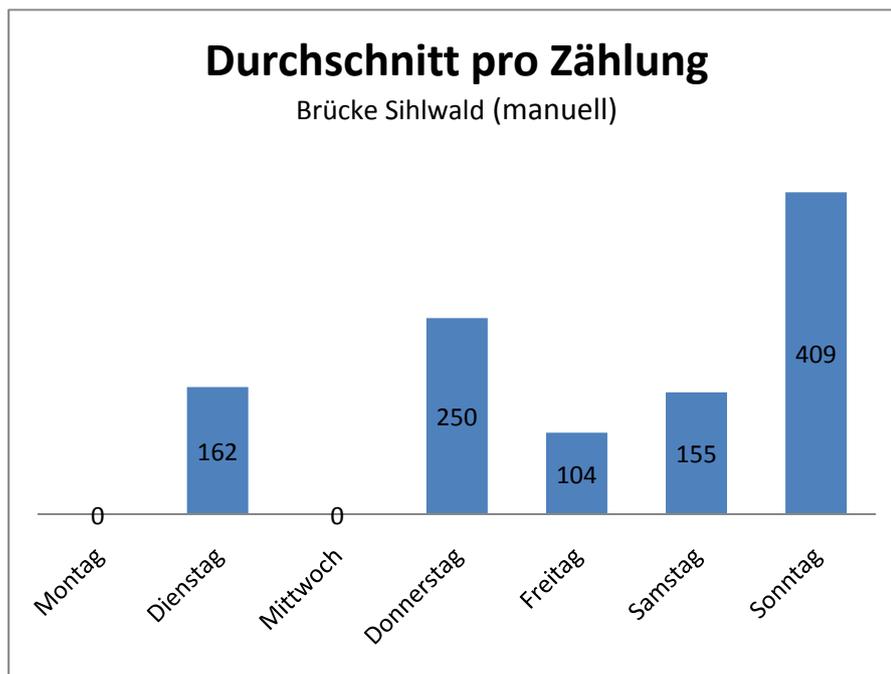
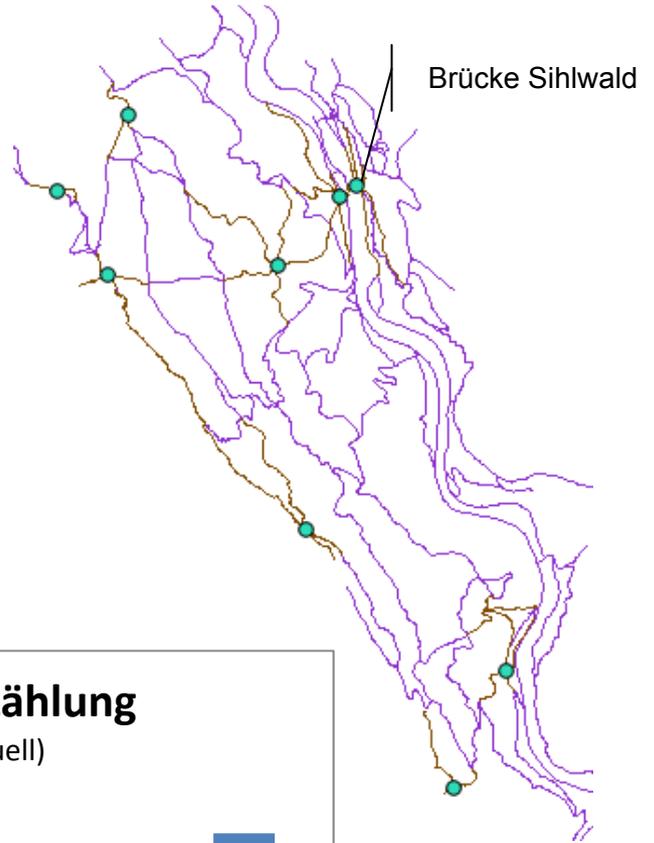


Abbildung 3: Durchschnittliche Besucherzahl pro Wochentag

Das die Interessen der Besucherauswertungen je nach Park unterschiedlich sind, wird aus der Grössen der weiteren Referenzpärke ersichtlich. Der Nationalpark ist flächenmässig sehr gross und Pfyn-Finges ist ein eher kleiner Park. Tanzboden ist ein Park mittlerer Grösse, bei dem man sich auf die Zählung von Velofahrer eingeschränkt hat.

Auch von diesen Pärken sind teilweise Besucherdaten schon vorhanden. In dieser Arbeit wird insbesondere auf den Referenzpark Sihlwald eingegangen, da von diesem Park am meisten Besucherdaten und vollständige Geoinformationsdaten vorhanden sind.

2.4 Abdeckung

Um einen Besucherstrom darstellen zu können, werden Besucherzahlen von verschiedenen Punkten und Wegen benötigt. Die Abdeckung des Gebietes bestimmt, wieviele Punkte erfasst werden. Je höher die Abdeckung des gesamten Gebietes ist, desto weniger muss auf Erfahrungswerte zurückgegriffen werden. Erfahrungswerte werden durch viele Zählungen und guten Kenntnissen der Besucher und des Gebietes erstellt und sind daher vor allem zu Beginn meist noch ungenau.

Die genaue Anzahl Zählstellen für die optimale Abdeckung zu bestimmen ist schwierig, da dies stark vom Gebiet und der Anzahl Wege abhängt. Eine komplette Abdeckung aller Wege ist aus Kostengründen und auf Grund des sehr hohen Aufwands nicht möglich.

Gemäss den Angaben des Herstellers der Zählmatten sind für ein Gebiet, wie es der Wildnispark Sihlwald hat, 5 bis 10 Zählmatten notwendig.

Mit zurzeit 4 Zählmatten liegt Sihlwald also knapp unter den Empfehlungen des Herstellers. Da der Wildnispark viel in die Technik und Erfahrung von Besucherzählungen investiert, kann davon ausgegangen werden, dass andere Parks vorerst eine nicht so hohe Abdeckung mitbringen.

2.5 Auswertung

Bis anhin können die Besucherzahlen aus der Datenbank gelesen werden und zum Beispiel in einer Excel Tabelle dargestellt werden. Eine Auswertung mit Daten aus Tabellen kann jedoch sehr zeitaufwendig sein, da die Zahlen aus verschiedenen Tabellen gelesen werden müssen und es nicht möglich ist, alle übersichtlich in einer Tabelle anzuzeigen. Dazu kommt, dass bei einer Auswertung mit Hilfe von Tabellen gute Kenntnisse über das vorhandene Wegenetz vorausgesetzt werden, um den Besucherstrom zu erkennen, da Kenntnisse über die Verbindungen zwischen den Wegen und Kreuzungen nötig sind.

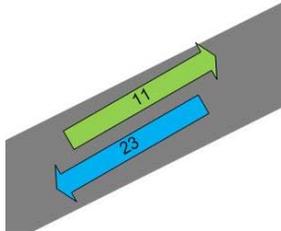
2.5.1 Überlegungen von Marcel Germann

- Die Besucherströme sollen übersichtlich visualisiert werden können. Dazu sollen verschiedene Hilfsmittel gebraucht werden, wie zum Beispiel die Karte oder farbige Diagramme.
- Die vielen Besucherströme sollen verglichen werden können. Das bedeutet, es soll einfach möglich sein die Besucherströme von verschiedenen Zeitperioden miteinander zu vergleichen und somit Unterschiede zu erkennen.
- Um spezifische Auswertungen erstellen zu können, müssen die Daten gefiltert werden. Die Filterung soll nicht nur die Zeit, sondern auch weitere Variablen wie zum Beispiel die Besucherstromgrösse oder unterschiedliche Attribute beinhalten.
- Zusätzliche Attribute, welche bei der manuellen Zählung hinzugefügt werden können, sollen auch in der Visualisierung ersichtlich sein.

2.5.2 Input von Mathias Manz (Projektmitglied VISIMAN)

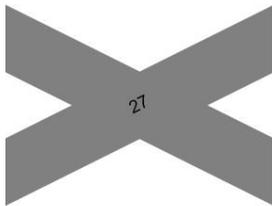
In der Projektgruppe VISIMAN wurde diskutiert welche Besucherströme in einer Auswertung benötigt werden. Man hat sich auf drei verschiedene Ströme geeinigt, welche gebraucht werden, um eine übersichtliche Auswertung zu erstellen.

Auslastung eines Weges



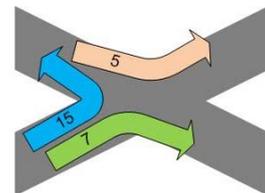
Bei normalen Wegen möchte man wissen, wieviele Besucher insgesamt den entsprechenden Weg passieren. In einer detaillierteren Visualisierung soll auch die Richtung mit angegeben werden.

Auslastung einer Kreuzung



Bei Kreuzungen möchte man wissen, wie deren Auslastung ist, wieviele Besucher über diese Kreuzung gegangen sind.

Strom in einer Kreuzung



Zusätzlich zur Auslastung möchte man wissen, woher die Besucher kommen und wohin sie gehen. Dabei soll angezeigt werden wieviele Besucher woher kommen und wieviele wohin gehen. Eine weitere Möglichkeit wäre es auch, zusätzlich eine Auswertung zu machen, welchen Weg die Besucher wählen, wenn sie aus einer bestimmten Richtung kommen.

Alle drei Visualisierungsarten sollten durch Filter detaillierter angezeigt werden können.

2.5.3 Input von Dominik Siegrist

Die Besprechung mit Herr Siegrist fand am 22. März 2010 statt.

Mit dem Besucherstrommodell soll ein Überblick über die räumliche Verteilung der Besucher und die Auslastung visualisiert werden. Die Richtung der Besucher ist zwar eine interessante Information, jedoch steht die Auslastung im zeitlichen Vergleich im Vordergrund.

Für die Visualisierung sind daher nicht nur Pfeile, sondern auch Balkendiagramme von Interesse. Zudem sollen bei der Visualisierung auch Zonen und POIs (Point of Interest = interessante Orte) dargestellt werden können.

Um Erklärungen für spezielle Besucherzahlen, wie z.B. extrem hohe Werte, direkt herauslesen zu können, wird eine Schnittstelle mit einer Agenda gewünscht. In dieser können Veranstaltungen oder auch Umbauten gespeichert und bei Bedarf angezeigt werden.

Eine Idee wäre es, aus dem Besucherstrommodell eine annähernde Besucherzahl herauszufinden.

Die Visualisierung ist hauptsächlich für das Parkmanagement gedacht. Es ist jedoch in Aussicht, auch eine abgespeckte Version der Öffentlichkeit zur Verfügung zu stellen.

2.5.4 Präsentation „SANU-Workshop“ von Dominik Siegrist

Herr Siegrist hat freundlicherweise seine Präsentation vom SANU-Workshop zur Verfügung gestellt. Aus der Präsentation wird ersichtlich, wozu Besuchermonitoring betrieben wird, was man mit der Besucherlenkung erreichen will und welche Rolle das Besucherstrommodell dabei hat.

2.5.5 Input von Ronald Schmidt

Die Besprechung mit Herr Schmidt fand am 13. April 2010 statt.

Zweck der Visualisierung

Mit der Applikation ArcGIS Desktop ist es bereits möglich, eine Visualisierung der Frequenzen zu erstellen. Dazu wird jedoch viel Erfahrung mit der Applikation benötigt und die Abfragen der Daten müssen manuell zusammengestellt werden. Ziel der Visualisierung sollte eine Applikation sein, die eine solche Abfrage mit den entsprechenden Filtern automatisch zusammenstellt und darstellt. Eines der Ziele ist daher **die Vereinfachung der Visualisierung**.

Ein anderer Wunsch ist **die Berechnung der Gesamtbesucherzahl** im Park. Dazu müssten alle Eingänge zum Park überwacht werden. Im Sihlwald ist dies nicht möglich, da es zu viele Zugänge gibt. Eine Alternative ist aus den Besucherfrequenzen, von den in der Nähe liegenden Zählstellen, Annahmen über die Frequenz bei den weiteren Eingängen zu treffen, um eine Annäherung an die Gesamtbesucherzahl zu erreichen.

Für die Ranger ist besonders die **räumliche Verteilung der Besucher** interessant, da Sie wissen möchten, welche Wege die Besucher am meisten benutzen.

Auch der Besucherstrom soll visualisiert werden. Das heisst man möchte wissen, in welche Richtung wieviele Besucher gehen. Mit dem Besucherstrom ist nicht die Verfolgung der Besucher durch den ganzen Park gemeint, sondern lediglich die Anzahl der Besucher in der entsprechenden Richtung.

Die Visualisierung soll in der Zukunft als Hilfsmittel zur Erstellung eines Besucherstrommodells zur Verfügung stehen. Bereits jetzt ein Besucherstrommodell zu erstellen, macht für Herr Schmidt keinen Sinn, da man noch zu wenig Erfahrung hat und nicht weiss, mit welchen Methoden das Modell automatisch erstellt werden könnte. Zuerst sollte eine gute und einfache Visualisierung existieren, aus der die nötigen Erfahrungen herausgelesen werden können.

Anforderungen an die Visualisierung

- Am wichtigsten ist die Darstellung der Besucherfrequenzen in der Gesamtübersicht.
- An den Kreuzungen ist auch die Richtung der Besucher interessant
- Ein Vergleich von verschiedenen Attributen sollte möglich sein, es ist jedoch nicht zwingend nötig, diese gleichzeitig zu sehen.
- An speziellen Orten (z.B. POIs) möchte man die Besucherzahlen überwachen und die Entwicklung der Zahlen über Jahre hinweg auswerten und dokumentieren.
- Die wichtigste Filterung sind die Attribute und die zeitliche Einschränkung. Wochentage sollten zusammengefasst werden können (zum Beispiel Montag, Dienstag und Donnerstag). Es soll möglich sein zwischen den automatischen und manuellen Zählungen wechseln zu können.
- Die Filterung nach Events und nach Wetter wird nicht benötigt.

Begründung für die Wahl der Zählstellen

Die Zählstellen sind an folgenden Orten platziert worden:

- An stark frequentierten Bereichen
- Bei Übergängen in die Kernzone (Eingänge)
- Bei zu überwachenden Standorten (z.B. POIs)

In Zukunft sind weitere manuelle Zählungen geplant. Die manuellen Zählungen dienen zur Erfahrungserfassung, wie sich die Situation im Park geändert hat.

Weitere automatische Zählstellen werden je nach Erfahrung, die mit der Visualisierung gesammelt wird, an den entsprechenden Orten platziert.

Visualisierung

Für die Visualisierung der Besucherfrequenz wird die Breite der Wege zusammen mit der Zahl als Legende bevorzugt. Wo vorhanden, soll auch die Richtung mit Hilfe von Pfeilen dargestellt werden.

Diagramme sind besonders für die Auswertung der Gesamtbesucherzahlen z.B. bei einem Tages- oder Wochenverlauf interessant. Für den Tagesverlauf genügt eine Quantisierung in Stunden.

Diagramme direkt auf der Karte darzustellen, wäre zwar eine Möglichkeit, die Gefahr besteht jedoch, dass die Visualisierung unübersichtlich wird. Die Funktion ist daher nicht erwünscht.

POIs sollen auf der Karte dargestellt werden können, jedoch soll auch hier beachtet werden, dass die Übersichtlichkeit hohe Priorität hat.

Realisierung

Als Realisierungsmöglichkeit sieht Herr Schmidt besonders eine Webapplikation, da für die Benutzung nichts installiert werden muss und einfach in das vorhandene Administrationswerkzeug integriert werden kann.

2.5.6 Zusammenfassung

Für die Visualisierung soll der Besucherstrom auf einer Karte mit Pfeilen oder Diagrammen angezeigt werden. Zusätzlich sollen weitere Informationen, wie zum Beispiel POIs oder Zonen der Umgebung angezeigt werden. Verschiedene Zeitperioden sollten einfach mit einander verglichen werden können.

Um detailliertere Auswertungen erstellen zu können, soll eine Filterung erstellt werden. Es soll nach Standort / Zonen, nach Zeit, nach entsprechenden Attributen und nach Weg Kategorien (Anzahl Personen) gefiltert werden können.

Ein Ziel der Visualisierung ist es, ein Besucherstrommodell erzeugen zu können. Das Modell wird aus den Erfahrungen, die aus der Visualisierung gelesen werden können, zusammengestellt. Bevor die Erfahrungen vorhanden und visualisiert sind macht es jedoch keinen Sinn, bereits Vorkehrungen für die automatische Erstellung des Modells zu treffen.

Die Erfahrungswerte sollen in der Visualisierung ersichtlich sein. Das Hauptziel bleibt jedoch, dass die Darstellung der Zählraten und die Erfahrungswerte als Hilfe für die Auswertung dienen sollen.

2.6 Erfahrungswerte

Wie in Kapitel 2.2 erwähnt, ist eine vollständige Abdeckung nicht möglich. Das Besucherstrommodell muss also mit zusätzlichen Daten ergänzt werden. Im Wildnispark Sihlwald wurden die Besucher während einer bestimmten Zeit beobachtet und die Wege danach drei verschiedenen Kategorien zugeordnet. Die Kategorien zeigen auf, wie hoch das Besucheraufkommen des Weges ist.

Die Kategorien können mit weiteren Informationen erweitert werden. Führt zum Beispiel an einer Kreuzung ein Weg zu einem Ausgang oder zu einem Parkplatz, kann davon ausgegangen werden, dass prozentual viele Besucher diesen Weg nehmen müssen, da z.B. ihr Auto auf dem Parkplatz steht.

Die Erfahrungswerte stellen Beobachtungen und Annahmen der Ranger und des Parkmanagements dar. Mit den realen Daten kann aufgezeigt werden ob diese Erfahrungswerte auch wirklich stimmen oder ob ein kleiner Weg vielleicht doch stark frequentiert ist. Die Erfahrungswerte sollen wie die Besucherzählungen in das System eingebunden und visualisiert werden können.

3 Funktionale Anforderungen

3.1 Funktionsliste

3.1.1 Notwendige Funktionen

Eingabe

- Daten aus Datenbank (manuelle und automatische Zählung)

Visualisierung

- Karte (z.B. Landkarte)
- Besucherfrequenz eines Weges
- Erfahrungswerte eines Weges
- Richtung des Besucherstromes
- Vergleich zweier Frequenzen mit Graphen (z.B. verschiedengrosse / -farbige Balken)
- Detailinformationen zur Besucherfrequenz mit genauen Besucher- und Erfahrungswerten
- Räumliche Verteilung der Besucher im Gebiet
- Zusätzliche Informationen wie stattgefunden Events oder Wetter

Filterung

Die Visualisierung soll mit folgenden Filtern spezialisiert werden können:

- Standort oder Zonen
- Zeitbereich (von / bis)
- Zeitabschnitt (z.B. Montag, Wochenende, ...)
- Attribute (zusätzliche Informationen)

Möglich wäre auch die Filterung nach:

- Wetter
- Events

3.1.2 Zusätzliche Funktionen

- Eingabe und Visualisierung von Erfahrungswerte (z. B. mit Klassifizierung der Wege)
- Darstellen von verschiedenen Zonen als Ebene in der Visualisierung
- Anzeigen von POIs und zusätzlichen Informationen in der Visualisierung
- Ereignissen und Veranstaltungen zu der entsprechenden Zeit anzeigen
- Berechnen der Gesamtbesucherzahl im Park

3.2 Szenarien

3.2.1 Zeige Auslastung des Aussichtsturms

Der Aussichtsturm wurde umgebaut und der Parkmanager möchte nun wissen, woher die meisten Besucher zu diesem Turm kommen. Er öffnet zwei Karten und wählt in den Filteroptionen für die erste Karte eine Zeitperiode vor und in der zweiten Karte eine Zeitperiode nach dem Umbau ein. Das System holt sich die nötigen Daten aus den Besucherzahlen und den Erfahrungswerten aus der Datenbank. In der Visualisierung erweitert das System nun die Wege mit verschiedenen grossen Balken.

Der Parkmanager erkennt anhand der Grösse der Balken, welche Wege am meisten und welche am wenigsten benutzt werden. In der Detailansicht liest er dann die gesamte und durchschnittliche Besucherzahl, die den den Aussichtsturm besucht haben.

3.2.2 Vergleiche Besucherfrequenz vor und nach den Lenkungsmassnahmen

Der Parkmanager wählt in der Gesamtübersicht den gewünschten Weg aus und wählt in der Filterung die Option Zeitperioden. Das System zeigt ihm eine Eingabemaske für die Vergleichsoptionen. Der Manager wählt nun als ersten Vergleich einen Monat vor und als zweiten Vergleich den Monat nach den Massnahmen aus.

Das System holt nun die Daten der Zählungen aus der Datenbank, verknüpft diese mit den Erfahrungswerten und zeigt beide Optionen als verschieden grosse Balken in der Visualisierung an. Der Parkmanager vergleicht diese Balken und schaut sich die genauen Besucherzahlen und Erfahrungswerte in der Detailansicht an.

3.2.3 Überwache sensible Bereiche

Der Parkmanager wählt den Modus „räumliche Verteilung“ aus und gibt in der Filterung die entsprechende Zeitperiode ein. Das System holt alle Besucherzahlen aus der Datenbank, kombiniert diese mit den Erfahrungswerten und stellt die Verteilung graphisch auf der Visualisierung dar.

In der Gesamtübersicht sieht der Parkmanager kritische Werte in den sensiblen Bereichen und holt sich die Detailansicht des kritischen Standorts. Anhand der effektiven Daten im und um den sensiblen Bereich herum kann er Massnahmen treffen um die Besucher einen anderen Weg benutzen zu lassen.

3.2.4 Eingabe der Erfahrungswerte

Über verschiedene Beobachtungen hat der Parkmanager festgestellt, dass 70% der Besucher an der Kreuzung 3 den breiten Weg in Richtung des Aussichtsturms nehmen. Er geht in die Verwaltung der Erfahrungswerte. Das System zeigt eine Gesamtübersicht des Gebietes, auf welchem der Manager nun die entsprechende Kreuzung und Wege markieren kann. Es geht eine neue Ansicht auf mit Eingabefeldern für die Prozentzahl der Benutzung. Der Parkmanager gibt in das Eingabefeld den berechneten Wert 70% ein.

Das System trägt speichert diese Information in der Erfahrungswertdatenbank.

3.2.5 Festlegen der Wegkategorien

Die Wege im Park sind durch Beobachtungen in verschiedene Kategorien eingeteilt worden. Der Parkmanager erstellt in der Verwaltung der Erfahrungswerte eine neue Wegkategorie und gibt eine Beschreibung dazu ein.

Das System zeigt eine Gesamtübersicht an, worauf die zur Kategorie gehörigen Wege angezeigt werden. Es sind nur die Wege auswählbar, die noch zu keiner Kategorie gehören. Nach der Auswahl speichert das System die Kategorie mit den dazugehörigen Wegen in der Erfahrungswertdatenbank ab.

4 Use Cases

4.1 Aktor Parkmanager

Im Rahmen des Visiman Projektes wurden die Benutzerprofile für das Besuchermonitoring der verschiedenen Referenzpärke durch Mathias Manz analysiert. Dabei wurde festgestellt, dass alle Pärke verschiedene Namen für die gleichen Funktionen haben.

Das Besucherstrommodell wird von allen Profilen benötigt, die im Management des Parks angestellt sind. Zum Management gehören der Direktor, die verschiedenen Bereichsleiter, und die Forscher. Diese Profile haben sehr ähnliche Anforderungen an das Besucherstrommodell, daher werden alle diese Profile zum **Parkmanager** zusammengefasst.

4.1.1 Ziele des Parkmanager

- Möchte die Interessen der Besucher kennen
- Will die räumliche Verteilung der Besucher analysieren
- Möchte aufgrund der Besucherströme Besucherlenkungsmassnahmen erarbeiten
- Will die Entwicklung vor- und nach der Besucherlenkung überprüfen
- Möchte einen Zusammenhang zwischen den Änderungen im Park und den Besucherströmen herstellen

4.1.2 Benötigte Daten

- Genaue Besucherdaten der Standorte
- Durchschnittliche Frequenzen an den Standorten und Wegen
- Direkte Vergleiche mehrerer Frequenzen, Ströme oder Besucherdaten
- Informationen über Standort

4.2 UC 1: Besucherstrom visualisieren

Primärer Aktor

Parkmanager

Vorbedingungen

Die Visualisierungsapplikation ist gestartet und Verbindung zum Server steht. Die Applikation befindet sich in der Gesamtübersicht.

