

Schwachstellenanalyse im Langsamverkehr

Hans-Jörg STARK, Lukas BÄHLER und Christian KARRIÉ

Dieser Beitrag wurde nach Begutachtung durch das Programmkomitee als „reviewed paper“ angenommen.

Zusammenfassung

Mit dem „Agglomerationsprogramm Basel“ der Schweizer Kantone Basel-Stadt, Basel-Land, Aargau, Solothurn sowie den Anliegerstaaten Frankreich und Deutschland wurde ein Programm ins Leben gerufen, das den Fahrrad- und Fußverkehr fördern soll. Das Programm sieht unter anderem die Erfassung und Typisierung von Schwachstellen in der bestehenden Infrastruktur vor. Dazu gehören beispielsweise fehlende Beschilderung, hohe Randsteine, mangelnde Sitzgelegenheiten usf. Nebst der Schwachstellenermittlung sollen auch Maßnahmen zur Verbesserung der Infrastruktur wie Sofortmaßnahmen, Schließen von Netzlücken usf. erhoben werden. Damit diese Schwachstellenanalyse durch unterschiedliche Teilnehmende erfolgen kann, wurde eine geeignete Webanwendung basierend auf Open-Source-Komponenten erstellt. Diese steht unter Berücksichtigung gewisser Berechtigungen den verantwortlichen Akteuren bereit. Sie ist so konzipiert, dass eine möglichst einfache Bedienung ohne Verzicht auf wichtige Informationen nicht GIS-Experten zur Verfügung steht. Damit leistet die Anwendung einen Beitrag im Bereich Public Participation GIS (PPGIS).

1 Einführung

1.1 Ausgangssituation

„Zu Fuß oder mit dem Velo unterwegs zu sein ist gesund, schon das Portemonnaie und die Umwelt. Dieser sogenannte Langsamverkehr ist attraktiv: keine Staus behindern das Vorankommen, die Parkplatzsuche entfällt und auf vielen Strecken ist das Velo das schnellste Fortbewegungsmittel.“ (BAU- UND UMWELTSCHUTZDIREKTION KANTON BASEL-LANDSCHAFT 2010)

So bildet der Langsamverkehr nebst dem motorisierten Individualverkehr und dem öffentlichen Verkehr das dritte Standbein des Personenverkehrs. Sein Wachstumspotenzial ist groß und noch nicht ausgeschöpft (ASTRA 2007). Dies gründet zumindest teilweise auf der Tatsache, dass Schwachstellen im Langsamverkehrsnetz vorliegen. Diese Schwachstellen lassen sich grob in drei Kategorien einteilen:

- Sicherheit (z. B. unbeleuchtete Tunnel, hohes Verkehrsaufkommen)
- Signalisation (z. B. fehlende, verwirrende oder widersprüchliche Signalisation)
- Infrastruktur (z. B. fehlende Querungen, hohe Randsteine, fehlende Sitzgelegenheiten)

Um den Langsamverkehr sicherer und attraktiver zu machen, beschloss das „Agglomerationsprogramm Basel“ für die Gemeinden der Region eine Wegleitung zur Erhebung, Analyse und Beseitigung von Schwachstellen im Langsamverkehr zu verfassen. Diese beschreibt konzeptionell die Situationsprüfung des Fuß- und Veloverkehrs, die Erfassung und Typisierung der Schwachstellen und die Abschätzung von Maßnahmen und hält Vorlagen und Beispiel für Erfassungsformulare, Grundlagenpläne für Feldbegehungen und einen Maßnahmenkatalog bereit.

1.2 Zielsetzungen

Werden die genannten Schwachstellen in einer Gemeinde erfasst und evaluiert, besteht die Möglichkeit, diese gezielt zu beheben. Die Gemeinden erhalten somit sowohl den Auftrag als auch die Grundlagen zur Umsetzung von kurz- und mittelfristigen Pflege- und Unterhaltsmaßnahmen sowie zur Planung von längerfristigen Maßnahmen, um solche erhobenen Schwachstellen zu beseitigen. Das Ziel ist die Steigerung der Attraktivität, Sicherheit und Gesundheit im und durch den Langsamverkehr.

Für die Erfassung vor Ort in einer Gemeinde ist die Gemeinde- und Bauverwaltung oder ein externes Büro zuständig. Vorzugsweise wird der Einbezug der Bevölkerung empfohlen, da diese die Gegebenheiten vor Ort kennt und dadurch eine wichtige Informationsquelle bezüglich den vorliegenden Schwachstellen ist.

1.3 Methodik

Für die Erfassung der Schwachstellen stehen entsprechende Formulare für die Feldbegehung zur Verfügung. Jedoch dienen diese lediglich als Arbeitsmaterial vor Ort. Ziel ist es, diese Schwachstellen in einer zentralen Datenbank digital zu verwalten. Die im Feld erhobenen Formulardaten sollen also über eine webbasierte Anwendung in die zentrale Datenbank eingepflegt werden. Die Anforderungen an eine solche Anwendung, welche die Problematik von Schwachstellen im Langsamverkehr dokumentiert, wie sie durch Laien und Experten erhoben werden, entspricht der Grundidee von Public Participation GIS (PPGIS) (RAMASUBRAMANIAN 2008, TULLOCH 2008). Damit sowohl Laien als auch Experten in den angestrebten Aushandlungs- und Dokumentationsprozess, der vorwiegend auf lokalem Expertenwissen beruht, gleichermaßen einbezogen werden können, ist eine einfach zu bedienende Oberfläche, und eine gute Zugänglichkeit der Anwendung nötig. Um den letztgenannten Anspruch zu erfüllen, wurde das Konzept einer webbasierten Anwendung gewählt.

2 Die Anwendung „Schwachstellenerfassung im Langsamverkehr“

In diesem Kapitel wird hauptsächlich auf die technische Umsetzung der Anwendung zur Erfassung von Schwachstellen im Langsamverkehr eingegangen.

2.1 Gesamtkonzept und Softwarekomponenten

Im Zuge der wachsenden Popularität und Produktivität von Open-Source-Software (GOLDEN 2008) auch im GIS Bereich (DONNELLY et al. 2009) wurde das Projekt vollständig mit freier Software entwickelt. Lediglich zum Testen der Weboberfläche wurden neben den freien Webbrowsern Mozilla Firefox und Google Chrome die proprietären Webbrowser Microsoft Internet Explorer und Apple Safari verwendet.

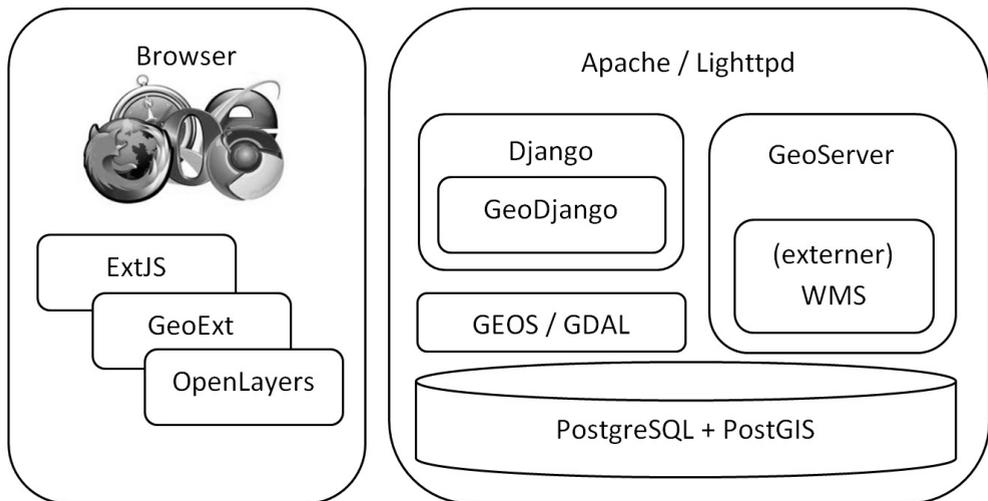


Abb. 1: Client- und serverseitig eingesetzte Softwarekomponenten

Serverseitig wird vorwiegend die Programmiersprache Python eingesetzt. Als Webframework kommt Django zum Einsatz. Die GIS-Erweiterung GeoDjango ermöglicht die Verwaltung von Geodaten und dadurch die Erfassung von Objekten direkt im Browser. Diese Funktion dient der Erhebung der Schwachstellen als Punkt- oder Linienobjekte. Neben der Programmlogik, die mit Django realisiert wird, werden die Daten in der Datenbank PostgreSQL gespeichert.