Nachbedingungen

Der Parkmanager kann die gewünschten Informationen herauslesen.

Ablauf

1. Der Parkmanager wählt die Anzeigeart aus
2. Das System zeigt die Filteroptionen an
3. Der Manager wählt die gewünschten Filteroptionen aus
4. Das System aktualisiert die Visualisierung mit den gefilterten Daten

Der Parkmanager kann die Schritte 3 – 4 beliebig wiederholen, bis er alle gewünschten Informationen hat.

Erweiterungen

- 1a) Als Anzeigeeoptionen stehen folgende Optionen zur Verfügung:
- Karte in der Gesamt-, Detailübersicht
 - Statistik mit Zahlen oder Diagrammen
- 2a) Zur Filterung der Daten stehen folgende Optionen zur Verfügung:
- Standort oder Zonen
 - Zeitbereich (von / bis)
 - Zeitabschnitt (z.B. Montag, Wochenende, ...)
 - Attribute (zusätzliche Informationen)
 - evtl. Wetter
 - evtl. Events

4.3 UC 2: Daten eingeben

Primärer Aktor

Parkmanager

Interesse des Parkmanager

Der Parkmanager möchte Informationen über das Gebiet wie z.B. Erfahrungswerte im System eintragen oder ändern.

Die Besucherzählungen werden automatisch vom Server geholt und können in der Visualisierung nicht mehr geändert werden.

Vorbedingungen

Die Visualisierungsapplikation ist gestartet und Verbindung zum Server steht. Die Applikation befindet sich in der Gesamtübersicht.

Nachbedingungen

Die Information ist im System gespeichert.

Ablauf

1. Der Parkmanager wählt die Informationsart aus
2. Das System zeigt je nach Informationsart die entsprechende Eingabemaske an
3. Der Manager füllt diese aus
4. Das System speichert die Informationen und zeigt diese in der Visualisierung an

Der Parkmanager kann die Schritte 1 – 4 beliebig wiederholen, bis alle gewünschten Informationen eingegeben sind.

Erweiterungen

- 1a) Als Informationsarten stehen folgende Optionen zur Verfügung:
- Erfahrungswert
 - evtl. Wetter
 - evtl. Events

5 Nichtfunktionale Anforderungen

5.1 Leistungsanforderung

Die Auswertung der Besucher und das Anzeigen der Visualisierung der Besucherströme sind nicht zeitkritisch. Die Antwortzeiten des Systems müssen daher nicht besonders beachtet werden.

5.2 Randbedingungen

Die Visualisierung wird mit Werten aus zwei verschiedenen Datenbanken gefüllt. Eine Verbindung vom System zu den Servern ist daher zwingend.

Es wird mit bestehenden Daten gearbeitet. Um die Kompatibilität zu gewährleisten, wird mit einem MySQL als Datenbankserver der Messdaten gearbeitet.

Für die Speicherung der Geodaten sind keine speziellen Anforderungen vorhanden. Es kann eine beliebige Technologie verwendet werden.

5.3 Qualität

Ein besonderes Augenmerk soll der **Benutzbarkeit** gewidmet werden. Die Applikation wird zwar von erfahrenen Computerbenutzern bedient, da jedoch verschiedene Mitarbeiter Zugriff auf die Visualisierung haben, soll diese möglichst einfach bedient werden können. Die Bedienelemente sollen selbsterklärend und übersichtlich angeordnet sein.

Die **Richtigkeit der Daten** kann nicht gewährleistet werden, da das Modell mit Erfahrungswerten und Beobachtungen erstellt wird und auch die Zählmatte eine Fehlerquote haben. Die angezeigten Daten sind also Näherungswerte.

5.4 Technologien

Für die Realisierungslösung wird keine besondere Technologie vorgegeben. Die Lösung soll schlussendlich auf verschiedene Parks angepasst werden können, daher wird eine vom System unabhängige Lösung angestrebt.

6 Lösungskonzept

Die Visualisierung wird in drei Konzepten aufgeteilt. Im Kapitel 6.1 wird das Ansichtskonzept erklärt, welches die eigentlichen Visualisierungarten beinhaltet. Das Kapitel 6.2 behandelt das Darstellungskonzept. Darin wird erläutert, wie die Messdaten auf der Visualisierung dargestellt werden können. Das Filterkonzept wird im Kapitel 6.3 verdeutlicht.

6.1 Ansichtskonzept

Das Ansichtskonzept beschreibt, auf welche Art die Besucherdaten dargestellt werden.

6.1.1 Karte

Eine der naheliegendsten Lösungen ist es, die Daten auf einer Karte darzustellen. Dabei kann das gesamte Gebiet (Gesamtübersicht), nur ein Teilbereich (Detailsicht) oder eine der beiden Sichten im direkten Vergleich nebeneinander (Vergleichsansicht) dargestellt werden.

In der **Gesamtübersicht** wird eine Karte des ganzen Parkgebietes dargestellt. Visualisiert werden können die räumliche Verteilung der Besucherfrequenzen und die Kategorien der Werte. Diese Ansicht dient Übersicht und wird direkt nach dem Programmstart angezeigt. Es können weitere Informationen wie Zonen oder POIs dargestellt werden.

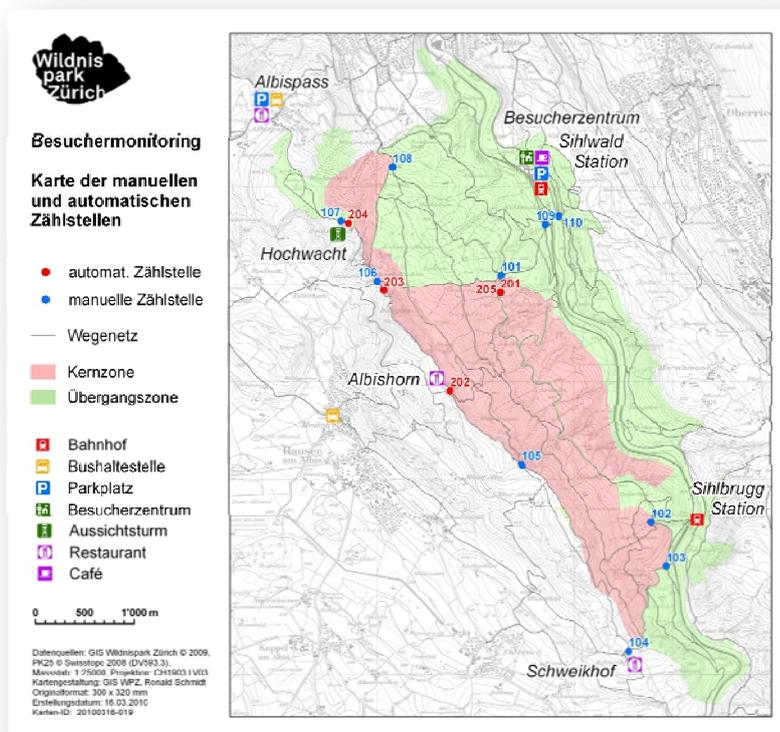


Abbildung 5: Ansichtskonzept Gesamtübersicht (Quelle: Präsentation von Herr Dominik Siegrist)

Der **Teilbereich** stellt nur noch ein Teil des Parks dar. In dieser Ansicht wird mit dem Standortfilter oder der Auswahl eines Bereichs aus der Gesamtübersicht gewechselt. In der Nahansicht kann wie in der Gesamtübersicht die Besucherfrequenz mit detaillierteren Informationen (Daten), sowie auch der Vergleichsmodus ausgewählt werden. Im Vergleichsmodus können verschiedene Filteroptionen miteinander verglichen werden. So kann zum Beispiel die durchschnittliche Besucherfrequenz während den Wochentagen mit denen am Wochenende verglichen werden.

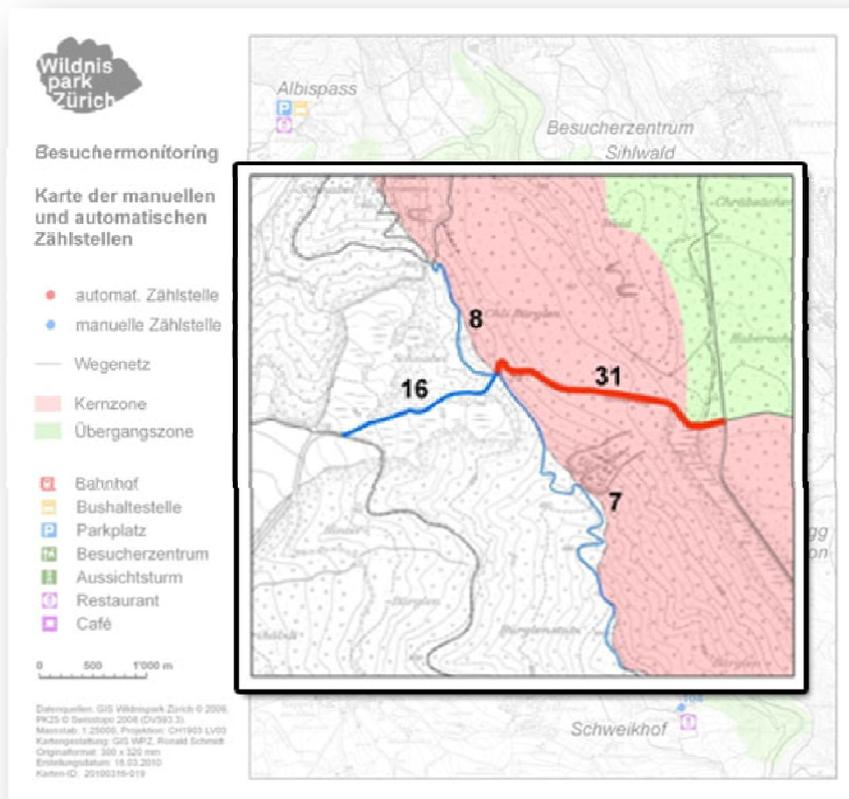


Abbildung 6: Ansichtskonzept Nahansicht (Quelle: Präsentation von Herr Dominik Siegrist)

Die **Vergleichsansicht** stellt mehrere Gesamt- oder Detailsichten gleichzeitig dar.

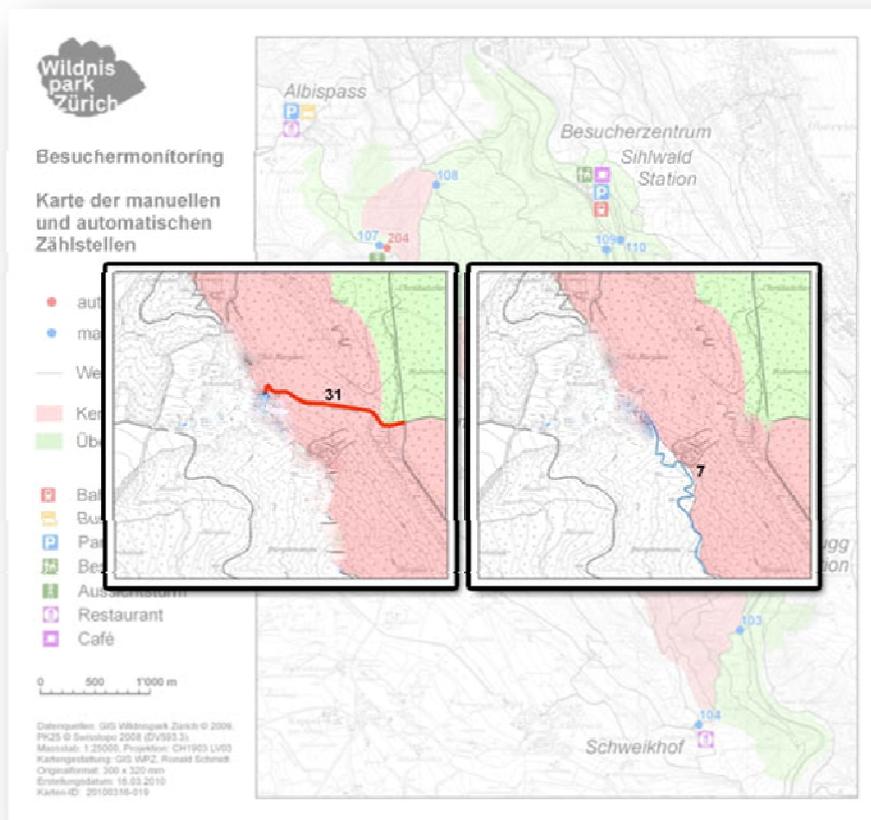


Abbildung 7: Ansichtskonzept Vergleichsansicht (Quelle: Präsentation von Herr Dominik Siegrist)

Der Vorteil dieses Konzeptes liegt in der übersichtlicheren Darstellung, da die Karte nicht überfüllt wird mit grafischen Elementen.

Für jedes Vergleichsattribut wird jedoch ein eigener Kartenausschnitt benötigt, was viel Platz verbraucht. Ein weiterer Nachteil dieses Konzeptes ist der direkte Vergleich. Da die Daten nicht direkt nebeneinander liegen, ist der Unterschied nicht immer klar ersichtlich.

6.1.2 Abstraktion

Statt einer Karte steht auch die Möglichkeit offen, eine Abstraktion zu verwenden. Abstraktionen werden oft für Visualisierungen von Strecken verwendet, die grössere Distanzen beinhalten. Die Information über die Proportionen der Strecken und die genaue Richtung werden dabei entfernt. Somit entsteht eine bessere Übersicht.

Für die Abstraktion wäre jedoch ein zusätzlicher Editor nötig. Eine automatische Erstellung ist nur schwierig möglich, da Wege unterschiedlichste Formen haben, die die Abstraktion beeinflussen. Diese Formen können nicht automatisch erkannt werden.

6.1.3 Statistik

Eine weitere Ansichtsart ist die Statistik. Dabei werden die Daten in Tabellen und Diagrammen dargestellt. In der Statistik können viele Detailinformationen über den Standort oder die Zählungen aufgelistet werden. Zu diesen Informationen zählen zum Beispiel die Anzahl der Zählungen, die bereits stattgefunden haben, durchschnittliche Besucherzahl pro Tag oder auch Beschreibungen über den Standort.

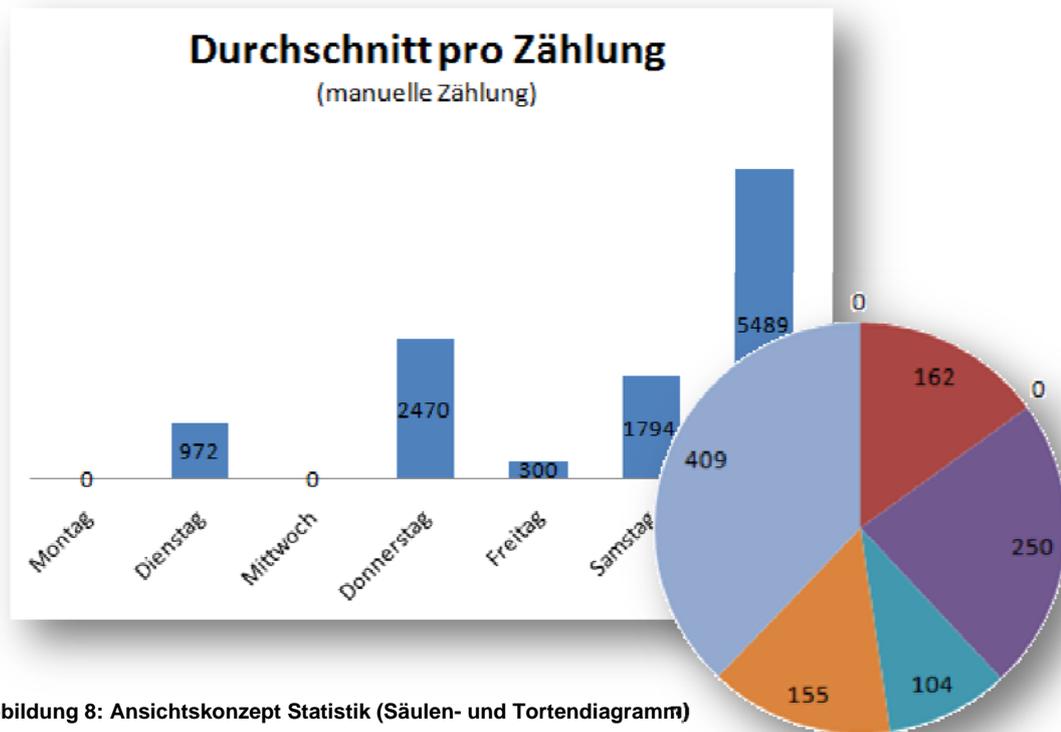


Abbildung 8: Ansichtskonzept Statistik (Säulen- und Tortendiagramm)

6.2 Darstellungskonzept

Das Darstellungskonzept beschreibt, wie die Daten auf der Visualisierung dargestellt werden können. Die Elemente des Darstellungskonzeptes werden daher als eine neue Ebene auf der Visualisierung angezeigt.

6.2.1 Darstellung auf einer Karte / Abstraktion

Auf der Karte können die Daten mit verschiedenen Elementen dargestellt werden.

Am naheliegendsten ist es, die Besucherzahlen gleich als **Zahl** darzustellen. Leider hat dies den grossen Nachteil dass man nicht genau weiss, ob sie den Spitzenwert, den Durchschnitt oder andere Werte darstellt. Weiter ist ein Vergleich nur mit Zahlen sehr unübersichtlich.

Eine übersichtliche Variante ist es, die **Wege mit unterschiedlichen Farben und Breiten** zu zeichnen. Durch die unterschiedlich breiten Wege sind stark frequentierte Wege schnell ersichtlich. Zusätzlich können auch die Zahlen angezeigt werden.

Um mehrere Daten desselben Weges oder Standorts anzuzeigen, ist eine andere Anzeigart nötig. Mit Diagrammtypen wie z.B. **Balken** ist es möglich, mehrere Daten

gleichzeitig darzustellen. Natürlich können auch andere Diagrammtypen wie z.B. Tortendiagramme, etc. verwendet werden. Wenn die Diagramme jedoch zuviel Platz benötigen, wird es schwierig, mehrere Elemente anzuzeigen und man würde besser von Beginn an die Statistik als Ansichtskonzept wählen.

Um den Besucherstrom auch mit Richtung anzuzeigen, kann entweder die Farbe der Elemente oder **Pfeile** statt Striche zur Hilfe genommen werden. Verschiedene Farben sind für den Benutzer nicht als Richtung ersichtlich und müssten in einer Legende speziell erwähnt werden. Pfeile eignen sich jedoch sehr gut, da sie die Richtung anzeigen und gleichzeitig auch mit der Dicke die Frequenz bestimmen lässt.

6.2.2 Darstellung in der Statistik

Die Darstellungselemente in der Statistik sind vielseitig. Zur Anzeige vieler Daten eignen sich **Tabellen**. Diese sind jedoch nur für unterschiedliche Daten übersichtlich, sobald Zahlen verglichen werden müssen, sind grafische Darstellungen besser geeignet.

Für den Vergleich von Besucherzahlen über mehrere Zeitperioden eignen sich besonders **Linien- und Säulendiagramme**, damit wird der zeitliche Verlauf sichtbar. Wenn verschiedene Attribute verglichen werden, wird normalerweise die gleiche Zeitperiode für alle Attribute gewählt und der Verlauf wird somit sinnlos. Mit einem **Torten- oder Balkendiagramm** ist der Unterschied dafür besser ersichtlich. Zur Erkennung von neuen Trends können Netzdiagramme hilfreich sein.

6.3 Filterkonzept

Die Filterung wird zur Eingrenzung der Daten benötigt. Die Filter müssen kumulierbar sein. Filterintern sind die Elemente additiv, das bedeutet wenn zum Beispiel zwei Attribute ausgewählt werden, werden die Zählungen beider angezeigt. Die Anzahl einbezogener Zählungen werden also erhöht.

Die Beziehung zwischen den Filtern jedoch ist ausschliessend. Wenn zum Beispiel ein Zeitbereich ausgewählt wird und verschiedene Standorte ausgewählt werden, werden nur die Zählungen angezeigt, die beide Kriterien erfüllen.

6.3.1 Zeitbereich

Der Zeitbereich wird in zwei Daten unterteilt. Einem von - Datum und einem bis - Datum. Der Zeitbereich ist der wichtigste Filter, da dieser bei fast allen Filterungen verwendet wird.

6.3.2 Standort oder Zonen

In der Standortliste werden alle Wege und Standorte aufgelistet, zu denen es Besucherzählungen oder Erfahrungswerte hat. Somit ist es bereits in der Gesamtübersicht möglich, nur einzelne Zählungen anzuzeigen. Zur besseren Übersicht können die Standorte mit Erfahrungswerten in Zonen unterteilt werden. Wird eine Zone ausgewählt, werden automatisch alle beinhaltenden Standorte selektiert.

6.3.3 Zeitabschnitt

Als Zeitabschnitt werden folgende Listen angezeigt:

- Wochentage (Montag, Dienstag, ...)
- Monate (Januar, Februar)
- Alle Jahre als Auswahlliste
- Täglich von xx:xx bis yy:yy Uhr

6.3.4 Attribute (zusätzliche Informationen)

Wenn viele verschiedene Zählungen gemacht worden sind, kann die die Attributliste unübersichtlich werden. Eine Variante, um nicht alle Attribute anzeigen zu müssen, ist eine Livesuche zu implementieren. Sobald ein Buchstabe in die Suche eingetragen wird, werden die ersten Attribute zur Auswahl gegeben.

Um mehrere Attribute auswählen zu können, werden die ausgewählten Attribute in einer Liste nebeneinander angezeigt.

6.3.5 Wetter (evtl.)

Das Wetter ist eine vordefinierte Liste, die als Mehrfachauswahl angezeigt wird. Die Werte für die Wetterliste können aus der Messdatenbank herausgelesen werden.

6.3.6 Events (evtl.)

Die Eventfilterung ist ein erweiterter Zeitbereichfilter, der Event kann mittels einer Volltextsuche gesucht werden. Nach Auswahl des entsprechenden Filters wird automatisch der Zeitbereich des Events übernommen.

7 Realisierungskonzept

Die Entscheidung, welche Realisierungsmöglichkeit in einem Prototyp implementiert wird, wird anhand von verschiedenen Kriterien entschieden. Die Kriterien sind speziell auf zeitliche Einschränkungen und die Ressourcen dieser Bachelorarbeit angepasst worden.

Die Gewichtung wird mit Zahlen von 1 bis 10 beschrieben, wobei 1 für „nicht relevant“ und 10 für „sehr entscheidend“ steht.

7.1 Kriterien

Name	Beschreibung	Gewichtung
Anschaffungskosten	Anschaffungskosten des Produkts	4 ¹⁾
Lizenz	Lizenz des Produkts. Ist die kommerzielle Nutzung erlaubt?	3
Dokumentation	Qualität und Umfang der Dokumentation	8
Unterstützung	Unterstützung durch Hersteller oder Drittpersonen mit Beispielen oder Diskussionsforen	6
Einarbeitungsaufwand	Aufwand der investiert werden muss, um das Produkt verstehen und anwenden können	5
Erweiterung	Möglichkeit das Produkt durch eigene Implementierungen zu erweitern	10
Unterstützte Erweiterungen	Programmiersprachen und Technologien, die für Erweiterungen verwendet werden können	7
Schnelligkeit	Geschätzte Reaktionszeiten für die Nutzung der Erweiterung und um die Auswertung anzuzeigen	5
Datenanbindung	Unterstützte Geodatenformate und Geodatenbankserver	8

- 1) Die Anschaffungskosten sind für die Erstellung des Prototyps nicht sehr entscheidend, da die Schule meist kostenlose Schullizenzen bekommt.

7.2 GIS Erweiterungen

7.2.1 ArcExplorer

Der ArcExplorer ist ein in Java geschriebener Geodaten Viewer, der kostenlos von ESRI (dem Hersteller von allen ArcGIS Produkten) bereitgestellt wird. Mit dem ArcExplorer ist es möglich Shapefiles zu importieren und auch Abfragen von Daten werden unterstützt.

Nachteil

Leider kann der ArcExplorer nicht erweitert werden, daher wird dieser nicht in Frage kommen.

7.2.2 Google Earth

Mit Google Earth stellt Google eine GIS Applikation für die geschlossenen Karten von Google Maps zur Verfügung.

Nachteil

Google Earth lässt sich zwar mit Punktdaten und eigenen Layern erweitern, jedoch können keine Shapefiles importiert werden und auch die Verbindung zu eigenen Geodatenservern ist nicht möglich. Daher ist auch Google Earth nicht brauchbar.

7.2.3 ArcGIS

ArcGIS ist eine Sammlung von vielen GIS Applikationen vom Hersteller ESRI. ArcGIS Applikationen sind kostenpflichtig und jedes Tool muss teuer lizenziert werden. Es eignet sich daher nicht für Pärke mit kleinem Budget.

Die Applikation ist in diverse Teilapplikationen aufgeteilt. Für diese Arbeit ist besonders der ArcGIS Server und ArcMap interessant.

ArcMap ist die zentrale Anwendung für die Kartendarstellung, die Analyse und das Editieren. Die Daten für die Anwendung können aus verschiedenen Quellen stammen. Die wichtigsten Quellen sind der Import über Shapefiles, die Verbindung über eine SDE (Spatial Data Engine) zu einer Oracle oder Postgres Datenbank oder die Geodatenbank, welche dateibasiert mit einer Microsoft Access (mdb) Datei oder ordnerbasiert realisiert werden kann.

Der **ArcGIS Server** kann die Daten aus den gleichen Quellen wie ArcMap holen und damit eine Auswertung erstellen. Im Gegensatz zu ArcMap wird die Auswertung jedoch nicht auf einer Karte dargestellt. ArcGIS Server kann also zur Datenbeschaffung benutzt werden, wenn eine weitere Applikation für die Kartendarstellung verwendet wird.

Nachteile

Die Installation und Konfiguration von ArcGIS Anwendungen ist sehr komplex und zeitaufwendig. Alleine für die Installation und Konfiguration des ArcGIS Servers werden nach Angaben von Herrn Stefan Keller ca. 3 Tage benötigt.

Erweiterungen für ArcGIS werden mit den Programmiersprachen Python und C# erstellt. In Java, wie auch mit VBA Makros sind zwar ebenso Erweiterungen möglich, VBA wird aber von Microsoft nicht mehr empfohlen und Java wird von Herrn Keller aufgrund schlechter Unterstützung und vielen Fehlern in der Implementation nicht empfohlen.

7.2.4 Quantum GIS

Quantum GIS, oder abgekürzt auch QGIS genannt, ist ein kostenloses Open Source GIS und wird von der OSGeo (Open Source Geospatial Foundation) unterstützt. QGIS kann mit eigenen Plugins erweitert werden. Die Plugins werden mit Python geschrieben.

QGIS hat eine grosse Anhängerschaft, die Entwicklung von Plugins wird daher gut beschrieben und Probleme werden in Entwicklerforen gelöst.

7.3 Desktop Applikation

Eine weitere Realisierungsmöglichkeit ist die Erstellung einer eigenen Applikation. **Java 2D** bietet viele Funktionen zur Darstellung von Vektordaten. ArcGIS stellt eine Java Bibliothek zur Verfügung, mit der eine Verbindung zu einem ArcGIS Server möglich ist.

Vorteil

Die Entwicklung einer eigenen Anwendung ist besonders für spezielle Anforderungen interessant, da es keine Beschränkungen durch verwendete Frameworks gibt.

Nachteil

Im Gegensatz zu den GIS Erweiterungen und zu Web-Anwendungen muss die ganze Benutzeroberfläche selbst erstellt werden und einige Methoden, die bei allen anderen Realisierungen zur Verfügung stehen, müssen selber programmiert werden. Zudem wird eine eigene Anwendung nicht schneller sein, als z.B. eine Webapplikation, da die Weblösungen bereits existieren und gut optimiert worden sind. Die eigene Applikation benötigt eine separate Installation auf den Computern und ist auch abhängig von den vorhandenen Ressourcen auf dem Computer.

7.4 Webapplikation

7.4.1 Begriffe und Erklärungen

OGC – Open Geospatial Consortium

Das Open Geospatial Consortium ist eine non-profit Organisation, die es sich zum Ziel gesetzt hat, die Entwicklung von Geospatial und Standortdiensten zu überwachen und zu standardisieren.

Basislayer

Der Basislayer (engl. Bezeichnung: base layer) ist die Grundebene für die Darstellung eines GIS. Meist wird eine Landkarte oder fotografische Aufnahmen dazu verwendet. Es kann nur ein Basislayer auf einmal angezeigt werden. Eine Auswechslung des Basislayer ist jedoch problemlos möglich

Themenlayer

Die Themenlayer (engl. Bezeichnung: overlays) werden über den Basislayer gesetzt. Als Themenlayer können alle möglichen Vektor- und Rasterdaten verwendet werden. Themenlayer werden zum Beispiel benötigt, um Zonen und Wege anzuzeigen. Auch Punkte (z.B. POIs) können damit auf einem Basislayer angezeigt werden.

WMS – Web Map Service

Schnittstelle zum Abrufen von Landkarten übers Internet. Kombiniert Vektor und Rasterdaten und liefert ein statisches Kartenbild.

WFS – Web Feature Service

Internetgestützter Zugriff auf Geodaten von einem GIS. Beschränkt sich auf Vektordaten. Liefert Daten in ursprünglicher Semantik. Somit ist die clientseitige Weiterverarbeitung und auch Darstellung möglich.

WFS-T – Transactional WFS

WFS-T ist eine Erweiterung zum WFS und ermöglicht eine bi-direktionale Verbindung zur Geodatenbank. Das bedeutet, Geodaten können verändert und wieder in die Geodatenbank zurückgeschrieben werden.

WCF – Web Coverage Service

Liefert selektierte Rasterdaten mit detaillierten Beschreibungen. Erlaubt komplexe Abfragen und liefert Daten in ursprünglicher Semantik. Somit ist die clientseitige Weiterverarbeitung möglich.

SLD – Styled Layer Description

Mit SLD ist es möglich, die Darstellung der Ebenen zu bearbeiten. Es können Attribute wie Farbe, Rahmen, Dicke, ... angegeben werden. Ausserdem können Ebenen ab bestimmten Zoomstufen ein- oder ausgeblendet werden.

Die SLD wird in einem XML Dokument erstellt und kann entweder einem WMS Request mitgegeben oder direkt auf Vektorebenen angewendet werden.

7.4.2 Logik auf Client

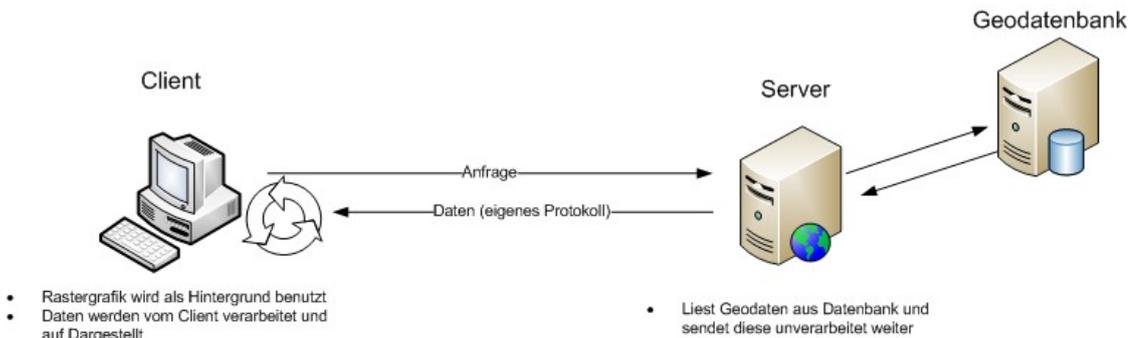


Abbildung 9: Realisierungskonzept - Logik auf Client

Die einfachste Möglichkeit einer Webapplikation ist, die ganze Logik auf den Client zu verlagern. Als Basislayer wird eine normale Bilddatei verwendet (Rastergrafik). Der Client schickt eine Anfrage an den Server, welcher lediglich die gewünschten Daten aus der Datenbank liest und an den Client zurückschickt. Die Übertragung zwischen Client und Server kann mit einem beliebigen Protokoll und Technologie erfolgen. Der Client verarbeitet die Daten und zeigt diese über dem Basislayer an.

Diese Variante ist zwar einfach zu realisieren, benötigt aber einen grossen Aufwand, da der Client sehr viele Aufgaben übernimmt. Die Geschwindigkeit der Anzeige ist abhängig von den Ressourcen des Clients. Es wird jedoch keine schnelle Anzeige erwartet, da die Resultate nicht zwischengespeichert werden können und Clientprogrammiersprachen wie Beispielsweise Javascript nicht sehr schnell sind.

7.4.3 Serverseitige Java Anwendung

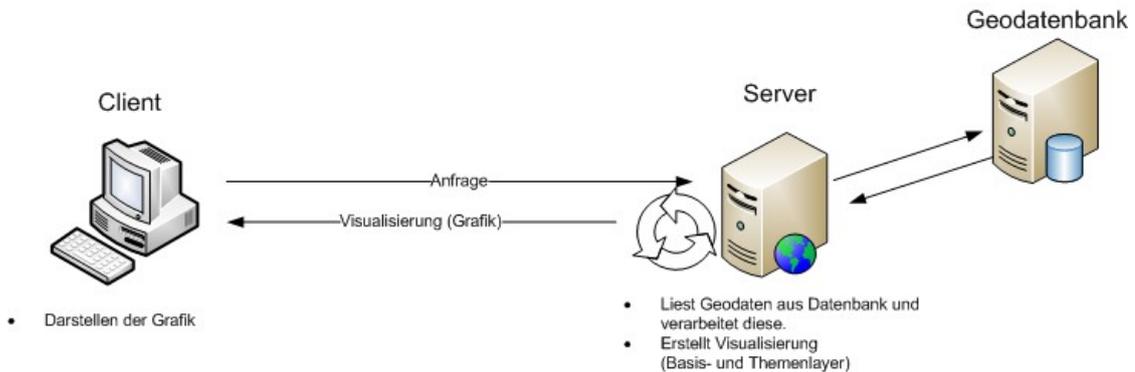


Abbildung 10: Realisierungskonzept - Serverseitige Java Anwendung

Die zweite Möglichkeit ist ähnlich wie die erste, jedoch wird die Verarbeitung der Daten komplett auf den Server ausgelagert. Der Client übernimmt nur die Anzeige der Visualisierung und der Bedienelemente.

7.4.4 WPS (Web Processing Service)

Ein Web Processing Service ermöglicht es, über das Internet räumliche Analysen von Geodaten zu erstellen. Der Basislayer wird wieder aus einer Rastergrafik erstellt, die Daten für den Themenlayer werden jedoch durch den WPS Server zusammengestellt. Als Server kann zum Beispiel PyWPS benutzt werden. PyWPS ist eine in Python erstellte Implementation von WPS. Unterstützt wird die Verbindung zu Geodatenbanken (empfohlen wird PostGIS) und der Import von Shapefiles.

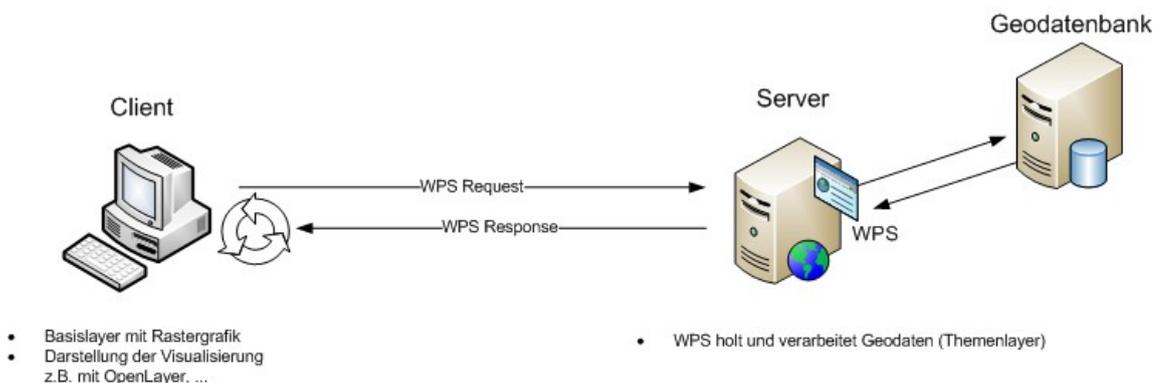


Abbildung 11: Realisierungskonzept - WPS

Die Daten werden über ein durch das OGC standardisiertes Protokoll übertragen und müssen vom Client selber dargestellt werden.

PyWPS kann mit zusätzlichen Paketen erweitert werden. Dadurch lässt sich eine ähnliche Funktionalität wie die des ArcGIS Server erreichen.

7.4.5 WMS und WFS

Die im Internet meist verwendete Realisierung ist mit Hilfe von WMS und WFS Servern. Der Client übernimmt dabei die Darstellung der Karte und der Bedienelemente.

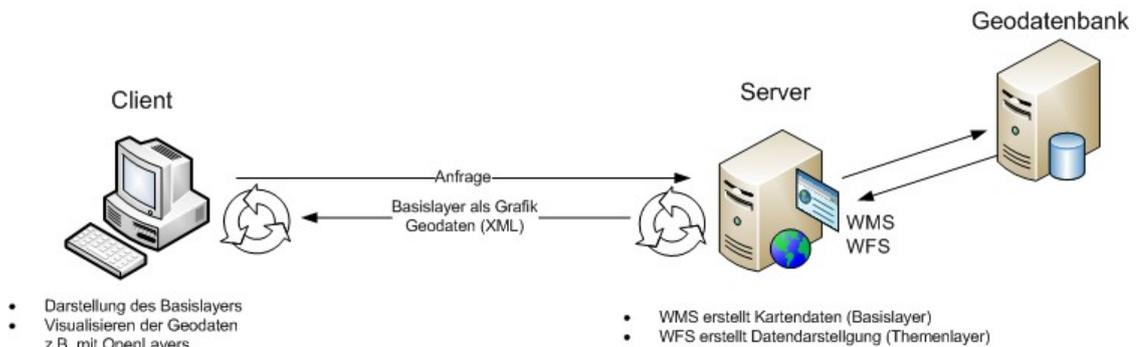


Abbildung 12: Realisierungskonzept - WMS und WFS

Der Basislayer wird über einen WMS und der Themenlayer entweder ebenso über WMS oder über einen WFS erstellt. Sowohl für den Server wie auch für den Client sind viele verschiedene Implementierungen vorhanden mit einer grossen und unterschiedlichen Auswahl an Funktionen.

Der Vorteil bei dieser Variante ist die Flexibilität. Die Übertragung der Daten findet mit standardisierten Formaten statt. Daher ist es möglich, verschiedene und auch eigene Implementierungen zu verwenden. Auch die Vermischung von Produkten ist möglich. Als Server kann zum Beispiel das kostenpflichtige Produkt ArcGIS Server benutzt werden und für den Client wird die OpenSource Variante OpenLayers verwendet. Bei einem Wechsel des Servers sind durch die Standardisierung auf dem Client keine Änderungen nötig.

7.5 Auswertung

Da für die Visualisierung eine eigene Darstellung nötig ist, werden nur Möglichkeiten in die Auswertung übernommen, die erweiterbar sind.

Name	Gewichtung	GIS Erweiterung QGIS	GIS Erweiterung ArcGIS	Java Desktop Applikation	Web: Javascript Logik auf Client	Web: PyWPS	Web: WMS & WFS Map/GeoServer	Web: WMS & WFS ArcGIS Server
Anschaffungs- kosten	4	keine	sehr hoch	keine	keine	keine	keine	sehr hoch
Lizenz	3		\$	-	-	OS	OS	\$
Dokumentation	8	+	+	--	--	-	++	+
Unterstützung	6	+	+	-	-	+	++	+
Einarbeitungs- aufwand	5	hoch	sehr hoch	hoch	nor- mal	hoch	normal	sehr hoch
Unterstützte Erweiterungen	7	Python	C#, .NET	Java	- (Web)	- (Web)	- (Web)	- (Web)
Schnelligkeit	5	++	+	--	-	0	0	0
Daten- anbindung	8	+	0	-	--	0	+	+
Ergebnis		+	-	--	--	0	++	+

Legende:

- \$ Kostenpflichtig
- OS OpenSource

Webapplikationen sind flexibel, erweiterbar und werden gut unterstützt durch Internetforen, verschiedene Dokumentationen und Beispiele. Die Technologien WMS und WFS schneiden in der Dokumentation und Erweiterbarkeit am besten ab.

Eine Evaluation, welche Implementation des WMS und WFS Servers sich besser für den Prototyp eignet, wird im nächsten Kapitel 8 „Evaluation“ durchgeführt.

8 Evaluation Technologien

8.1 Server

Als Server wird eine Implementation gesucht, die WMS und WFS bietet.

8.1.1 UMN MapServer

Der MapServer ist eine sehr einfache und verständliche Implementation. Die Konfiguration ist auf die nötigen Funktionen beschränkt und einfach zu verstehen. Es stehen viele Datenquellen zur Verfügung. Eine grosse Einschränkung des MapServer ist, dass Geodaten nur gelesen werden können. Das Schreiben von Änderungen wäre nur indirekt über die Datenbank möglich.

8.1.2 GeoServer

Der GeoServer besitzt ähnliche Funktionen wie der MapServer. Es stehen jedoch mehr Funktionen zur Verfügung, unter anderem auch das Schreiben in die Geodatenbank, und ist daher ein bisschen komplizierter in der Konfiguration. Der GeoServer bietet zudem weitere Darstellungsformate als der MapServer.

8.1.3 ArcGIS Server

Der ArcGIS Server ist ein sehr grosser und umfangreicher Geoserver. Für die Installation und Konfiguration wird viel Zeit benötigt. Der Einsatz von ArcGIS Server wird besonders empfohlen, wenn auch andere Produkte von ArcGIS verwendet werden.

8.2 Client

Der Client soll WMS und WFS Anfragen erstellen und die Antwort des Servers darstellen können. Ausserdem wird ein Framework gesucht, welches die Erstellung einer Benutzeroberfläche vereinfacht.

8.2.1 OpenLayers

OpenLayers ist eine Javascript-Bibliothek, die es ermöglicht, Geodaten im Webserver anzuzeigen. Unterstützt verschiedene Schnittstellen, unter anderem auch die vom OGC standardisierten Formate WFS und WMS. Es können aber auch geschlossene Daten wie GoogleMaps, Virtual Earth, etc. eingebunden werden.

OpenLayers ist eine der meistverwendeten Implementationen für die Darstellung von Karten im Internet. Damit ist auch die Unterstützung durch dritte gegeben.

8.2.2 GeoExt / ExtJS

GeoExt verbindet die Funktionalität von OpenLayers mit dem Benutzerinterface von ExtJS. ExtJS ist eines der grössten Javascript Frameworks für Rich Internet Applications. Es ist möglich, Fenster darzustellen, die für den Benutzer keinen optischen Unterschied mehr zwischen Desktop und Web Applikation machen.

8.2.3 MapFish

MapFish ist ein Framework zur Erstellung von Web-Mapping Applikationen. Es baut auf GeoExt und somit auf OpenLayers sowie der mächtigen Bibliothek ExtJS auf. Für die Serverseite sind Implementierungen in Ruby/Rails und in PHP/Symfony vorhanden. MapFish implementiert die Client- wie auch die Server-Komponente

Zur Erstellung von Web Applikationen mit MapFish sind Kenntnisse in HTML, Javascript, einer unterstützten Skriptsprache auf dem Server, sowie des MapFish-eigenen API Voraussetzung.

Als Datenbankkomponenten werden von MapFish die meisten SQL Varianten wie zum Beispiel Postgres, MySQL, MS-SQL, SQLite, usw. unterstützt.

8.2.4 MapBender

MapBender gehörte zu den ersten Frameworks zur Darstellung von Maps im Web. Während MapFish auf Ext JS als Javascript Framework setzt, baut MapBender auf der schlanken Bibliothek jQuery auf. Auch MapBender bietet einen Client und Server in einem.

MapBender lässt sich mit sogenannten Widgets erweitern. Eine Visualisierung kann somit ohne Programmierkenntnisse in einem Administrationsinterface von MapBender erstellt werden. Vorausgesetzt werden die Kenntnisse in die eigene Anwendungslogik von MapBender.

Als Datenbankkomponenten werden PostgreSQL und PostGIS unterstützt.

8.2.5 Ka-Map

Ka-Map ist eine weitere OpenSource Implementation von MapTools.org. Ka-Map wurde in der Diplomarbeit Open Map Sihlwald (<http://www.openmapsihlwald.ch>) verwendet. Es ist ein junges Produkt, das 2005 begonnen hat und im Februar 2007 die Version 1.0 (zurzeit aktuellste) Version herausgebracht hat. Die Funktionen sind dadurch nicht soweit fortgeschritten wie z.B. bei OpenLayers und auch Beispiele und Dokumentationen sind nur wenige vorhanden.

Speziell an Ka-Map ist, dass die Verarbeitung der Daten auf dem Server geschieht. Dadurch eignet es sich nicht für dynamisch änderbare Karten und ist somit für diese Arbeit nicht geeignet.

8.2.6 Unterschied zwischen OpenLayers/GeoExt und MapFish/MapBender

OpenLayers und GeoExt sind Frameworks für die Darstellung von Kartendaten auf dem Client. Das bedeutet, man hat sich auf die Benutzeroberfläche und die Darstellung der Karte spezialisiert.

MapFish und MapBender zwingen sich erst bei grösseren Applikationen auf, denn neben den Javascript Bibliotheken für den Client sind auch Implementierungen auf dem Server nötig. Durch die zusätzliche Verarbeitung auf dem Server können Abfragen schneller verarbeitet werden. Dafür ist ein wesentlich höherer Aufwand zur Erstellung einer Applikation nötig und es werden Vorkenntnisse in weiteren Frameworks und APIs vorausgesetzt.

8.3 Auswertung Technologien

Für die Implementierung des Prototyps wird auf der Serverseite der UMN MapServer zusammen mit der Erweiterung PHP/MapScript verwendet. Der MapServer unterstützt alle benötigten Funktionen und ist im Gegensatz zum GeoServer und zum ArcGIS Server einfach zu konfigurieren. Auch die Dokumentation ist ausführlich und es lassen sich viele Beispiele im Internet und in Büchern finden.

Als Client wird ExtJS als Framework für die Benutzeroberfläche mit OpenLayers für die Darstellung der Karte verwendet. OpenLayers ist das beliebteste Framework für WebGIS Anwendungen und somit existieren viele Dokumentationen und Tutorials im Internet. Auch die Beispiele auf der offiziellen Homepage sind ausführlich und zeigen die grosse Funktionenvielfalt von OpenLayers.

Mit GeoExt können Karten einfach in ExtJS dargestellt werden. ExtJS eignet sich ebenso gut für die Erstellung von Webapplikationen. Das Framework benötigt zwar einen hohen Einarbeitungsaufwand, doch bietet es eine grosse Vielfalt an Layouts, Bedienfelder und Erweiterungen wie zum Beispiel für Diagramme und Formulare.

MapFish und MapBender eignen sich nicht für den Prototyp, da der Einarbeitungsaufwand in alle verwendeten Frameworks und Technologien zu gross und die Erweiterbarkeit eingeschränkt ist.

9 Verknüpfung der Mess- und Geodaten

Der Basislayer kann aus verschiedenen Quellen erstellt werden. Google Maps wie auch OpenStreetMap (OSM) stellen ihre Karten frei zur Verfügung. Möglich ist auch die Einbindung einer Grafik mit WMS. Der Basislayer ist unabhängig vom Themenlayer und lässt sich ohne grossen Aufwand auf dem Client austauschen.

Die Darstellung des Themenlayer ist komplizierter, da die Daten nicht einfach dargestellt werden können, sondern zuerst mit den Messdaten verknüpft werden. Je nach Messdaten bekommen die Wege einen anderen Style, das bedeutet, eine bestimmte Farbe und Breite.

In den nachfolgenden Unterkapiteln werden 4 Möglichkeiten zur Darstellung des Themenlayer aufgezeigt. Die Möglichkeiten unterscheidet sich in den verwendeten Technologien und in der Seite wo die Messdaten mit den Geodaten verknüpft werden (Client- oder Serverseitig).

9.1 Veränderbare Attributtabelle durch Datenbankview

Die erste Variante für die Erstellung des Themenlayer ist die Nutzung von MapScript. MapScript ist eine Schnittstelle von MapServer für Skriptsprachen, wie zum Beispiel PHP oder Python. Es ist damit möglich, Anfragen an den MapServer zu senden.

In den Geodaten ist es möglich, neben den Vektordaten auch weitere Attribute zu speichern. Beim Erstellen einer Karte kann die Darstellung abhängig von den Attributen verändert werden. Damit dies automatisiert möglich ist, werden die Attribute zuerst in eine Attributtabelle gespeichert und eine View mit den Geodaten und den entsprechenden Attributen erstellt. Der MapServer holt bei einer Anfrage die Daten nicht direkt aus der Geodatenbank, sondern aus der entsprechenden View.

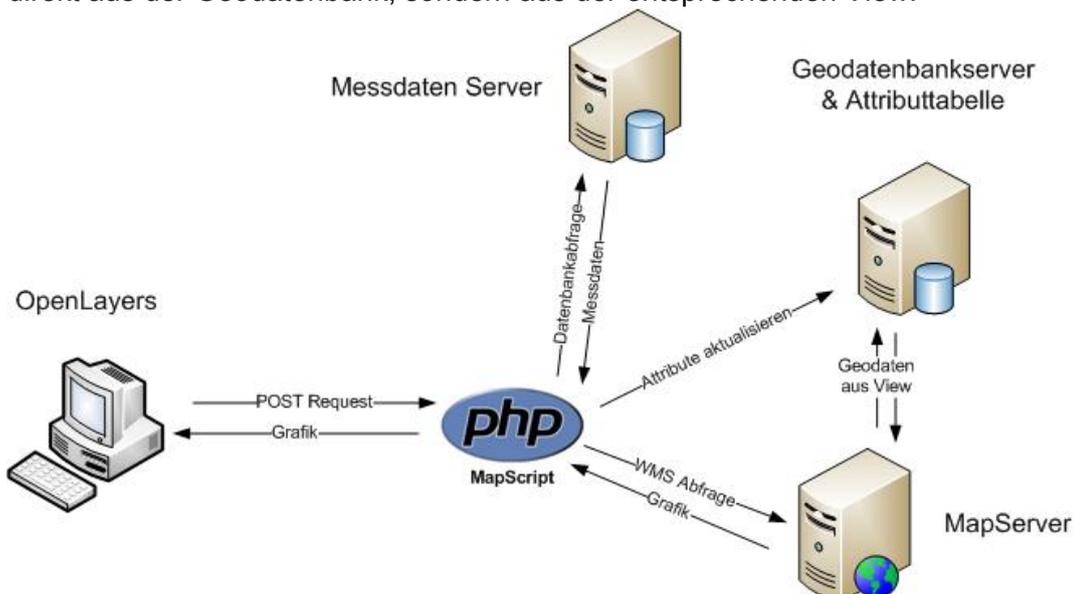


Abbildung 13: Verknüpfung Mess- und Geodaten durch eine Attributtabelle und View

Nachteil dieser Methode ist, dass relativ viele Daten temporär angelegt werden. Somit entstehen viele Schritte und die Kartenerstellung wird verlangsamt. Zudem muss eine Geodatenbank eingesetzt werden. Der Einsatz von Shapefiles wird nicht unterstützt.

9.2 Verändern der Konfiguration

Die zweite Variante ist ähnlich der Ersten. Jedoch werden keine Attributdaten in die View integriert. Um die Wege in verschiedenen Farben und Breiten darzustellen, wird mit MapScript die Konfiguration bearbeitet.

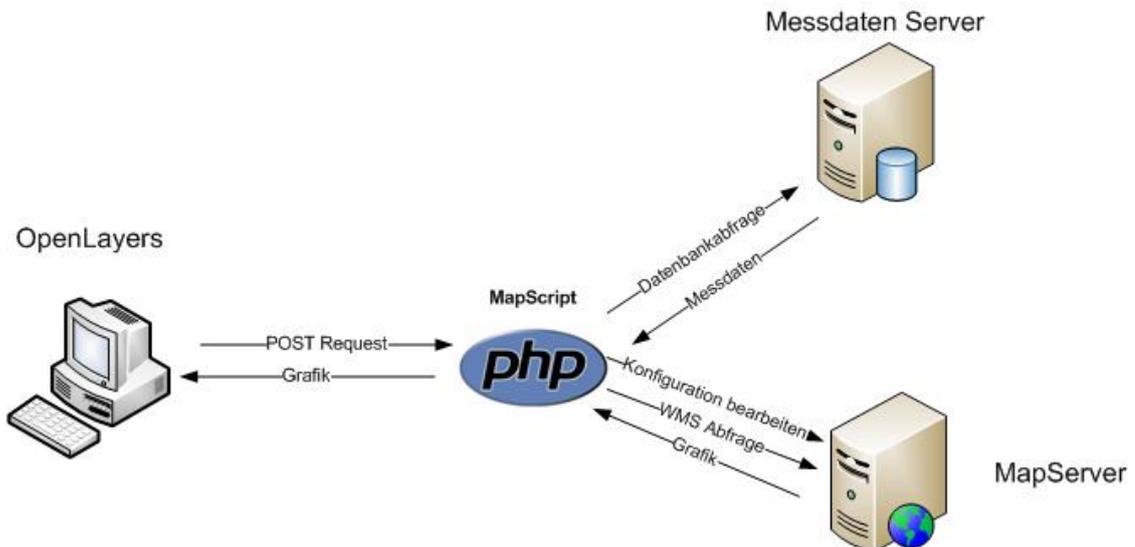


Abbildung 14: Verknüpfung Mess- und Geodaten durch ändern der Konfiguration

In der Variante mit MapScript und der dynamischen Konfiguration wird eine selber generierte POST Anfrage an den Server geschickt. Diese wird mit einer Skriptsprache, wie zum Beispiel PHP, verarbeitet und eine entsprechende Datenbankabfrage an den Messdaten Server gesendet. Die im Ergebnis zurückkommenden Messdaten werden vom Skript wieder verarbeitet und die Wege mit entsprechenden Styleangaben in einem Map-Konfigurations-Objekt gespeichert. Dieses Objekt kann entweder eine Datei, das MapFile, sein oder direkt als Objekt über MapScript dem MapServer übergeben werden.

Über MapScript wird danach die Abfrage an den MapServer gesendet, welcher über den WMS die Karte als Grafik zurückliefert.

9.3 Konfiguration durch SLD verändern

Anstatt die Konfigurationsdatei direkt zu bearbeiten, ist es möglich, die Konfiguration mit einem Style zu verändern. Der Style wird in einem XML Format beschrieben und kann entweder in einer Datei abgespeichert oder direkt mit den Parametern mitgegeben werden.

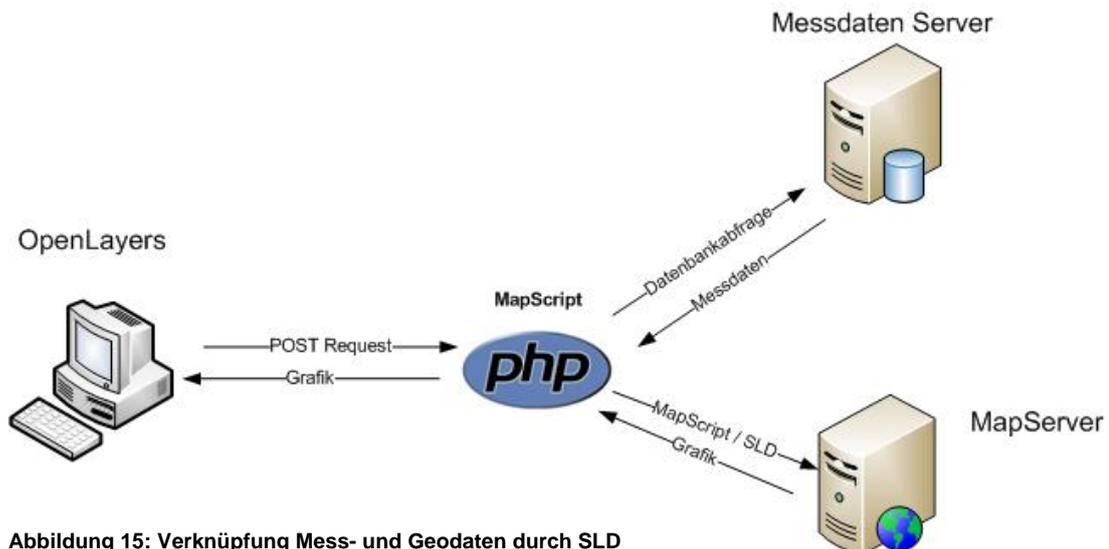


Abbildung 15: Verknüpfung Mess- und Geodaten durch SLD

Durch die Verwendung von Styles ist es möglich, die Kartengenerierung auch von einem anderen MapServer durchführen zu können. Somit können auch Karten von anderen Servern im Aussehen verändert werden, bei denen man keinen Zugriff auf die Konfiguration hat.

9.4 Berechnung auf Client

In dieser Variante übernimmt der Client die Filteraufgabe. Der Client erstellt zuerst eine Datenbankabfrage und verarbeitet danach das Resultat. Daraus wird ein entsprechender WMS Request erstellt und an den MapServer gesendet. Der Server erstellt die entsprechende Karte als Grafik und liefert diese zurück an den Client.

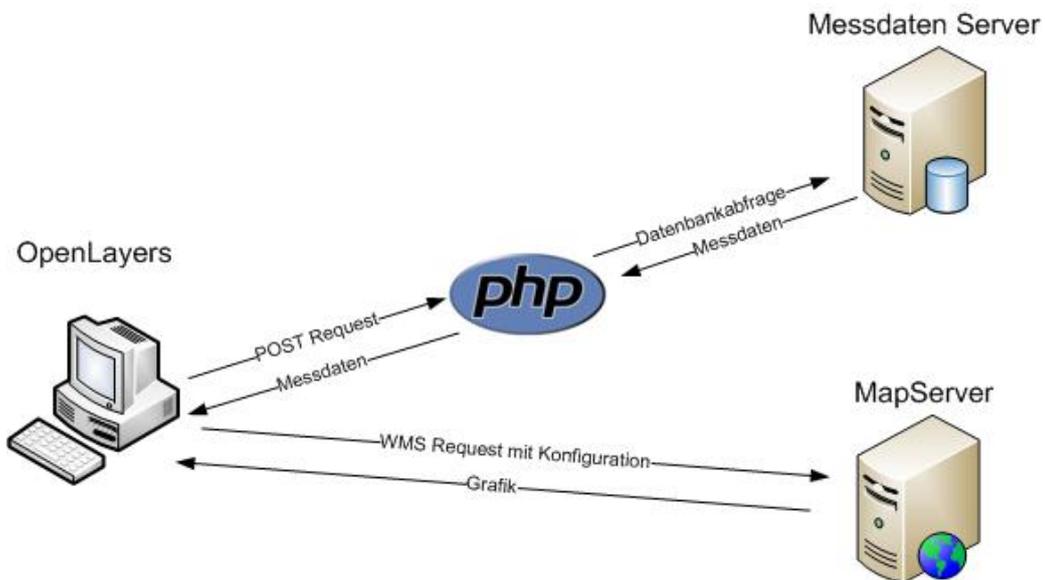


Abbildung 16: Verknüpfung Mess- und Geodaten auf dem Client

Ein Vorteil der Verarbeitung der Daten auf dem Client ist die weitere Nutzung der Messdaten. Diese können auch für die Darstellung neben der Karte in anderen Formaten gebraucht werden. Man kann z.B. Diagramme erstellen und anzeigen, ohne eine weitere Serverabfrage zu tätigen.

9.5 Themenlayer WFS – SLD auf Client

Wie in der vorherigen Variante übernimmt der Client die Verarbeitung der Daten. Der grosse Unterschied liegt an der Anzeige des Themenlayers. Statt einer Grafik werden die Wege und Punkte als Vektordaten zurückgeliefert und vom Client verarbeitet. Die Visualisierung wird erst auf dem Client erstellt. Dadurch entsteht der Vorteil, dass bei Änderungen der Filterung nur noch die Messdaten geladen werden müssen und die Visualisierung schneller und dynamischer angezeigt werden kann. Es ermöglicht auch, Informationen mit einem Mouseover Effekt darzustellen.

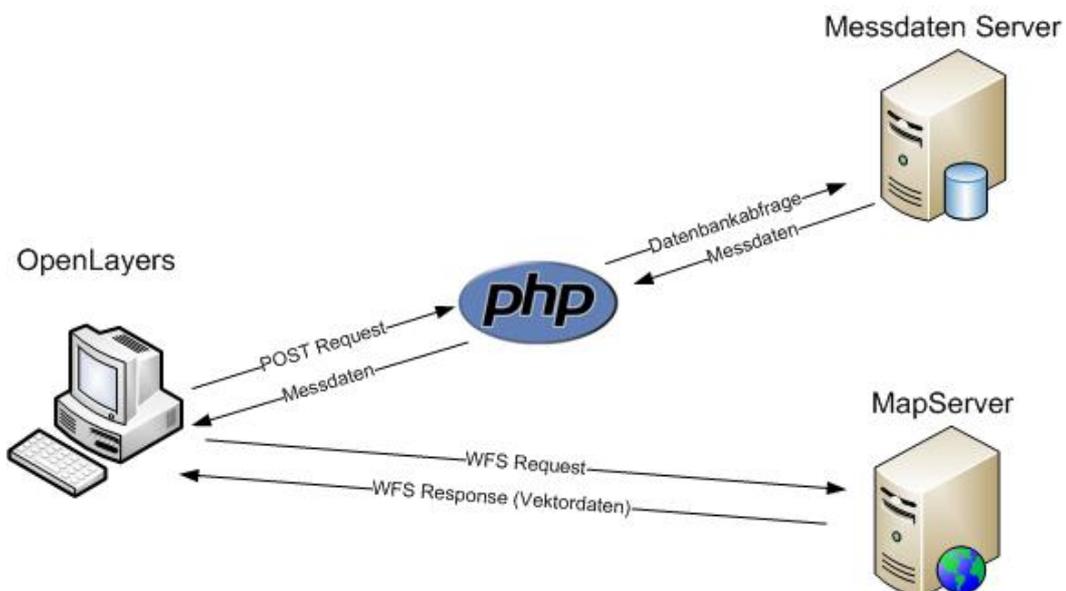


Abbildung 17: Verknüpfung Mess- und Geodaten mit WFS und SLD

Ein Nachteil dieser Variante ist die Beschränkung der Elemente. Da OpenLayers mit Javascript realisiert ist, können maximal 5000 Elemente angezeigt werden. Und nach ca. 1000 Elementen kommt der Client ins Stocken. Diese genaue maximale Anzahl Elemente ist je nach Browser und System unterschiedlich. Bei einem kleinen Park, wie z.B. dem Sihlwald, ist dies kein Problem, da nie so viele Elemente gleichzeitig angezeigt werden müssen.

9.6 Auswertung der Verknüpfungsmethoden

Nach Überprüfung der Varianten hat sich gezeigt, dass es nicht möglich ist, für jedes Element aus der Geodatenbank eine eigene Konfiguration zu erstellen, wie dies in den Varianten 9.2, 9.3, 9.4 und 9.5 gemacht wird. Die Variante 9.1 mit einer Attribut-tabelle und einer View ist daher die einzig funktionierende Lösung.

Shapefiles unterstützen keine dynamischen Änderungen der Attribute, daher müssen diese in eine Geodatenbank wie zum Beispiel PostGIS portiert werden.

Statt die Ebene wie vorgeschlagen mit WMS auf dem Server zu erzeugen und diese als Grafik auf dem Client anzuzeigen, kann die Erstellung der Ebene auch direkt auf dem Client durchgeführt werden. Dafür werden die Geodaten mit WFS aus der View geholt und an den Client gesendet. Dieser erstellt lokal auf dem Client die entsprechenden Vektordaten und zeigt diese auf der Karte dar. Dabei wird im Gegensatz zur Variante 9.1 nur die Erstellung der Karte auf den Client verschoben, der restliche Ablauf bleibt gleich.

10 Installation Server

Alle Installationen wurden auf einem Fedora Linux durchgeführt. Für die Installation auf anderen Systemen, sind die Hinweise und Installationsanleitungen auf den jeweiligen Projektwebseiten zu beachten.

10.1 Installation GDAL

GDAL (Geospatial Data Abstraction Library) ist eine Übersetzungsbibliothek für Geo-Rasterdaten. Integriert in die Bibliothek ist die OGR Bibliothek, welche einfache Übersetzungen von Vektordaten machen kann. Ein Beispiel für eine Übersetzung kann die Konvertierung von einem Koordinatensystem in ein anderes sein. Die aktuellste Version kann auf der Projektwebseite (<http://trac.osgeo.org/gdal/wiki/DownloadSource>) heruntergeladen werden

Installationsschritte

```
tar xzvf gdal-1.7.2.tar.gz
cd gdal-1.7.2
make
make install
```

Damit die Funktionen der Bibliothek genutzt werden können, muss der Runtime Linker (ld) neu konfiguriert werden.

```
cd /etc/ld.so.conf.d
echo /usr/local/lib >local.conf
ldconfig
```

10.2 Installation Proj.4

Proj.4 ist eine Bibliothek für die Projektion von Karten. Die aktuellste Version kann auf der Projektwebseite (<http://trac.osgeo.org/proj/>) heruntergeladen werden.

Installationsschritte

```
tar xzf proj-4.7.0.tar.gz
cd proj-4.7.0/
./configure
make
make install
```

10.3 Installation GEOS

GEOS (Geometry Engine Open Source) ist eine Bibliothek die verschiedene Funktion für Berechnungen mit Geodaten bereitstellt. Die aktuellste Version kann auf der Projektwebseite (<http://trac.osgeo.org/geos/>) heruntergeladen werden.

Installationsschritte

```
tar xjf geos-3.2.2.tar.bz2
cd geos-3.2.2/
./configure
make
make install
```

10.4 Installation PostgreSQL

PostgreSQL, abgekürzt auch Postgres genannt, ist ein objektrelationales Datenbankmanagementsystem, welches von einer Open-Source-Community entwickelt wird. Die aktuelle Version kann auf der Projektwebseite (<http://www.postgresql.org/download/>) heruntergeladen werden.

Installationsschritte

```
./configure
gmake
su
gmake install
```

Nach der Installation muss ein neuer Benutzer „postgres“ und ein Ordner für die Daten erstellt werden.

```
adduser postgres
mkdir /usr/local/pgsql
mkdir /usr/local/pgsql/data
```

Nun kann die Datenbank initialisiert werden.

```
chown postgres /usr/local/pgsql/data
su - postgres
/usr/local/pgsql/bin/initdb -D /usr/local/pgsql/data
```

10.4.1 PostgreSQL starten

PostgreSQL besitzt eine Sicherheitsabfrage, welche es verbietet die Datenbank mit einem root-Benutzer auszuführen. Um die Datenbank zu starten, muss zuerst der Benutzer gewechselt werden.

```
su - postgres
```

Die Datenbank kann nun gestartet werden. Mit dem Parameter `-D` wird das Datenverzeichnis angegeben. Die zu verwendende Logdatei wird mit dem Parameter `-l` konfiguriert.

```
pg_ctl start -D /usr/local/pgsql/data -l /var/log/postgres.log
```

Um die Datenbank beim Systemstart zu starten, kann der Befehl auch in die Datei `/etc/rc.d/rc.local` geschrieben werden.

```
/usr/local/pgsql/bin/pg_ctl start -D /usr/local/pgsql/data -l
/var/log/postgres.log
```

10.5 Installation PostGIS

Um PostGIS 1.5 installieren zu können, muss PostgreSQL in der Version 8.3 oder höher auf dem System installiert sein. Für Rechenoperationen mit Geodaten werden die Bibliotheken Proj4 und GEOS benötigt (die Installation von Proj4 und GEOS wird im Kapitel 10.2 „Installation Proj.4“ respektive Kapitel 10.3 „Installation GEOS“ behandelt).

Die aktuellste Version von PostGIS und eine ausführliche Dokumentation können auf PostGIS Projektseite (<http://postgis.refractory.net/>) heruntergeladen werden.

Installationsschritte

```
tar xvfz postgis-1.5.1.tar.gz
cd postgis-1.5.1
./configure
make
make install
```

10.5.1 PostGIS Datenbank erstellen

Für eine PostGIS Datenbank wird zuerst eine normale Datenbank angelegt. Mit dem Befehl `createlang plpgsql` wird der Datenbank eine prozedurale Sprache beigebracht. Mit den weiteren Befehlen werden die Informationen zu Geometriedaten und zu den verschiedenen Koordinationssystemen in die Datenbank geladen.

```
createdb geodb
createlang plpgsql geodb
psql -d geodb -f postgis.sql
psql -d geodb -f postgis_comments.sql
psql -d geodb -f spatial_ref_sys.sql
```

10.5.2 PostGIS testen

Um die erstellte Datenbank zu testen, kann folgender Befehl ausgeführt werden:

```
ogrinfo PG:"host=127.0.0.1 user=postgres password=postgres dbname=geodb port=5432"
```

Eine mögliche Antwort wäre:

```
INFO: Open of `PG:host=127.0.0.1 user=postgres password=postgres
dbname=geodb port=5432'
      using driver `PostgreSQL' successful.
1: geodb (Multi Line String)
```

10.5.3 Shapes in PostGIS Daten konvertieren und in Datenbank laden

Um die Shapedateien in die Geodatenbank zu laden, ist das Tool *shp2pgsql* nötig. Die Syntax um ein Shapefile zu konvertieren lautet wie folgt:

```
shp2pgsql -s <SRID> shapefile.shp tabelle datenbank > wegenetz.sql
```

In einem konkreten Beispiel sieht der Befehl folgendermassen aus:

```
shp2pgsql -s 900913 wegenetz.shp wegenetz geodb > wegenetz.sql
```

Das Skript *wegenetz.sql* muss noch mit OIDS erweitert werden:

```
CREATE TABLE "wegenetz" (gid serial PRIMARY KEY,  
"objectid" int4,  
...  
"qgeom" varchar(50),  
"shape_leng" numeric)  
WITH (  
        OIDS=TRUE  
);  
...
```

Nun kann das SQL Skript ausgeführt werden

```
psql -d geodb -f wegenetz.sql
```

mit dem Parameter `-s 900913` wird der SRID (Spatial Reference Identifier), der räumliche Referenzbezeichner, angegeben.

10.5.4 Attributtabelle und View erstellen

Um die Attribute der Geoinformationen dynamisch ändern zu können, braucht es eine eigenständige Attributtabelle, die unabhängig von den Geodaten bearbeitet werden kann. Mit folgendem SQL Befehl wird die Attributtabelle erstellt.

```
CREATE TABLE "attr_123" (id serial PRIMARY KEY, gis_pathid int4, size  
int4);
```

Da der MapServer keine SQL-Joins und somit nur den Zugriff auf eine einzelne Tabelle unterstützt, muss eine View erstellt werden. Mit der View lassen sich die Attribut- und Geodatentabellen zu einer automatisch aktualisierten „Tabelle“ verbinden.

```
CREATE VIEW view_123 AS SELECT w.oid, w.*, a.size FROM wege-  
netz_sihlwald w LEFT JOIN attr_123 a ON w.vm_path_gi = a.gis_pathid;
```

Damit die View als Geodatentabelle erkannt wird, muss diese als solche hinzugefügt werden.

```
INSERT INTO geometry_columns VALUES ('', 'public', 'view_123',  
'the_geom', 2, 900913, 'MULTILINESTRING');
```

Das Erstellen der Attributtabelle und View wird beim Aufruf des MapScripts abgearbeitet. Somit kann für jede Anfrage eine eigene Attributtabelle erstellt werden, was sich positiv auf die Performance des Servers und die Antwortzeit für Kartenerstellung auswirkt.

10.6 Einführung in MapServer

MapServer ist, wie oben beschrieben, eine Implementation der OGC Standards. Es ist eine Anbindung an die meisten Geodatenquellen wie z.B. Shapefiles oder PostGIS vorhanden und es werden die meisten OGC Standards als Dienste angeboten.

Die Konfiguration erfolgt über eine Konfigurationsdatei, die im folgenden Dokument MapFile genannt wird. Es handelt sich um eine reine Textdatei, die bei jeder Anfrage an den Server ausgelesen wird.

10.7 Installation von MapServer auf einem Linux Server

Nachdem alle benötigten Tools und Bibliotheken installiert sind kann MapServer installiert werden. Vor der Installation müssen die Pfade zu den Bibliotheken konfiguriert werden.

```
./configure --with-ogr=/usr/local/bin/gdal-config --with-gdal=/usr/local/bin/gdal-config --with-httpd=/usr/sbin/httpd --with-wfsclient --with-wmsclient --enable-debug --with-curl-config=/usr/bin/curl-config --with-proj=/usr/local --with-tiff --with-gd=/usr/local --with-jpeg --with-freetype=/usr/include/freetype2 --with-threads --with-wcs --with-xml2-config=/usr/bin/xml2-config --with-sos --with-libiconv=/usr --with-postgis=/usr/bin/pg_config --with-threads --with-php=/usr/include/php
```

Danach wird die MapServer kompiliert.

```
make
```

Um über den Webserver darauf zugreifen zu können, muss es in das cgi-bin Verzeichnis des Webservers kopiert werden.

```
cp mapserv /var/www/cgi-bin/mapserv
```

Der MapServer kann nicht über `make install` installiert werden, da ein Skript generiert worden ist.

10.8 Konfiguration mit dem MapFile

Das Herzstück von MapServer ist das MapFile. Darin befindet sich die Konfiguration für das Erstellen einer Karte. Es können mehrere Konfigurationen auf dem gleichen Server laufen gelassen werden, da bei jedem Aufruf das MapFile angegeben werden muss.

Beispiel für eine Konfiguration:

```
MAP
  NAME "sample"
  STATUS ON
  SIZE 600 400
  SYMBOLSET "../etc/symbols.txt"
  EXTENT -180 -90 180 90
  UNITS DD
  SHAPEPATH "../data"
  IMAGECOLOR 255 255 255
  FONTSET "../etc/fonts.txt"

  #
  # Start of web interface definition
  #
  WEB
    IMAGEPATH "/ms4w/tmp/ms_tmp/"
    IMAGEURL "/ms_tmp/"
  END

  #
  # Start of layer definitions
  #
  LAYER
    NAME 'global-raster'
    TYPE RASTER
    STATUS DEFAULT
    DATA bluemarble.gif
  END
END
```

10.8.1 Erstellen mit dem Plugin „MapServer Export“ von Quantum GIS

Quantum GIS stellt ein Plugin zur Verfügung, das aus dem aktuellen Projekt eine MapServer Konfiguration erstellt.

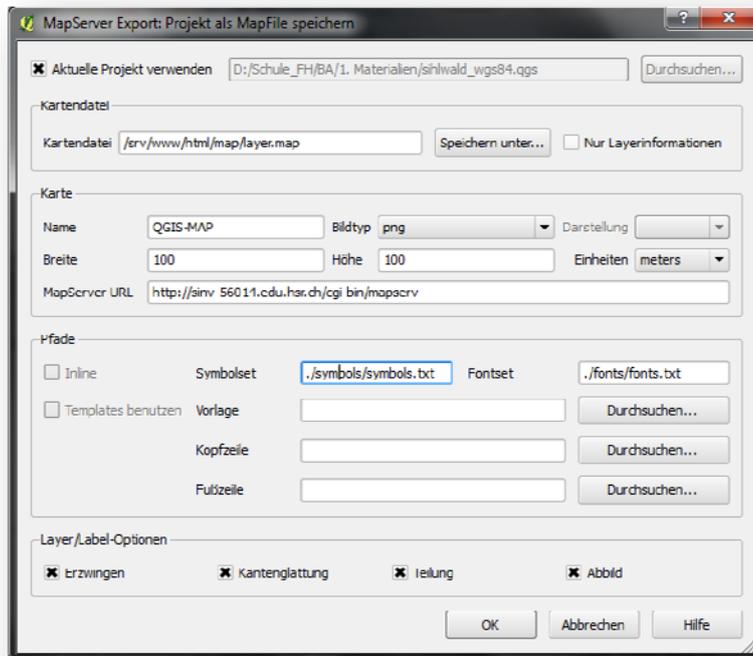


Abbildung 18: MapFile Export mit Quantum GIS

Nach dem Export müssen die Pfade noch mit einem normalen Editor dem Server angepasst werden.

10.9 Konvertieren von Shapefiles

Die Shapefiles können aus verschiedenen Koordinatensystemen stammen. Die großen frei erhältlichen Kartendaten von Google Maps und OpenStreetMap verwenden ein System Namens Google Maps Global Mercator. In der Schweiz wird ein eigenes Format CH9013 oder neu CH9013+ verwendet. Um verschiedene Layer übereinander anzuzeigen, müssen alle aus dem gleichen Koordinatensystem stammen. Das einfachste ist es, Shapefiles in WGS84 zu konvertieren. Dazu gibt es verschiedene Möglichkeiten.

In der Dokumentation zur aktuellsten Version von MapServer wird zwar ein Weg beschrieben, wie die Konvertierung on-the-fly durchgeführt wird. Diese funktioniert jedoch nicht und wird auch in der Online-Community als nicht brauchbar dargestellt.

10.9.1 Konvertieren mit Geoconverter

Das Institut für Software / GIS von der HSR stellt im Internet eine eigene Applikation zur Verfügung, die Shapefiles von einem beliebigen Koordinatensystem in ein Anderes umwandelt. Dabei kann auch das Ausgabeformat angegeben werden. Zu finden ist dieser Konverter online unter <http://geoconverter.hsr.ch/>. Um ihn zu benutzen, ist eine kurze Registration im Wiki notwendig.

Die vier Geodaten-Dateien mit den Endungen .dbf, .prj, .shp und .shx werden dabei in ein Zip-Archiv gepackt und mit dem Geoconverter hinaufgeladen.

Abbildung 19: Shapefiles Online mit GeoConverter konvertieren

Im 2. Schritt wird das Ausgabeformat (im Bild Shapefile) gewählt und das Quell- und Ziel-Koordinaten System eingetragen.

Nach dem Klick auf „Convert“ können die konvertierten Shapedateien wieder heruntergeladen werden.

10.9.2 Mit Kommandozeilentool ogr2ogr

Das Kommandozeilentool ist über verschiedene Pakete erhältlich. Das umfassendste Paket ist FWTools.

Am einfachsten funktioniert die Konvertierung mit der FWTools Shell, die nach der Installation im Startmenü von Windows zu finden ist. In der Shell kann mit dem Befehl „cd“ in das Verzeichnis mit den Shapefiles gewechselt werden.

Um ein Shapefile zu konvertieren, wird folgender Befehl eingegeben:

```
>ogr2ogr -s_srs EPSG:2056 -t_srs EPSG:900913 gmaps zaehlstellen_manuell.shp
```

Die Parameter `s_srs` und `t_srs` stehen für das Quell- und Ziel-Koordinatensystem.

Die entsprechenden EPSG-Nummern können in der Spatial Reference (<http://www.spatialreference.org/>) nachgelesen werden. EPSG:2056 steht hierbei für das neue Schweizer Koordinatenformat CH1903+/LV95 und EPSG:4326 ist das weit verbreitete GPS-Format WGS 84. Da Google Maps eine eigene Implementation verwendet, ist die Konvertierung nicht mit allen Tools möglich. Die dafür gebrauchte EPSG Nummer lautet 900913.

Mit dem nächsten Parameter (im Beispiel „gmaps“) kann ein Zielordner oder eine Zieldatei angegeben werden. Wird kein neuer Dateinamen verwendet, wird die neue Datei den gleichen Namen haben. Existiert bereits eine Datei mit dem gewünschten Namen, wird die Konvertierung abgebrochen. Mit dem Parameter `-overwrite` kann die vorhandene Datei ersetzt werden.

11 Benutzeroberfläche

Für die Benutzeroberfläche wird das JavaScript Framework ExtJS verwendet. ExtJS bietet Unterstützung bei Ajax-Funktionen und Manipulation von HTML Elementen. Zudem besitzt ExtJS eine riesige Bibliothek mit Elementen für die Benutzeroberfläche. Es ist ein Hilfsmittel zur Erstellung von interaktiven Webapplikationen.

11.1 Bedienelemente

Bereich	Funktionen	UI - Elemente	Benutzung
Menu	Drucken	Button	wenig
	Einstellungen	Fenster	wenig
	Basislayer	DropDown	mässig
	Fenster zentrieren	Button	oft
Visualisierung	Karte	MapPanel	sehr oft
	Verschieben und Zoomen	Maus- und Tastaturaktionen	sehr oft
	Diagramm	Panel	oft
Anzeigelayer	Standortfilterung	Checkboxen	mässig
Filteroptionen	Zeitbereich einstellen	DatePicker	oft
	Zeitabschnitt	Checkboxen	oft
	Attribute	Checkboxen	oft
Anzeigeoptionen	Visualisierungsart	RadioBoxen	oft
	Zahl darstellen	Checkbox	oft
	Frequenzvisualisierung	RadioBoxen	mässig
	Wegbreite anpassen	Slider	mässig
	Farbe anpassen	Slider	mässig
	Farbauswahl	ColorPicker	oft
Detailansicht	Messdaten und Beschreibungen des Standorts anzeigen	Tabelle	mässig

Legende:

- ColorPicker: Farbauswahl
- DropDown: Auswahlbox
- DatePicker: Datumauswahl
- Panel: Fenster
- Slider: Schieberegler

11.1.1 Visualisierung

Die Visualisierung beinhaltet entweder eine Karte oder ein Diagramm. Es soll möglich sein, mehrere Visualisierungen zu erstellen, die gleichzeitig angezeigt werden. Jede Visualisierung hat eigene Filter- und Anzeigeebenen.

Die Visualisierung hat erste Priorität und soll immer zu sehen sein.

11.1.2 Anzeigeebenen

Die Karte enthält neben der Visualisierung des Besucherstroms auch weitere Ebenen z.B. mit den Zählstellen oder POI. Diese sollten für jede Karte individuell ein- und ausgeblendet werden können.

11.1.3 Filteroptionen

In den Filteroptionen wird die Visualisierung konfiguriert. Jede Visualisierung hat eigene Optionen. Da es mehrere Visualisierungen geben kann und die Filteroptionen zusammen mit den Anzeigeebenen viel Platz verbrauchen, werden nur die Optionen zur aktiven Visualisierung angezeigt.

Die Filteroptionen stehen im direkten Zusammenhang zu den Anzeigeebenen und sollten daher auch nebeneinander positioniert werden.

Wie die Anzeigeebenen, werden auch die Optionen am Anfang definiert und sollten aus Platzgründen ausgeblendet werden können.

11.1.4 Anzeigeeinstellungen

In den Anzeigeebenen wird definiert, wie eine gefilterte Visualisierung aussehen soll. Man kann ein Symbol wählen, welches für die Anzeige verwendet wird. Die Frequenz kann mit der Breite des gewählten Symbols visualisiert werden oder mit einem Farbspektrum. Wenn kein Farbspektrum verwendet wird, muss trotzdem eine Farbe für die Symbole gewählt werden können.

Die Anzeigeebenen werden im Gegensatz zu den Filteroptionen nicht so oft gewechselt. Sie sollten auch ausgeblendet werden können, damit genügend Platz für die Filteroptionen bleibt.

11.1.5 Detailansicht

In der Detailansicht werden die Messzahlen und Beschreibungen zum aktiven Standort angezeigt. Um Daten in der Detailansicht anzuzeigen, muss entweder der Standort in der Visualisierung aktiviert oder in einer DropDown Liste ausgewählt werden.

Da die Details sich auf die Filterung beziehen, werden diese nur von der aktuellen Visualisierung angezeigt.

Die Detailansicht im Vordergrund erscheint nur, wenn die Ansicht auch wirklich benötigt wird. Das bedeutet, man muss die Möglichkeit haben, die Detailansicht ein- und ausblenden zu können.

11.2 Layout

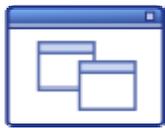
11.2.1 BorderLayout



Ein BorderLayout erlaubt es, verschiedene Layouts geordnet anzuzeigen. Es gibt einen Hauptbereich, der immer angezeigt wird. Die weiteren Bereiche, die um den Hauptbereich herum positioniert sind, können nach Belieben auf- und zugeklappt werden.

Das BorderLayout hat den Vorteil, alles an seinem Platz zu haben. Es entsteht ein einheitliches Layout, das durch auf- und zuklappen trotzdem angepasst werden kann. Dadurch wird kein Platz verschwendet und es kommen keine Fenster der Karte in den Weg.

11.2.2 Frei verschiebbare Fenster



Im Layout mit verschiebbaren Fenstern wird jedes Bedienelement in einem eigenen Fenster angezeigt. Die Fenster können individuell vergrößert, verkleinert und positioniert werden.

Vorteil dieses Layouts ist, dass die Oberfläche frei angepasst werden kann. Jeder Benutzer kann seine eigenen Fenster anzeigen und positionieren. Viele grosse Softwareapplikationen arbeiten mit diesem Layout, um dem Benutzer die Freiheit zu geben, nur die Elemente anzuzeigen, welche auch gebraucht werden.

Da in der Visualisierung der Besucherströme nicht so viele Elemente benötigt werden, ist dieses Layout nicht sehr praktisch, denn die Fenster können überlappen und es erfordert viel Zeit, diese immer wieder richtig zu positionieren.

11.2.3 Auswertung Layout

Im Prototyp wird ein BorderLayout verwendet. Im Gegensatz zum Fensterlayout hat jedes Element einen fixen Platz und die Webapplikation bleibt übersichtlich. Trotzdem ist das BorderLayout flexibel, da die Bereiche ein- und ausgeblendet werden können.

11.3 Entwurf einer Benutzeroberfläche

Anhand der Bedienelemente und deren Priorisierungen wird eine Benutzeroberfläche im BorderLayout entworfen. Die Pfeile bei den Balken zeigen, dass dieses Element ein- und ausgeklappt werden kann.

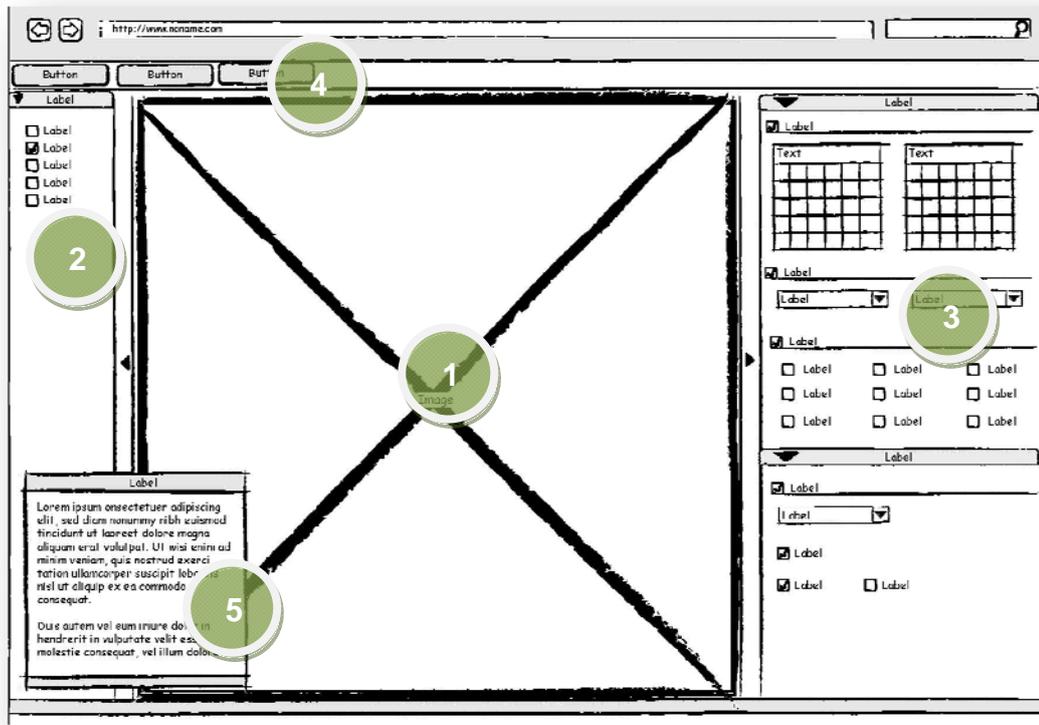


Abbildung 20: Entwurf der Benutzeroberfläche

Legende:

- (1) Der grösste Bereich ist für die Darstellung der Karte reserviert
- (2) Auflistung der Ebenen, die auf der Karte ein- und ausgeblendet werden können
- (3) Karteneinstellungen mit den Elementen Filteroptionen und Anzeigeeinstellungen
- (4) Menu den Basisfunktionen für die Webapplikation
- (5) Detailinformationen in einem frei verschiebbaren Fenster

11.4 Realisierte Benutzeroberfläche

Die Benutzeroberfläche wird mit Hilfe des Entwurfs aus dem Kapitel 11.3 implementiert.

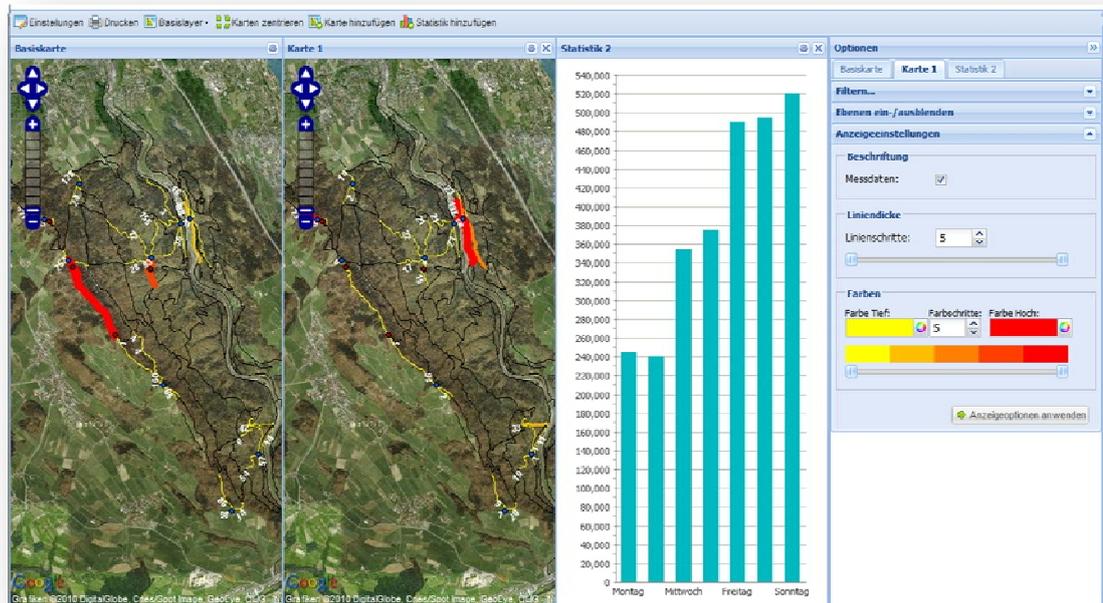


Abbildung 21: Implementierte Benutzeroberfläche

11.4.1 Unterschiede zum Entwurf

Tabs in Karteneinstellungen

Da ExtJS keine aktiven Fenster kennt, muss eine andere Möglichkeit gesucht werden, Einstellungen für alle Karten und Diagramme darzustellen. Mit Tabs und einem Button im Kopfteil der Karte, der direkt die entsprechenden Einstellungen anzeigt, ist eine Lösung entstanden, die übersichtlich ist und alle Anforderungen erfüllt.

Verschiebung der Ebenenauswahl

Der Bereich für die Ebenenauswahl wird in die Karteneinstellungen auf der rechten Seite verschoben. Mit der Verschiebung wird Platz gespart, da nur noch ein Fenster mit Einstellungen geöffnet werden muss. Es ist es nun möglich, individuell für jede Karte Ebenen ein- und auszublenden.

12 Implementation

12.1 Webapplikation

12.1.1 Basislayer darstellen

Im Prototyp kann über das Menu „Basislayer“ zwischen vier verschiedenen Karten gewählt werden:

- Google Hybrid
- Google Streets
- Google Satellite
- Sihlwald

Um die drei Googlekarten darstellen zu können, wird das JavaScript von Google Maps benötigt. Mit einem kostenlosen Google Konto kann ein Schlüssel angefordert werden, welcher für die erfolgreiche Einbindung des JavaScript benötigt wird.

Das Einbinden von Karten wie z.B. von Google wird von OpenLayers unterstützt und in den Beispielen der Projektwebseite gut beschrieben.

Auch eigene Daten von einem WMS Server können als Basislayer eingebunden werden. Dies wird durch die Auswahl „Sihlwald“ gezeigt.

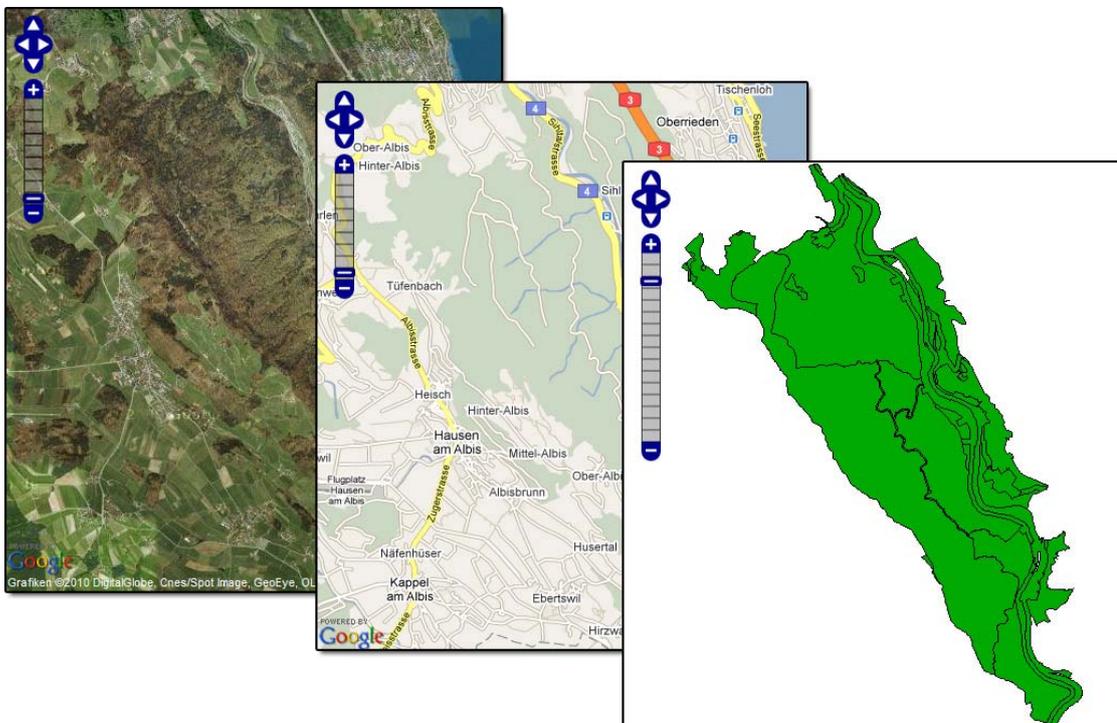


Abbildung 22: Verschiedene Basislayer zum Auswählen

12.1.2 Themenlayer darstellen

Die Themenlayer werden über dem Basislayer angezeigt und können im Einstellungstab ein- und ausgeschaltet werden. Das Wegenetz wird als unterste Ebene schwarz angezeigt. Darüber werden in blau und rot die manuellen und automatischen Zählstellen als Punkte dargestellt. Diese drei Themenlayer sind fix konfiguriert und können nur auf dem MapServer verändert werden.



Abbildung 23: Bedienelement - Ebenen ein-/ausblenden

Der vierte Themenlayer, die visualisierten Wege, wird je nach Filteroptionen und Anzeigeeinstellungen angezeigt.

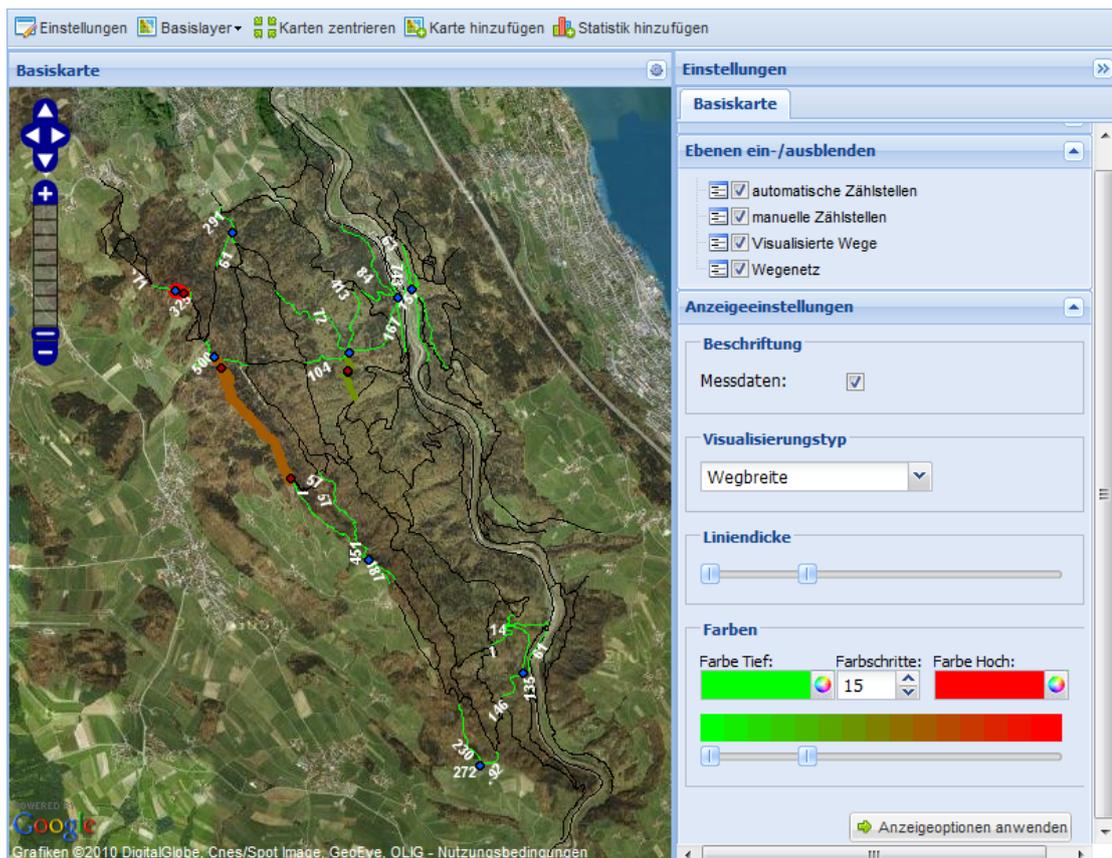


Abbildung 24: Visualisierung der Themenlayer auf einem Basislayer

Eine Standortauswahl oder Nahansicht wird nicht implementiert. OpenLayers stellt eine Zoom-Funktion bereit, die den gleichen Zweck erfüllt. Somit wird auch das Hinein- und Herauszoomen mit der Tastatur (Tasten +/-) oder mit der Maus (Scrollrad und Doppelklick) unterstützt.

Zusammenspiel zwischen Server und Client bei der Kartenerstellung

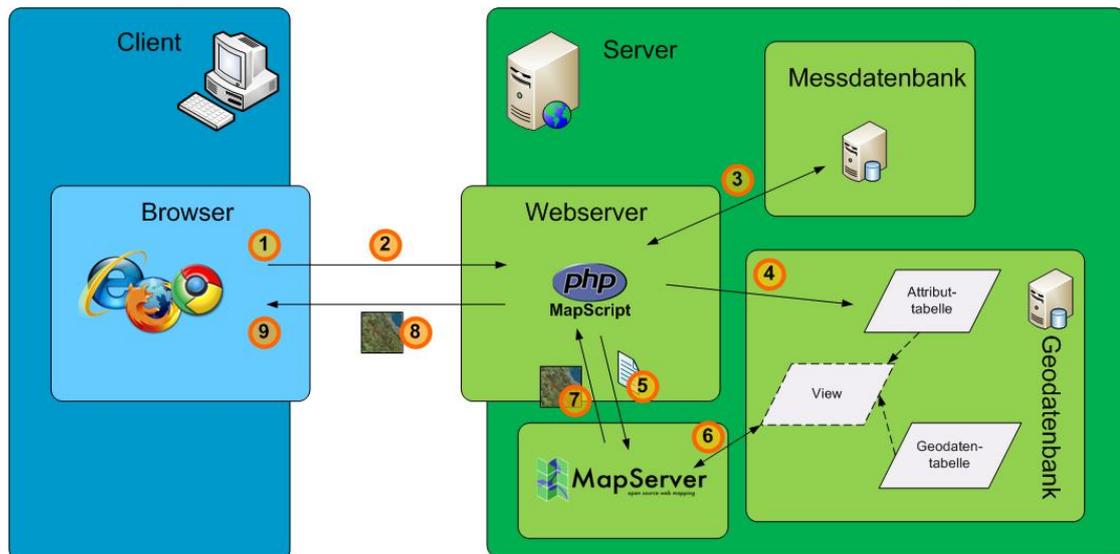


Abbildung 25: Zusammenspiel zwischen Server und Client bei der Kartenerstellung

Ablauf:

- (1) Der Benutzer wählt Filteroptionen und Anzeigeeinstellungen aus.
- (2) Die gewählten Parameter werden an den Webserver übermittelt
- (3) Auf dem Webserver werden die Parameter von einem PHP Skript empfangen und ausgewertet. Anhand dieser Parameter wird eine Anfrage an die Messdatenbank gesendet, welche die entsprechenden Messdaten, die Besucherzahlen, zurückliefert.
- (4) Das PHP Skript baut eine Verbindung zum Geodatenbankserver auf und speichert die Messdaten in die Attributtabelle.
- (5) Wieder anhand der Parameter erstellt das PHP Skript ein SLD. Das SLD baut auf einer Vorlage auf und wird im XML Format gespeichert. In die Vorlage wird für jeden Farbschritt ein Filter erstellt, der eine entsprechende Wegbreite und Farbe zugewiesen bekommt. Dieses SLD wird zusammen mit den GIS Parameter an den MapServer geschickt.
- (6) Der MapServer holt die Geodaten und die dazugehörigen Attribute aus der View, die sich aus der Attributtabelle und der Geodaten-tabelle zusammensetzt.
- (7) Aus den Geodaten, den Attributen und dem SLD generiert der MapServer eine Grafik, welche die Wege visualisiert.
- (8) Die Grafik wird vom MapScript wieder zurück an den Browser geschickt.
- (9) Die im Browser laufende Webapplikation stellt diese Grafik über dem Basis-layer dar.

Für die Erstellung der Karte wird der Dienst WMS benutzt.

12.1.3 Besucherfrequenz mit Wegbreite und Farben darstellen

In den Anzeigeeinstellungen im Einstellungstab kann die Visualisierung individuell angepasst werden.

Über die „**Beschriftung**“ kann ausgewählt werden, ob die berechneten Besucherzahlen angezeigt werden sollen. Der MapServer achtet bei der Erstellung der Karte darauf, dass keine Labels, dazu gehören auch die Besucherzahlen, übereinander liegen. Wenn ein Konflikt entsteht, entscheidet der MapServer selbstständig, welcher Label ausgeblendet wird. Da die Wege sehr nahe beieinander stehen, kann es vorkommen, dass die nicht alle Besucherzahlen angezeigt werden. Ein Hineinzoomen in die Karte löst dieses Problem.

Der „**Visualisierungstyp**“ entscheidet, ob die Besucherzahlen mit Wegbreite und Farben visualisiert werden oder mit Pfeilen. Mehr zu der Visualisierung mit Pfeilen im Kapitel 12.1.4 „Besucherstrom mit Pfeilen darstellen“.

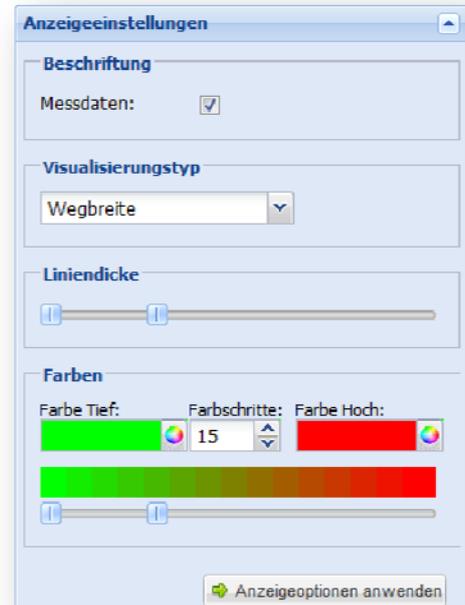


Abbildung 26: Bedienelement Anzeigeeinstellungen

Das Bedienelement „**Liniendicke**“ dient dazu, die Minimum- und Maximumgrenze für die Breite anzupassen. Zum Beispiel kann diese Anzeigeeinstellung verwendet werden, um Spitzenwerte aus Zählungen herauszufiltern. Ohne diese Filterung kann es vorkommen, dass nur ein Weg breit und in der Maximumfarbe dargestellt wird und alle anderen Wege in der Minimumeinstellung.

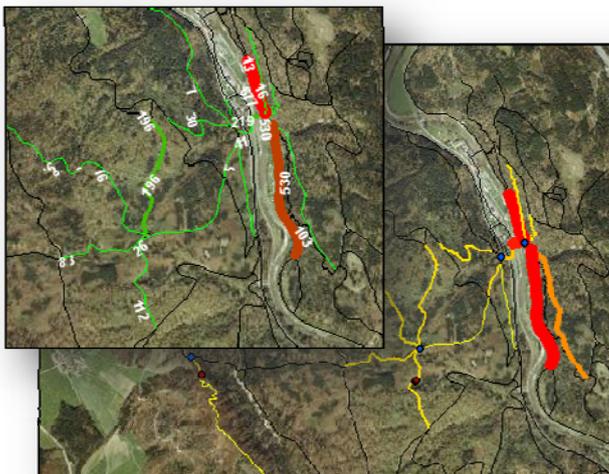


Abbildung 27: Visualisierung mit Wegbreite und Farben

Im Bedienfeld „**Farben**“ können die Minimal- und Maximalwerte für die Visualisierung ausgewählt werden. Der Farbverlauf wird dabei automatisch neu berechnet.

Mit den Farbschritten wird die Anzahl der Farben eingestellt. Je höher der Wert, desto genauer ist der Unterschied ersichtlich.

Wie auch bei der Liniendicke, gibt es auch für die Farben ein Schieberegler für die Grenzen.

Die Einstellungen werden erst nach einem Klick auf „Anzeigeoptionen anwenden“ auf der Karte ersichtlich. Eine sofortige Anzeige der Änderungen erzeugt zu viele Anfragen an den Server und bringt damit jeden Browser zum Absturz.

12.1.4 Besucherstrom mit Pfeilen darstellen

Wird als Visualisierungstyp „**Pfeile**“ ausgewählt, werden die Besucherzahlen mit verschieden grossen und andersfarbigen Pfeilen angezeigt. Leider gibt es in MapServer kein Geometrietyp Pfeil. Und wenn es einen gäbe, wäre es schwierig, die Richtung, in die der Pfeil zeigen soll, anzuzeigen. Was der MapServer jedoch beherrscht, ist das Folgen der Linie bei Labels. Das bedeutet, Beschriftungen werden immer entlang der Linie angezeigt.

Das Follow-Feature kann natürlich auch für die Darstellung von Pfeilen verwendet werden, sofern der Pfeil eine Beschriftung ist. Da der MapServer verschiedene Zeichensätze kennt, ist dies kein Problem, denn z.B. unter dem ASCII Code hex2192 versteckt sich ein Pfeil nach rechts (→).

MapServer unterstützt keine statischen Labels, jedoch kann in Labels statischer Text mit dem Property vermischt werden. Mit folgender kleinen Umgehung kann der MapServer auch nur statischen Text anzeigen.

```
<ogc:Label>&#x2192;<ogc:PropertyName></ogc:PropertyName></ogc:Label>
```

Das Problem mit dem statischen Text ist bei MapServer schon seit einiger Zeit bekannt, jedoch noch nicht behoben worden.

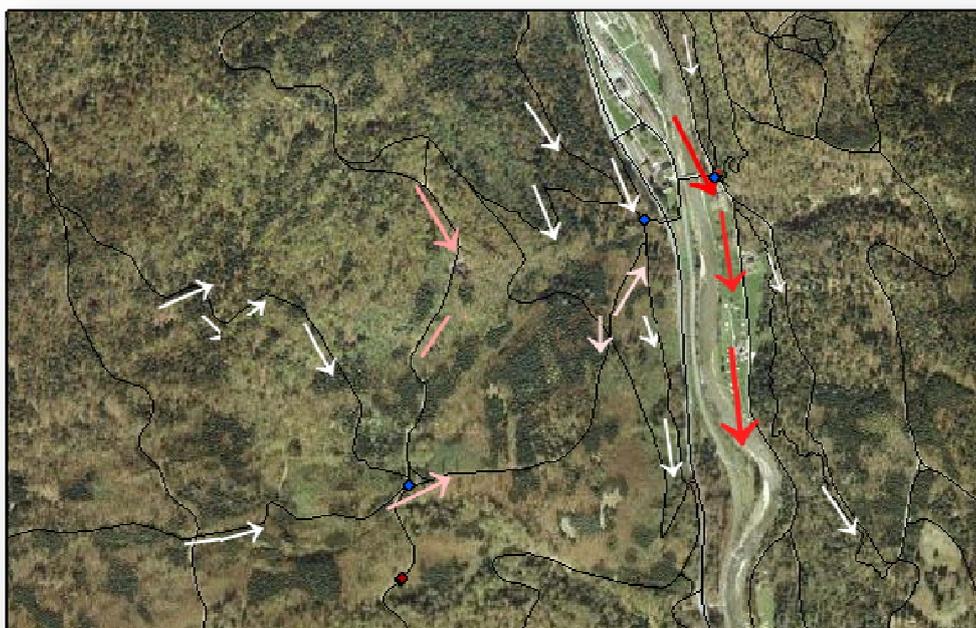


Abbildung 28: Visualisierung mit Pfeilen

Damit nicht die einen Pfeile ganz klein und andere Pfeile ganz gross dargestellt werden, müssen die minimale und die maximale Schriftgrösse nahe bei einander liegen. Im Prototyp wurde dies erreicht, indem zur eigentlichen Schriftgrösse zuerst 2-mal die maximale Schriftgrösse addiert wird, und das ganze danach durch 3 geteilt wird.

Beispiel:

Schriftgrösse: 3

Maximale Schriftgrösse: 15

Berechnung neue Schriftgrösse: $(3+15+15)/3 = \mathbf{11}$

12.1.5 Diagramm

Das Erstellen von Diagrammen in ExtJS ist kein Problem. Einer Diagrammkomponente kann lediglich ein Objekt mitgegeben werden, welches die aktuellen Zahlen, sowie deren Beschriftungen enthält und das Diagramm wird sofort aktualisiert.

Um ein Diagramm anzuzeigen, werden die Filteroptionen mit einem Ajax-Request an das PHP Skript „*getData.php*“ übergeben. Das Skript holt aus dem Messdatenbankserver die entsprechenden Daten und gibt diese gleich richtig formatiert als JSON Objekt zurück. Das Objekt kann ohne Änderungen in das Diagramm geladen werden.

Die Filterung im PHP Skript funktioniert gleich wie bei Kartenerstellung, daher wird nicht weiter darauf eingegangen.

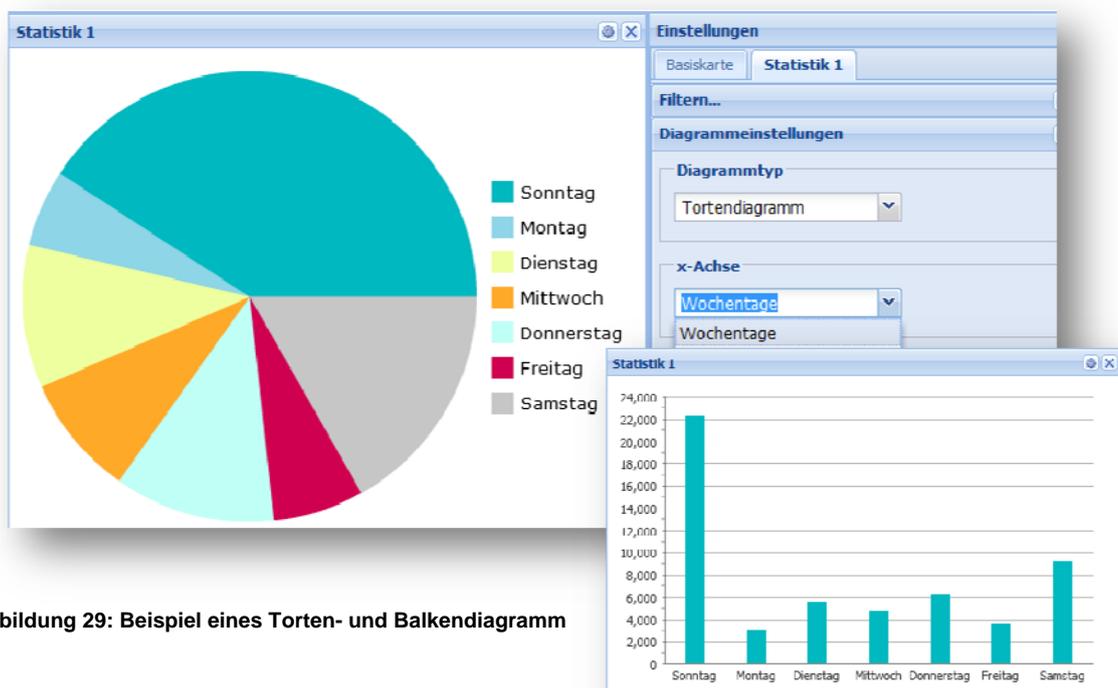


Abbildung 29: Beispiel eines Torten- und Balkendiagramm

12.1.6 Diagrammeinstellungen

Der Prototyp beschränkt sich auf die Diagrammtypen Balkendiagramm, Liniendiagramm und Tortendiagramm. Weitere Typen werden von ExtJS zur Verfügung gestellt und könne problemlos eingebaut werden.

Für die x-Achse kann zwischen den Wochentagen, den Monaten und den Attributen ausgewählt werden. Die Auswahl der gewünschten Wochentage, Monate oder Attribute kann über die Filteroptionen beschränkt werden.

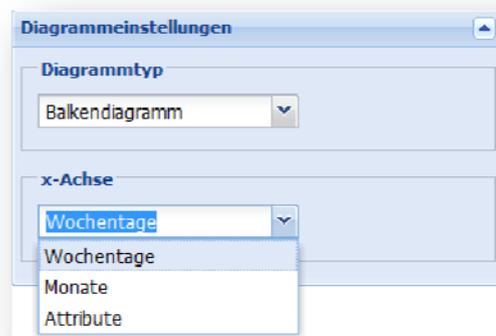


Abbildung 30: Bedienelement Diagrammeinstellungen

12.1.7 Filteroptionen

Die Filteroptionen sind für Karten wie auch für Diagramme gleich. Implementiert wurden die Filter, die im Lösungskonzept ausgearbeitet wurden.

Der Zeitabschnitt wird ohne die Auswahl der Jahre implementiert, da für den Prototyp nur Messdaten aus dem Jahr 2009 zur Verfügung stehen.

Bei der Bedienung muss beachtet werden, dass wenn bei den Filteroptionen eine Filtergruppe aktiviert ist



Abbildung 32: Filterung nach Wochentagen

und keine Elemente ausgewählt werden, der Filter wirkungslos ist und vom Server nicht verarbeitet wird.

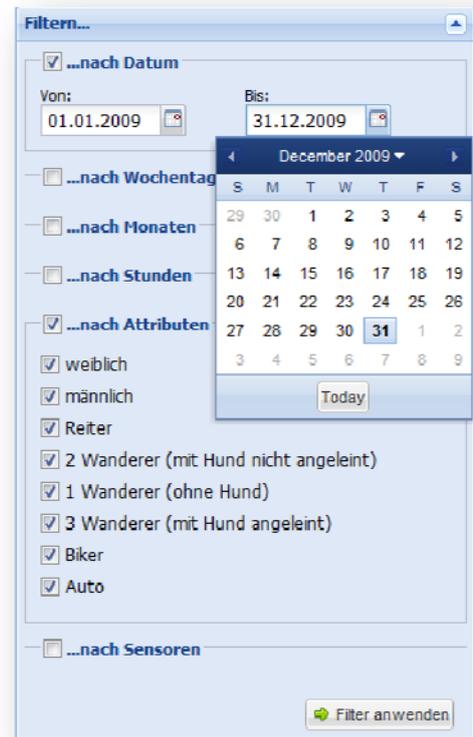


Abbildung 31: Bedienelement Filter

12.1.8 Ladezeiten optimieren

Um möglichst gute Ladezeiten zu erreichen, gibt es verschiedene Massnahmen. Am effektivsten ist es, möglichst wenige Anfragen an den Server zu schicken, da die Anzahl gleichzeitiger Serveranfragen vom Browser beschränkt ist. Bei aktuellen Browsern (InternetExplorer 8 und FireFox 3) sind in der Standardeinstellung maximal 6 Anfragen an eine Domain erlaubt. Ältere Browser erlauben meist nur 2 gleichzeitige Anfragen.

Um möglichst wenige Anfragen zu generieren, ist es von Vorteil, die Javascripts in möglichst wenige Dateien zusammenzufügen. Das gleiche kann auch mit den CSS gemacht werden. Nachteile entstehen dadurch keine, einzig die Übersicht bei der Fehlersuche leidet ein bisschen darunter.

Die JavaScript Dateien können mit dem YUI Compressor komprimiert werden, dies beschleunigt das Laden zusätzlich, weil weniger Daten übertragen werden müssen.

Im Prototyp wurden alle Plugins, die für verschiedene ExtJS Elemente benötigt werden, in einer Datei zusammengefügt und komprimiert. Dieses Skript wird nach diesen Massnahmen im Durchschnitt dreimal schneller geladen.

12.2 Server

12.2.1 Konfiguration MapFile

Die im Prototyp verwendeten Ebenen werden in der Konfigurationsdatei „wms.map“ beschrieben. Das MapFile wurde mit QuantumGIS automatisch erstellt und danach angepasst.

Im ersten Bereich werden die Defaultwerte und Allgemeinen Einstellungen gemacht.

```
MAP
  NAME "Sihlwald"
  # Map image size
  SIZE 600 600
  UNITS meters

  EXTENT 942459.838350 5977415.995503 961706.930573 5989013.602355
  FONTSET './fonts/fonts.txt'
  SYMBOLSET './symbols/symbols.txt'
  PROJECTION
    'proj=longlat'
    'ellps=WGS84'
    'datum=WGS84'
    'no_defs'
  END
  ...
```

Die Parameter `SIZE` und `UNITS` werden zwar benötigt, aber durch die OpenLayers Parameter wieder überschrieben.

`EXTENT` beschreibt die Erstreckung und Grösse der Karte. Dieser Parameter wurde durch Quantum GIS gesetzt und könnte auch aus dem Shapefile gelesen werden

`FONTSET` definiert welche Schriftarten gebraucht werden können und wo diese auf dem Server zu finden sind. Es können alle TrueType Schriftarten verwendet werden. Wenn der Server diese nicht selber mitbringt, können diese ganz einfach von einem Windows-Rechner aus dem Ordner „C:Windows\Fonts“ auf den Server kopiert werden. Für die Definition in der Datei „`fonts.txt`“ wird folgendes Format verwendet.

```
name /pfad/zur/schriftart/schriftart.ttf
```

Das kann zum Beispiel folgendermassen aussehen.

```
arial /usr/share/fonts/truetype/arial.ttf
```

Mit `SYMBOLSET` können selber definierte Symbole angegeben werden. Diese werden später im Bereich `STYLE` referenziert. Im Prototyp wird das Beispielsymbolset aus der MapServer-Installation verwendet.

Der Bereich `PROJECTION` beschreibt die Defaultwerte für die Projektion, welches Koordinatensystem verwendet werden soll.

Die Ebenenkonfiguration

Für jede Ebene wird ein eigener Konfigurationsabschnitt erstellt. Über den Parameter `NAME` kann die Ebene von einem Client einzeln abgerufen werden

```
LAYER
  NAME "wege"
  STATUS ON
  TYPE LINE
  CONNECTIONTYPE POSTGIS
  CONNECTION "host=127.0.0.1 port=5432 dbname=db user=user pass-
word=password"
  DATA "the_geom from wegenetz_view"
  CLASS
    NAME 'wege'
    STYLE
      SYMBOL 'myline'
      SIZE 1.0
      OUTLINECOLOR 38 43 253
      COLOR 255 0 0
    END
  END
END
```

Mit `STATUS` wird eingestellt, ob die Ebene bei einem Aufruf ohne Layer-Parameter angezeigt werden soll.

Die beiden `CONNECTION/TYPE` Parameter werden nur für den Einsatz eines Geodatenbankservers gebraucht. Die Syntax ist je nach verwendeter Datenbank anders. Im Prototyp wird PostGIS verwendet

Der wichtigste Parameter ist `DATA`. Hier wird definiert, wo die Geodaten hergeholt werden. Als Wert kann entweder der vollständige Pfad zum Shapefile

```
DATA '/var/www/html/map/shapes/wegenetz.shp|layerid=0'
```

oder wie im Beispiel oben die Tabelle mit der entsprechenden Spalte im Format

```
DATA "<column> from <table>"
```

angegeben werden.

Im Bereich `CLASS` kann die Darstellung angepasst werden. Diese wird jedoch im Prototyp nicht benötigt, da der Style mit einem SLD angepasst wird.

12.2.2 Attribute und Sensoren auslesen

Die Besucherzahlen sollen unter anderem nach Attributen und Sensoren gefiltert werden. Diese können über ein weiteres Tool hinzugefügt oder verändert werden. Damit in der Webapplikation die Attribute und Sensoren immer aktuell sind, werden diese direkt aus der Messdatenbank gelesen. Das Auslesen der Daten wird von dem PHP Skript „getAttributes.php“ erledigt.

Das Skript wird über ein Ajax-Request aus der Webapplikation aufgerufen und holt die entsprechenden Daten aus der Messdatenbank. Die Daten werden in einem Array gespeichert, in ein JSON Objekt umgewandelt und an die Webapplikation zurückgesendet.

```
echo "{attributs:".json_encode(to_utf8($attributes)).", sensors:
.json_encode(to_utf8($sensoren))."}";
```

12.2.3 Karte erstellen mit MapScript

Für das Erstellen der Karte ist das PHP Skript „getMap.php“ zuständig, welches MapScript implementiert.

Einen Überblick, wie der Vorgang abläuft, wird in der Abbildung 25 „Zusammenspiel zwischen Server und Client bei der Kartenerstellung“ erläutert.

GIS Parameter in Kartendarstellung integrieren

Die MapScript Funktion „*loadparams*“ funktioniert in einer reinen PHP Umgebung nicht, daher müssen die GIS Parameter manuell entgegengenommen und weitergeleitet werden. OpenLayers stellt die Karte im Koordinatensystem 900913 dar. Damit der MapServer mit den Geodaten arbeiten kann, wird das Koordinatensystem auf WGS84 (EPSG:4326) umgestellt.

```
$params = $_REQUEST;  
$params['SRS'] = "EPSG:4326";  
foreach ($params as $k => $v) {  
    $request->setParameter($k, $v);  
}
```

Filter übernehmen und in SQL-WHERE umwandeln

Die Filterparameter, welche von der Webapplikation übergeben werden, müssen überprüft und richtig formatiert werden. Für den Fall, dass ein Parameter nicht übergeben wird, muss ein Defaultwert eingestellt werden.

```
$fromDate = "2009-01-01";  
if(isset($params['FROMDATE'])) {  
    $fromDate = date('Y-m-d', strtotime($params['FROMDATE']));  
}
```

Im Beispiel wird der Filter „von – Datum“ überprüft. Da das Datum in einem anderen Format übertragen wird, muss es zuerst mit `strtotime()` in ein Zeitstempel und danach mit `date('Y-m-d', ...)` richtig formatiert werden.

Das formatierte Datum kann nun in die Datenbankabfrage integriert werden. Dafür wird eine WHERE – Abschnitt erstellt und in einem Array gespeichert.

```
if($params['FILTERDATE'] == "1")  
    $where .= " AND c.time >= '".mysql_real_escape_string($fromDate)."'  
    AND c.time <= '".mysql_real_escape_string($toDate)."'";
```

Die Funktion `mysql_real_escape_string` wandelt markiert alle nicht erlaubten Sonderzeichen.

Attributtabelle und View erstellen

Bei jeder Änderung der Karteneinstellungen werden die entsprechenden Messdaten aus der Datenbank geholt und in die Attributtabelle geschrieben. Bei Änderungen an mehreren Karten kommt es zu Engpässen in der Datenbank, da immer wieder andere Attribute geschrieben und ausgelesen werden.

Um diese Engpässe zu verhindern, wird für jede Karte eine eigene Attributtabelle und View erstellt. Die Tabelle und die View müssen nur erstellt werden, wenn sie noch nicht vorhanden sind. Aus diesem Grund wird zuerst abgefragt ob eine Tabelle mit dem gleichen Namen existiert und wenn das Ergebnis negativ ist, die entsprechenden Befehle ausgeführt.

```
$rs = pg_query("SELECT relname FROM pg_class WHERE relname =  
'attr_'.${tablename}.'");  
if(pg_num_rows($rs) < 1){  
    pg_query('CREATE TABLE "attr_'.${tablename}.'"  
    (id serial PRIMARY KEY, gis_pathid int4, size int4);');  
    pg_query('CREATE VIEW view_'.${tablename}.'"  
    AS SELECT w.oid, w.*, a.size FROM wegenetz_sihlwald w  
    LEFT JOIN attr_'.${tablename}.'" a  
    ON w.vm_path_gi = a.gis_pathid;');  
    pg_query("INSERT INTO geometry_columns  
    VALUES ('', 'public', 'view_'.${tablename}.'"', 'the_geom',  
    2, 900913, 'MULTILINESTRING');");  
}
```

Datenquelle anpassen

Im MapFile ist für die Datenquelle der Konfigurationsparameter DATA vorgesehen. Dieser muss mit MapScript auf die aktuelle View geändert werden. Dafür wird zuerst eine Kopie der bestehenden Konfiguration angelegt.

```
$oMap = ms_newMapobj("wms.map");  
$newMap = $oMap->clone();
```

Danach wird die entsprechende Ebene gesucht und dem Konfigurationsparameter DATA den Namen der neu erstellten View zugewiesen.

```
$oLayer = $newMap->getLayerByName("wege");  
$oLayer->set(data, "the_geom from view_'.${tablename}');
```

13 Probleme

13.1 GeoExt / OpenLayers und Internet Explorer

Obwohl die Applikation mit Firefox und Chrome ohne Fehler angezeigt wird, zeigt der Internet Explorer eine leere Seite und bringt einen Fehler.

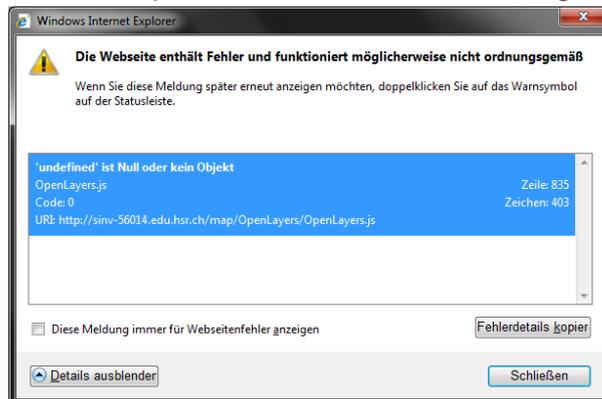


Abbildung 33: Fehlermeldung im Internet Explorer

Das Problem liegt darin, dass GeoExt und OpenLayers mit Arrays konfiguriert werden. Internet Explorer erlaubt es im Gegensatz zu anderen Browsern nicht, leere Objekte in Arrays zu verwenden. Das bedeutet, irgendwo im Code existiert ein Array, das mit einem Komma abgeschlossen wird.

Fehlerhaftes Beispiel:

```
controls: [  
    new OpenLayers.Control.KeyboardDefaults(),  
    new OpenLayers.Control.MouseDefaults(),  
    new OpenLayers.Control.PanZoomBar(),  
]
```

Wenn das Komma am Schluss entfernt wird, funktioniert die Applikation auch im Internet Explorer.

Eine weitere Besonderheit von Internet Explorer macht sich bemerkbar, wenn ein ExtJS Form-Element existiert, das kein `fieldLabel` angegeben hat. Wenn ein Formularelement keine Beschriftung erhalten soll, kann dies mit dem Parameter `hideLabel: true` konfiguriert werden. Wenn nicht alle Form-Elemente entweder ein `hideLabel` oder ein `fieldLabel` Parameter haben, meldet sich der Internet Explorer mit einer Warnung. Die Seite wird jedoch trotzdem vollständig geladen.

13.2 Überlappende Beschriftungen werden von MapServer nicht dargestellt

Wenn sich zwei Beschriftungen bei der Visualisierung überschneiden, wird nur eine angezeigt. Der MapServer entscheidet sich selber für eine Beschriftung.

Der MapServer sollte so konfiguriert werden, dass alle Zahlen dargestellt werden. Im MapFile kann dies mit der Konfiguration `FORCE true` im Bereich `LABEL` erreicht werden. Ein ähnlicher Parameter existiert in SLD jedoch nicht, daher müsste mit unterschiedlichen (negativen) Abständen zwischen den Beschriftungen getestet werden.

13.3 Ladeanzeige nicht möglich

ExtJS bringt mit Ext.LoadMask eine Ladeanzeige mit, die über jedes Element und jede Komponente gelegt werden kann. Die Ladeanzeige verhindert das Bearbeiten von Elementen unter der Lademaske, bis alles geladen ist. Der Zeitpunkt des Ein- und Ausblendens kann durch den Entwickler bestimmt werden.

Leider ist es nicht möglich, diese Lademaske oder sogar einen Ladebalken anzuzeigen, da weder ExtJS noch OpenLayers eine Möglichkeit bieten, zu überprüfen, wann die Elemente fertig geladen sind. In ExtJS gibt es den Event `loadend`, dieser wird jedoch schon gefeuert, wenn noch nicht alle Elemente geladen sind. Für die Anzeige, wann alle Bilder geladen sind, gibt es weder ein Event noch eine Statusabfrage.

Eine Lösung dieses Problems kann die Kartengenerierung mit WFS auf dem Client bringen. Da die Karte erst generiert werden kann, wenn die Geodaten übertragen worden sind, hat man eine bessere Kontrolle über den Status.

13.4 Verschieben der Karte während der Ladephase

Wird während des Ladens eines Themenlayer die Karte verschoben, kann es vorkommen, dass die Themenlayer nicht mehr sauber über dem Basislayer sind. Wie im Kapitel 13.3 „Ladeanzeige nicht möglich“ kann nicht überprüft werden, ob irgendwelche Daten am laden sind oder wann die Karte vollständig geladen ist. Daher kann das Verschieben und Zoomen auch nicht deaktiviert werden während dieser Zeit.

14 Schlussfolgerungen

14.1 Was wurde erreicht

Die Wünsche und Anforderungen der Endbenutzer wurden analysiert und in einem Lösungskonzept zusammengetragen. Es wurde erkannt, dass neben der Visualisierung der Besucherfrequenzen und Besucherströme auch das Berechnen der Gesamtbesucherzahl und das Erstellen eines Besucherstrommodells mit Einbezug aller Wege gewünscht werden. Diese Anforderungen hätten den zeitlichen Rahmen der Arbeit gesprengt und man hat sich auf die Visualisierung der Besucherfrequenzen von vorhandenen Messdaten begrenzt.

Das Lösungskonzept wurde in drei Konzepte unterteilt. Das **Ansichtskonzept** zeigt auf, in welcher Form die Besucherzahlen visualisiert werden können. Im **Darstellungskonzept** wird beschrieben, wie die Unterschiede zwischen den Besucherzahlen erkennbar dargestellt werden können. Mit dem **Filterkonzept** wurden Methoden entwickelt, wie die Besucherzahlen sinnvoll gefiltert werden können.

Im Realisierungskonzept wurden Technologien und Methoden evaluiert, die als Hilfsmittel gebraucht wurden, um das Lösungskonzept zu implementieren. Als Resultat der Evaluation wurde beschlossen, eine Webapplikation mit den Technologien ExtJS und OpenLayers für die Visualisierung der Karte und MapServer mit MapScript auf der Serverseite zu erstellen.

Mit dem implementierten Prototyp wurde aufgezeigt, dass es möglich ist, das erstellte Lösungskonzept zu realisieren. Die Webapplikation wurde mit Grundfunktionen ausgestattet, die es erlaubt, mehrere Karten und Diagramme anzuzeigen, die individuell gefiltert und in der Darstellung angepasst werden können.

14.2 Was wurde nicht erreicht

Die Grundfunktionen für die Visualisierung sind im Prototyp enthalten. Nicht implementiert wurde die Detailansicht (Prototyp). Die Detailansicht ist nichts anderes, als ein Fenster mit Texten und Tabellen, darum wurde aus zeitlichen Gründen auf die Implementierung verzichtet.

Auch aus zeitlichen Gründen wurde auf das Einfügen und Anzeigen von Erfahrungswerten verzichtet (Prototyp 7).

14.3 Erweiterungen für die Webapplikation

14.3.1 WFS statt WMS

Mit der Umstellung von WMS auf den WFS Dienst, verschiebt sich die Erstellung der Karte vom Server auf den Client. Dies hat viele Vorteile zu bieten:

- Der grösste Geschwindigkeitsverlust ist bei Webanwendungen durch zuviele Requests zu erkennen. Durch WFS wird ein schnelleres Laden der Ebenen erreicht, da nicht so viele Bildanfragen durch Browser gesendet werden müssen
- Die Daten stehen dem Client zum Beispiel für Darstellung der Detailinformationen zur Verfügung
- Elemente im Themenlayer können mit einem MouseOver Effekt belegt werden und somit weitere Informationen anzeigen

Die Umstellung hat auch zur Folge, dass das SLD nicht mehr auf dem Server, sondern auf dem Client erstellt und übergeben werden muss.

14.3.2 Sessionhandling auf Server

Im Prototyp werden die Messdaten bei jedem Kartenteil aus der Messdatenbank geladen und in die Attributtabelle geschrieben.

Mit einem Sessionhandling im PHP / MapScript kann das aufwändige Lesen und Schreiben auf ein einziges Mal reduziert werden. Dazu muss jedoch bei jeder Anfrage überprüft werden, ob die Parameter geändert haben.

Das Resultat dieser Massnahme zeigt sich in wesentlich schnelleren Antwortzeiten und damit schnellerem Laden der Karte und der Themenlayer.

14.3.3 Mehrere Werte im gleichen Diagramm

Um Diagramme sinnvoll auszunutzen, sollte es möglich sein, mehrere Filterungen im gleichen Diagramm anzeigen zu können, was von ExtJS unterstützt wird. Dazu müssen die Diagrammeinstellungen erweitert werden, damit es möglich wird, neue Filteroptionen dynamisch hinzuzufügen.

Bei mehr Filterungen pro Diagramm sind besser Vergleiche möglich.

14.3.4 Kartenbereich auf Parkgrösse beschränken

Im aktuellen Prototyp kann die Karte auch auf ein Gebiet ausserhalb des Parkgebietes navigiert werden. Damit beim Verschieben der Karte die entsprechenden Grafiken nicht immer nachgeladen werden müssen, lädt OpenLayers die angrenzenden Kartengrafiken von Beginn an. Dies führt zu längeren Ladezeiten und zu mehr Requests.

14.4 Ausblick

Mit dem Prototyp wird gezeigt, dass das Lösungs- wie auch das Realisierungskonzept möglich ist. Die Webapplikation kann nun ausgebaut und erweitert werden, damit ein produktives System entsteht.

Glossar

Abkürzung	Name	Beschreibung
Ajax	Asynchronous JavaScript and XML	Konzept für eine asynchrone Datenübertragung
CGI	Common Gateway Interface	Schnittstelle für Datenaustausch zwischen Webserver und dritter Software / Skripte
CSS	Cascading Style Sheets	Sprache zur optischen Gestaltung von strukturierten Dokumenten
EPSG	European Petroleum Survey Group	Beschreibung von Koordinatenreferenzsystemen
ExtJS		Framework zur Erstellung von Webapplikationen mit JavaScript und Ajax
GDAL	Geospatial Data Abstraction Library	Übersetzungsbibliothek für Raster-Geodaten
GEOS	Geometry Engine Open Source	Bibliothek für die Handhabung von Geometrien
GET		Übertragen von Parametern in einer URI
GIS	Geoinformationssystem	System zur Erfassung und Bearbeitung von geografischen Daten
HTML	HyperText Markup Language	Auszeichnungssprache zur Darstellung und Strukturierung von Inhalten (Text, ...)
JSON	JavaScript Object Notation	Schlankes Datenaustauschformat für den Webbereich
OID	Object Identifier	Eindeutiger Bezeichner eines Objektes
OpenLayers		JavaScript Bibliothek, die es ermöglicht, Geodaten im Browser darzustellen
OSM	OpenStreetMap	Sammlung von freien Karten (Geodaten)
PHP	PHP Hypertext Preprocessor	Verbreitete Skriptsprache für die Erstellung dynamischer Webseiten
POI	Point of Interest	Interessanter Ort, Sehenswürdigkeit
POST		Übertragung von Parametern im Header
PostGIS		Erweitert eine PostgreSQL Datenbank um geografische Objekte und Funktionen
Postgres	PostgreSQL	Objektrelationales Datenbankmanagementsystem

Proj.4		Bibliothek für die Projektion von Karten
Python		Skriptsprache
RIA	Rich Internet Application	Anwendung im Web, die eine intuitive Benutzeroberfläche bietet
SLD	Styled Layer Descriptor	Format, um das Aussehen von Kartenebenen zu definieren
SRID	Spatial Reference Identifier	Eindeutige Erkennnummer von Koordinatensystemen
SRS	Spatial Reference System	Räumliches Bezugssystem
View	Datenbanksicht	Virtuelle Tabelle aus Daten von anderen Tabellen
WCS	Web Coverage Service	Schnittstelle zum Abrufen eines Kartenteils aus einer Rastergrafik
WFS	Web Feature Service	Schnittstelle zum Abrufen von Geodaten
WFS-T	Transactional WFS	Ermöglicht den WFS Funktionen das Hochladen von Geodaten
WGS	World Geodetic System	Koordinatensystem in dezimalen Grad, WGS84 wird z.B. in GPS verwendet
WMS	Web Map Service	Schnittstelle zum Abrufen einer Karte aus Geodaten in Form einer Rastergrafik
WPS	Web Processing Service	Ermöglicht die Durchführung räumlicher Analysen von Geodaten im Internet
XML	Extensible Markup Language	Auszeichnungssprache zur Darstellung hierarchisch strukturierter Daten

Quellen

Literaturverzeichnis

- [1] Präsentation „SANU-Workshop“ von Dominik Siegrist
100324_Siegrist_VISIMAN_print.pdf
- [2] Präsentation „Internet Vektorgraphiken & WebGIS“ von Dominik Ramp
webgis_dominik_ramp.pdf
- [3] Auswertung Server Implementationen
http://gis.hsr.ch/wiki/UMN_MapServer (April 2010)
- [4] Ka-Map Wiki
<http://ka-map.ominiverdi.org/wiki/> (April 2010)
- [5] Ka-Map – Offizielle Projektwebsite
<http://ka-map.maptools.org/> (April 2010)
- [6] Unterschied OpenLayers, MapFish und MapBender:
[http://www.mapbender.org/Eine Typisierung: OpenLayers, Mapbender und MapFish](http://www.mapbender.org/Eine_Typisierung:_OpenLayers,_Mapbender_und_MapFish) (April 2010)
- [7] Spatial Reference – Sammlung von EPSG Codes
<http://www.spatialreference.org/> (April 2010)
- [8] MapFish – Offizielle Projektwebseite
<http://mapfish.org/> (April 2010)
- [9] MapBender – Offizielle Projektwebseite
<http://www.mapbender.org/> (April 2010)
- [10] OpenLayers – Offizielle Projektwebseite mit Dokumentation und Beispielen
<http://www.openlayers.org/> (Juni 2010)
- [11] GeoExt – Offizielle Projektwebseite mit Dokumentation und Beispielen
<http://www.geoext.org/> (April 2010)
- [12] GeoServer – Offizielle Projektwebseite
<http://www.geoserver.org/> (April 2010)
- [13] MapServer – Offizielle Projektwebseite mit Dokumentation
<http://www.mapserver.org/> (Juni 2010)
- [14] ExtJS – Offizielle Projektwebseite
<http://www.extjs.com/> (Juni 2010)
- [15] Bill Kropla. Beginning MapServer: Open Source GIS Development.
Apress (2005)
- [16] Thorsten Fischer. UMN MapServer 4.0: Handbuch und Referenz.
MapMedia (2003)

Abbildungsquellen

[Abbildung 1] & [Abbildung 4]

Aus der OpenSource Software Quantum GIS

[Abbildung 2] & [Abbildung 3]

Microsoft Office Word 2007 Diagramm

[Abbildung 5] – [Abbildung 7]

Aus der Präsentation SANU-Workshop

„100324_Siegrist_VISIMAN_print.pdf“ von Dominik Siegrist

[Abbildung 8]

Microsoft Office Word 2007 Diagramm

[Abbildung 9] – [Abbildung 17]

Microsoft Office Visio 2007

[Abbildung 19]

<http://geoconverter.hsr.ch/> (21. April 2010)

[Abbildung 20]

Microsoft Visio 2007 mit Zusatz „Sketch GUI shapes“

[Abbildung 25]

Microsoft Visio 2007

Abbildungsverzeichnis

Abbildung 1: automatische Zählstellen Sihlwald	11
Abbildung 2: Vergleich der Besucherzahlen nach Wochentagen	11
Abbildung 3: Durchschnittliche Besucherzahl pro Wochentag	12
Abbildung 4: manuelle Zählstellen Sihlwald	12
Abbildung 5: Ansichtskonzept Gesamtübersicht	24
Abbildung 6: Ansichtskonzept Nahansicht	25
Abbildung 7: Ansichtskonzept Vergleichsansicht	26
Abbildung 8: Ansichtskonzept Statistik (Säulen- und Tortendiagramm)	27
Abbildung 9: Realisierungskonzept - Logik auf Client	33
Abbildung 10: Realisierungskonzept - Serverseitige Java Anwendung	34
Abbildung 11: Realisierungskonzept - WPS	34
Abbildung 12: Realisierungskonzept - WMS und WFS	35
Abbildung 13: Verknüpfung Mess- und Geodaten durch eine Attributtabelle und View	40
Abbildung 14: Verknüpfung Mess- und Geodaten durch ändern der Konfiguration	41
Abbildung 15: Verknüpfung Mess- und Geodaten durch SLD	42
Abbildung 16: Verknüpfung Mess- und Geodaten auf dem Client	43
Abbildung 17: Verknüpfung Mess- und Geodaten mit WFS und SLD	43
Abbildung 18: MapFile Export mit Quantum GIS	51
Abbildung 19: Shapefiles Online mit GeoConverter konvertieren	52
Abbildung 20: Entwurf der Benutzeroberfläche	56
Abbildung 21: Implementierte Benutzeroberfläche	57
Abbildung 22: Verschiedene Basislayer zum Auswählen	58
Abbildung 23: Bedienelement - Ebenen ein-/ausblenden	59
Abbildung 24: Visualisierung der Themenlayer auf einem Basislayer	59
Abbildung 25: Zusammenspiel zwischen Server und Client bei der Kartenerstellung	60
Abbildung 26: Bedienelement Anzeigeeinstellungen	61
Abbildung 27: Visualisierung mit Wegbreite und Farben	61
Abbildung 28: Visualisierung mit Pfeilen	62
Abbildung 29: Beispiel eines Torten- und Balkendiagramm	63
Abbildung 30: Bedienelement Diagrammeinstellungen	63
Abbildung 31: Bedienelement Filter	64
Abbildung 32: Filterung nach Wochentagen	64
Abbildung 33: Fehlermeldung im Internet Explorer	69

A - Aufgabenstellung



Bachelorarbeit, FS 2010

Visualisierung eines Besucherstrommodells

Aufgabenstellung

Einführung

Im Rahmen des KTI-Projekts VISIMAN (KTI: Kommission für Technologie und Innovation) wird erstmals ein IT-basiertes Werkzeug für das Besuchermanagement in Natur- und Nationalparks entwickelt. Das endgültige System soll heterogene Komponenten zu einem Gesamtsystem integrieren können und umfasst folgende Subsysteme:

- Besuchermonitoringsystem
- Experteninformationssystem
- Besucherinformationssystem
- Dispatcher

In vorangegangenen Bachelor- bzw. Studienarbeiten wurde in einer Technologiestudie die Eignung von Nagios als Dispatcher veranschaulicht. Durch die Implementation wurde gezeigt, dass sich Nagios als Dispatcher hervorragend eignet, verschiedene Sensoren anzubinden, deren Messwerte abzufragen, Auswertungen zu generieren und Alarmierungen auszulösen.

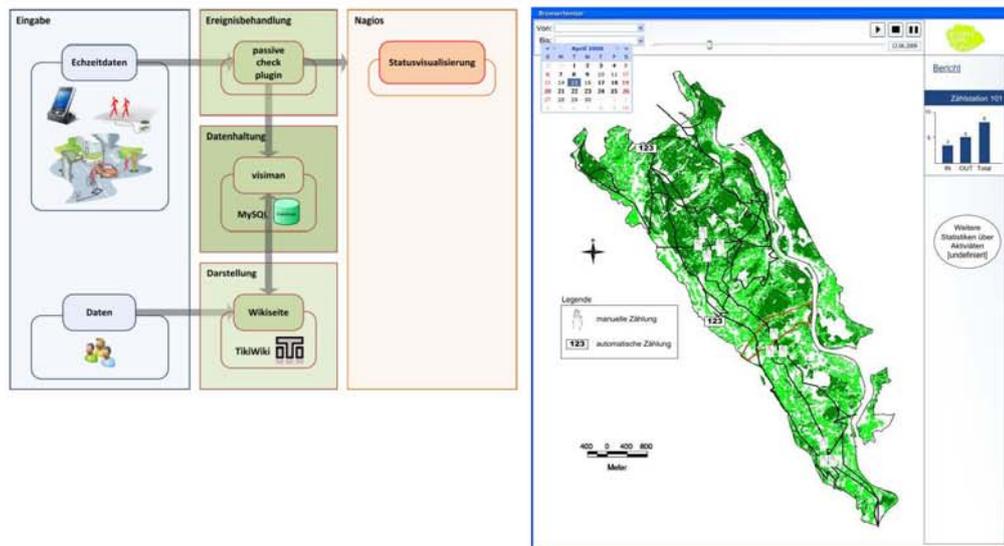
Darauf basierend wurde in einer weiteren Studienarbeit die drahtlose Anbindung einer mobilen Applikation als Ersatz einer Zählmatte an das Nagios System realisiert.

Thematik

Aufbauend auf dem vorhandenen Datenmodell und den vorhandenen Daten sollen für die Monitoringdaten Auswertungen entworfen und implementiert werden, welche der Parkverwaltung als Grundlage für die Besucherlenkungsmassnahmen dienen. Die Daten liegen dabei als Eingabedaten oder Echtzeitdaten in der Datenbank visiman vor.

Die besondere Herausforderung besteht darin, neben der Darstellung von Monitoringdaten, auch räumliche Geoinformationen im ArcGIS Format zu visualisieren um daraus ein Besucherstrommodell zu entwickeln.

Es werden Webtechnologien bevorzugt, da eine nahtlose Integration in das bereits vorhandene Wiki-System TikiWiki erwünscht wird. Die Arbeit bietet zudem Raum für weitere Auswertungsvorschläge.



Thema dieser Arbeit & Aufgabe

Im Rahmen dieser Arbeit soll ein Konzept für eine Visualisierung von Besucherströmen entworfen und als Prototyp realisiert werden. D.h. es sollen

- die Bedürfnisse der Endbenutzer erfasst und analysiert werden
- Lösungskonzepte für eine Visualisierung entworfen werden
- ein oder mehrere dieser Lösungskonzepte als Prototyp realisiert werden
- diese Prototypen mit den Endbenutzern überprüft werden

Vorgehen, Anforderungen, Lösungs- und Realisierungskonzepte sind in enger Zusammenarbeit mit dem Betreuer und dem Projektmitarbeiter Mathias Manz zu erarbeiten.

Es soll ingenieurmässig vorgegangen werden.

Es ist nach einem Projektplan zu arbeiten. Der tatsächliche Arbeitsaufwand ist zu erfassen.

Erwartete Resultate

Lösung der vorstehend beschriebenen Aufgabe.

Abzugeben ist ein Bericht (in 1 Exemplar auf Papier, in 3 Exemplaren als CD-ROM), welcher enthält:

- die Aufgabenstellung der Arbeit (im Original),
- eine Zusammenfassung der Arbeit auf einer Seite auf dem dafür vorgegebenen Formular,
- eine Management-Summary,
- einen Bericht über die Arbeit,
- die bei der Arbeit erstellten Dokumente,
- die erstellten Programme (als Softcopy),
- den Projektplan und die tatsächlichen Aufwände.

Rechte

Die Rechte werden in der nachstehenden Vereinbarung geregelt.

Termine

Abgabe der Aufgabenstellung und Beginn der Arbeit: 22.2.2010

Abgabe des Berichts und Ende der Arbeit: 18.6.2010, 17:00 h

Betreuung

Betreuer: Lothar Müller

Mit dem Betreuer werden regelmässige Sitzungen durchgeführt. Bei Bedarf können weitere Termine vereinbart werden.

Rapperswil, 22.2.2010

Prof. Dr. Lothar Müller

Vereinbarung

1. Gegenstand der Vereinbarung

Mit dieser Vereinbarung werden die Rechte über die Verwendung und die Weiterentwicklung der Ergebnisse der Bachelorarbeit "Visualisierung eines Besucherstrommodells" von Marcel Germann unter der Betreuung von Prof. Dr. Lothar Müller geregelt.

2. Urheberrecht

Die Urheberrechte stehen dem Studenten zu.

3. Verwendung

Die Ergebnisse der Arbeit dürfen von der HSR und von den Partnern des KTI-Projekts nach Abschluss der Arbeit verwendet und weiter entwickelt werden.

Rapperswil, den 22.2.2010

.....
Marcel Germann

Rapperswil, den 22.2.2010

.....
Prof. Dr. Lothar Müller

B - Protokoll der Sitzung mit Herr Dominik Siegrist

Sitzungsprotokoll

Visualisierung eines Besucherstrommodells

Zweck	Teammeeting
Datum	22.03.2010
Zeit	16:00
Ort	6.111
Teilnehmer	Dominik Siegrist Mathias Manz Marcel Germann
Entschuldigt	Lothar Müller
Protokollführer	Marcel Germann

Traktandum	Zweck der Visualisierung
Inhalt	<ul style="list-style-type: none"> Die Visualisierung des Besucherstroms dient zu einer zeitlichen Darstellung der Besucherfrequenz und der räumlichen Verteilung im Gebiet.
Diskussion	

Traktandum	Benutzung
Inhalt	<ul style="list-style-type: none"> Ziel ist es zwei Visualisierungen zu haben <ul style="list-style-type: none"> Eine ausführliche für das Parkmanagement Eine für die Öffentlichkeit (sehr abgespeckt, z.B. nur ein Bild) Es kann davon ausgegangen werden, dass das Parkmanagement das Gebiet gut kennt und Grundfunktionen am PC beherrscht.
Diskussion	<ul style="list-style-type: none"> Die öffentliche Visualisierung würde den Rahmen dieser Arbeit sprengen und wird daher nicht in die Konzeptlösung und den Prototyp eingebaut. Der Prototyp dient nur zum Beweis, dass die Konzeptlösung funktioniert, er wird nicht vollständig funktional sein.

Traktandum	Abdeckung
Inhalt	<ul style="list-style-type: none"> Der Sihlwald wird einer der besser ausgebauten Pärke sein. Sihlwald ist ein eher kleines Gebiet, das viele Besucher hat. Gemäss Angaben des Herstellers Eco-Counter werden für ein Gebiet wie den Sihlwald 5 bis 10 Zählmatten gebraucht.
Diskussion	

Marcel Germann

24.03.10

Seite 1 / 2

Sitzungsprotokoll

Visualisierung eines Besucherstrommodells

Traktandum	Auswertung
Inhalt	<ul style="list-style-type: none"> • Im Sihlwald wurden die Wege bereits in 3 Kategorien unterteilt (kann zum Beispiel für Erfahrungswerte gebraucht werden) • Die Zählmatten sind installiert und liefern Daten • Die Visualisierung soll nun diese beiden Daten verbinden und anzeigen • Die Zählmatten sind besonders an folgenden Orten interessant <ul style="list-style-type: none"> ○ bei Standorten mit grossen Frequenzen ○ bei kritischen Punkten (z.B. Naturschutzgebiet) ○ bei speziellen Orten (z.B. POIs, Restaurant, Ein- und Ausgänge) • Eine Idee für die Visualisierung existiert bereits <ul style="list-style-type: none"> ○ Wird von Herr Siegrist per E-Mail verschickt • Die Richtung des Besucherstroms ist nicht priorisiert. Daher sollten statt Pfeile besser verschiedenfarbige und –grosse Striche verwendet werden • Balkendiagramme können zum Beispiel für Vergleiche benutzt werden. • Die Visualisierung soll mit Symbolen für kritische Punkte und POIs ergänzt werden können. • Eine Schnittstelle zu einer Agenda wird gewünscht. • Die Auswertung soll erweiterbar sein. • Das Wetter spielt eine grosse Rolle für die Auswertung • Man ist sehr interessiert aus dem Besucherstrommodell die Gesamtbesucherzahl zu bestimmen → schwierig realisierbar, sehr mathematisch
Diskussion	

Traktandum	Filterung
Inhalt	<ul style="list-style-type: none"> • Zu den bereits erwähnten Filterungen sind folgende Filtermöglichkeiten besonders wichtig: <ul style="list-style-type: none"> ○ Wegkategorien / Zonen ○ Wetter ○ Momentaufnahme
Diskussion	

C - Protokoll der Sitzung mit Herr Ronald Schmidt

Sitzungsprotokoll

Visualisierung eines Besucherstrommodells

Zweck	Teammeeting
Datum	13.04.2010
Zeit	08:00
Ort	Universität Irchel
Teilnehmer	Ronald Schmidt Lothar Müller Mathias Manz Marcel Germann
Protokollführer	Marcel Germann

Traktandum	Zweck der Visualisierung
Inhalt	<ul style="list-style-type: none"> • Räumliche Verteilung der Besucher • Besucherströme an Kreuzungen (von Wo – nach Wo) • Gesamtbesucherzahl berechnen • Flächendeckend Besucherfrequenzen auswerten • Interessen der Besucher herausfinden
Diskussion	<ul style="list-style-type: none"> • Die Gesamtbesucherzahl ist schwierig berechenbar, da alle grossen Eingänge überwacht werden müssten und Besucher teilweise mehrfach gezählt würden • Besucherströme der Routen durch das ganze Gebiet sind nicht relevant • Besucherströme an Kreuzungen durch automatische Zählungen schwierig berechenbar

Traktandum	Was soll mit Besucherstrommodell erreicht werden
Inhalt	<ul style="list-style-type: none"> • Vereinfachung der Erstellung einer Visualisierung • Berechnen der Gesamtbesucherzahl • Erfahrungen aus der Visualisierung herauslesen können
Diskussion	<ul style="list-style-type: none"> • Für Gesamtbesucherzahl im Park müssten alle Eingänge überwacht werden • Besucherstrommodell automatisch zu erstellen ist nicht sinnvoll <ul style="list-style-type: none"> ○ Zuerst muss eine Visualisierung existieren um Erfahrungen herauslesen zu können ○ Mit den neuen Erfahrungen kann ein Modell entwickelt werden ○ Automatisches Erstellen des Modells schwierig

Traktandum	Anforderung an die Visualisierung
Inhalt	<ul style="list-style-type: none"> • Besucherstromrichtung für Auswertung an Kreuzungen interessant • Am wichtigsten ist die Visualisierung der Besucherfrequenzen in der Gesamtübersicht • Vergleich von verschiedenen Attributen sollte möglich sein, ob die

Marcel Germann

16.04.10

Seite 1 / 3

Sitzungsprotokoll

Visualisierung eines Besucherstrommodells

Diskussion	<p>Auswertungen gleichzeitig oder nacheinander angezeigt werden ist nicht relevant</p> <ul style="list-style-type: none"> • Besucherzahlen von POIs will man überwachen zur Auswertung der Entwicklung über mehrerer Jahre hinweg • Filter <ul style="list-style-type: none"> ○ Sehr wichtig ist die Filterung nach Attributen ○ Wochentage sollen zusammengefasst werden können ○ Wechsel zwischen manuellen und automatischen Zählungen ○ Filterung nach Events und Wetter wird nicht benötigt • Darstellung der POIs auf der Visualisierung wäre interessant, aber nicht prioritär und nur wenn die Visualisierung nicht unübersichtlich wird <p>• Beispiel für die Nutzung der Besucherstromrichtung: Wenn z.B. Velofahrer in eine verbotene Zone fahren, soll ausgewertet werden können, von welcher Seite her die Tafel aufgestellt werden soll</p>
------------	---

Traktandum	Standortwahl für die Zählstellen
Inhalt	<ul style="list-style-type: none"> • Die Zählstellen sind an <ul style="list-style-type: none"> ○ Stark frequentierten Bereiche ○ Übergänge in die Kernzone (grosse Eingänge) ○ Zu überwachende Standorte (z.B. POIs) • Ziel ist es alle grossen Eingänge abzudecken • Durch die manuellen Zählungen sollen Erfahrungen über die Interessen der Besucher gesammelt werden
Diskussion	<ul style="list-style-type: none"> • Eingang am Albishorn und der südliche Teil des Gebietes ist nicht abgedeckt • Manuelle Zählungen sind in ein paar Jahren geplant zur Auswertung der Entwicklung • Automatische Zählungen werden je nach Entwicklung neu platziert • In nächster Zeit werden zwei neue Zählmatten in der Nähe Besucherzentrum vergraben

Traktandum	Erfahrungswerte
Inhalt	<ul style="list-style-type: none"> • Kategorisierung der Wege wurde durch Beobachtungen der Ranger erstellt • Verknüpfungen der Erfahrungswerte und der Besucherdaten erst in einem Strommodell interessant • Visualisierung der Erfahrungswerte wird nicht benötigt, Herr Schmidt sieht keine Nutzung der Informationen in der Visualisierung
Diskussion	<ul style="list-style-type: none"> • Ziel ist es Erfahrungen und Wegkategorien aus der Visualisierung herauslesen zu können, dafür wird zuerst eine Auswertung mit der Visualisierung benötigt

Sitzungsprotokoll

Visualisierung eines Besucherstrommodells

Traktandum	Ideen für Visualisierung
Inhalt	<ul style="list-style-type: none"> • Bevorzugt werden Dicke der Wege, Zahlen und evtl. die Richtung • Diagramme sind interessant bei Auswertung der Gesamtzahlen z.B. bei einem Tages- oder Wochenverlauf • Quantisierung in Stunden genügt • Diagramme auf einer Karte sind zwar möglich, jedoch nicht prioritär, da schnell unübersichtlich
Diskussion	<ul style="list-style-type: none"> • Diagramme sind schwierig in einer Auswertung, da kein Zusammenhang zu einer Karte vorhanden ist

Traktandum	ArcGIS
Inhalt	<ul style="list-style-type: none"> • Visualisierung (dicke der Wege) kann mit ArcGIS bereits gemacht werden, jedoch ist die Bedienung nur mit guten ArcGIS Kenntnissen und einem gewissen Aufwand verbunden • Sihlwald besitzt Lizenzen für alle ArcGIS Tools • Eingesetzt werden zurzeit <ul style="list-style-type: none"> ○ ArcGIS Desktop / Workstation für Ein- und Ausgabe ○ ArcGIS Server für Daten ○ Trackinganalyst für zeitliche Darstellung der Daten • ArcGIS wird in der neusten Version 9.3 verwendet der Trackinganalyst beinhaltet in der neusten Version jedoch noch ein paar Fehler, weshalb für zeitliche Auswertungen die Version 9.2 verwendet wird

D - Protokoll der Sitzung mit Herr Stefan Keller

Sitzungsprotokoll

Visualisierung eines Besucherstrommodells

Zweck	Besprechung Möglichkeiten für Prototyp
Datum	16.04.2010
Zeit	13:15
Ort	6.108
Teilnehmer	Stefan Keller Marcel Germann
Protokollführer	Marcel Germann

Traktandum	GIS Erweiterungen
Inhalt	<ul style="list-style-type: none"> • ArcExplorer <ul style="list-style-type: none"> ○ + ist kostenlos (wird von ESRI bereitgestellt) ○ + shapefiles können importiert werden ○ + unterstützt diverse Abfragen ○ - ist nicht programmierbar ○ → nicht brauchbar, da nicht programmierbar • Google Earth <ul style="list-style-type: none"> ○ Keine Integration weiterer Daten <ul style="list-style-type: none"> ▪ Shapefiles können nicht importiert werden ▪ Keine Verbindung zu Geodaten-Server • Arc GIS <ul style="list-style-type: none"> ○ Kostenpflichtig ○ Sehr komplex in der Installation und Konfiguration ○ Aufgeteilt in diverse Teilapplikationen <ul style="list-style-type: none"> ▪ ArcGIS Server → hat kein UI ▪ ArcMap → stellt Daten dar, Import über: <ul style="list-style-type: none"> • Shapefiles • SDE → Oracle / Postgres Datenbank • Geodatenbank mit mdb (MS Access) • Geodatenbank Ordnerbasiert ○ Erweiterungen möglich <ul style="list-style-type: none"> ▪ Python ▪ C# ▪ (Java) → keine gute Unterstützung durch ESRI ▪ (VBA Makros) → möglich, wird jedoch von Microsoft nicht mehr unterstützt • Quantum GIS <ul style="list-style-type: none"> ○ Kostenlos ○ Einfache Erweiterung durch PlugIns <ul style="list-style-type: none"> ▪ Programmierung durch Python ○ Geodatenbank mit SQLite oder Postgres • Vorteile von GIS Erweiterungen: <ul style="list-style-type: none"> ○ Schnelle Anzeige

Marcel Germann

16.04.10

Seite 1 / 3

Sitzungsprotokoll

Visualisierung eines Besucherstrommodells

	<ul style="list-style-type: none"> • Nachteile: <ul style="list-style-type: none"> ○ Desktop Applikation notwendig (Installation) ○ Keine oder nur schwierige Portierung auf andere GIS Applikationen ○ Schwierige Implementation von eigenen Grafiken (z.B. Diagramme) ○ Schwierig andere Bedienelemente auszublenden um Bedienung zu vereinfachen
--	---

Traktandum	Eigene Applikation
Inhalt	<ul style="list-style-type: none"> • Java 2D <ul style="list-style-type: none"> ○ Bibliotheken zu GIS (z.B. ArcGIS) • Vorteile: <ul style="list-style-type: none"> ○ Unabhängig zu Frameworks ○ GIS APIs z.B. in Java vorhanden • Nachteile: <ul style="list-style-type: none"> ○ Sehr aufwendig ○ Viele Funktionen (Zoom, Auswahl, ...) müssen selber implementiert werden ○ Webapplikation mit hoher wahrscheinlichkeit schneller ○ Braucht separate Installation

Traktandum	Webapplikationen
Inhalt	<ul style="list-style-type: none"> • Logik auf Client <ul style="list-style-type: none"> ○ Karte aus Rastergrafik (Basislayer) ○ Visualisierung wird von Client gemacht (Themenlayer) ○ Server liefert nur Daten ○ Client verarbeitet Daten und zeigt diese an • Serverseitige Java Anwendung <ul style="list-style-type: none"> ○ Basislayer aus Rastergrafik ○ Server holt Daten aus Geodatenbank ○ Server verarbeitet Daten und erstellt Visualisierung ○ Daten werden mit eigenem JSON Protokoll übertragen ○ Client stellt Daten nur dar • WPS (Web Processing Service) <ul style="list-style-type: none"> ○ Basislayer aus Rastergrafik ○ Themenlayer durch WPS Server (z.B. Python WPS Server) ○ Benutzt standartisiertes Protokoll (ähnlich SOAP) ○ Python WPS Server unterstützt: <ul style="list-style-type: none"> ▪ GeoDatenbank (PostGIS) ▪ Shapefiles ○ Client bekommt nur Daten und muss selber um Darstellung

Sitzungsprotokoll

Visualisierung eines Besucherstrommodells

	<p>kümmern</p> <ul style="list-style-type: none"> • WMS und WFS <ul style="list-style-type: none"> ○ Client (Javascript) nur für Navigation und Darstellung der Karte ○ Basislayer über WMS (Karte) ○ Themenlayer über WFS (Vektordaten) ○ Server implementationen <ul style="list-style-type: none"> ▪ MapServer <ul style="list-style-type: none"> • Einfach • Verständliche Konfiguration • Zugriff auf alle Datenquellen (DB, shapefiles, ...) • Nur Lesen der Geodaten <ul style="list-style-type: none"> ○ Schreiben nur direkt über Datenbank ▪ GeoServer <ul style="list-style-type: none"> • Ähnlich wie MapServer • Konfigurierbarer als MapServer • Besitzt mehr Darstellungsformate als MapServer ▪ ArcGIS Server <ul style="list-style-type: none"> • Kein MySQL Server Support • Extrem Komplex (Installation und Konfiguration) • Vorteile: <ul style="list-style-type: none"> ○ Keine Installation notwendig ○ Anzeigen von verschiedenen Kartentypen
--	---

Traktandum	Javascript Frameworks
Inhalt	<ul style="list-style-type: none"> • OpenLayer <ul style="list-style-type: none"> ○ OpenSource ○ Darstellung von WMS, WFS und WCS Daten ○ Darstellung von geschlossenen Karten <ul style="list-style-type: none"> ▪ OpenStreetMap ▪ GoogleMaps ▪ YahooMaps ▪ Virtual Earth ○ Darstellung von weiteren Daten <ul style="list-style-type: none"> ▪ GeoRSS ▪ Vektorgrafiken ○ Editor von Vektordaten integriert • GeoExt <ul style="list-style-type: none"> ○ Nutzt OpenLayer und ExtJS ○ Gute grafische Unterstützung durch ExtJS <ul style="list-style-type: none"> ▪ Viele Diagramme ○ ExtJS ist ziemlich umfangreich → lange Einarbeitung