Clientseitig wird nur ein moderner Webbrowser zur Darstellung der Weboberfläche benötigt, der die Ausführung von JavaScript erlaubt. In der Weboberfläche kommen Techniken wie AJAX und JavaScript zum Einsatz, die eine dynamische Benutzerführung, Kartendarstellung und Datenverarbeitung ermöglichen. Diese Techniken werden u. a. durch die JavaScript-Bibliotheken ExtJS, GeoExt und OpenLayers zur Verfügung gestellt.

2.2 Django, GeoDjango und PostgreSQL/PostGIS

Aufgrund des umfangreichen, integrierten Funktionsumfang wie Benutzerverwaltung, Verwaltung von URLs, Zugriff auf zahlreiche Datenbankmanagementsysteme, objektorientierter Zugriff auf Datenbankeinträge, Administrationsoberfläche, Template-System, Sprachunterstützung und der GIS-Erweiterung GeoDjango, wurde Django als Webframework gewählt.

Django ist in der Programmiersprache Python geschrieben und agiert nach dem MVC-Prinzip (Model, View, Controller). So kann neben dem Datenbankmodell, die Programmlogik und Funktionalität innerhalb einer Entwicklungsumgebung implementiert werden, ohne dass zusätzliche Softwarepakete benötigt werden. Django bietet mit seinem Modul GeoDjango eine Schnittstelle zur Geospatial Data Abstraction Library (GDAL)- und Geometry Engine Open Source (GEOS)-API an und unterstützt mit der Einbindung von PostgreSQL-Datenbanken mit PostGIS-Erweiterung die räumlichen Aspekte der Anwendung.

Mit GeoDjango werden neben der Verwaltung der Geometriedaten in der Datenbank PostgreSQL auch geometrische Operationen durchgeführt. Konkret wird vor der Speicherung von Geometrien einer Schwachstelle überprüft, ob sich diese innerhalb der Gemeindegrenze befindet, für die der erfassende Benutzer eine Autorisierung hat.

Als Webserver können verschiedene Systeme eingesetzt werden. Als Beispiele sind hier Lighttpd mit FastCGI oder Apache mit mod_python oder mod_wsgi zu nennen. Im vorliegenden Fall wurde Apache mit mod_wsgi gewählt.

2.3 Weboberfläche mit ExtJS, GeoExt und OpenLayers

Zur Verwaltung des Webseitenlayouts wurde die umfangreiche JavaScript-Bibliothek ExtJS eingesetzt. ExtJS bietet die Unterstützung von sogenannten Widgets an, die die Funktionalität und das Aussehen von Komponenten einer graphischen Benutzeroberfläche (z. B. Buttons, Auswahllisten, Tabellen, Eingabeformulare, Toolbars, Menüs, usw.) innerhalb eines Webbrowsers simulieren (Abb. 2). Neben den Widgets liegt die Stärke von ExtJS in der Verwaltung von Daten (sogenannte Stores) und in der umfangreichen Unterstützung von AJAX (Kommunikation zwischen Webbrowser und Django).

Via der GeoExt-Erweiterung für ExtJS wird in der Anwendung das Kartenframework OpenLayers eingesetzt. Als Kartenlayer kommen sowohl die Geometriedaten der Schwachstellen in Form von Vektordaten als auch Hintergrunddaten wie Orthophotos und Straßenachsen als Rasterdaten zum Einsatz. Die Vektordaten werden mittels GeoDjango in das GeoJSON-Format konvertiert und die Rasterdaten werden als Kacheln zur Verfügung gestellt.

Aufgrund der Möglichkeit, mehrere Geometrien (Linien und Punkte) der gleichen Schwachstellen zuzuordnen, entstehen netzwerkartige Beziehungen. D. h. eine Schwachstelle kann mehrere, eine oder keine Geometrieobjekte besitzen. Diese Komplexität verlangt eine Erweiterung der OpenLayers-Funktionalität. So werden bei der Auswahl einer Geometrie oder einer Schwachstelle, alle zugehörigen Geometrien gesondert hervorgehoben.

3 Erkenntnisse

3.1 Programmversionen

Die Modellierung und Funktionalität der Webapplikation mit Django gelang dank der sehr guten und umfangreichen Dokumentation in kurzer Zeit. Veränderungen am Datenmodell haben aufgrund des objektorientierten Ansatzes nur wenige Änderungen in der Programmlogik zur Folge.

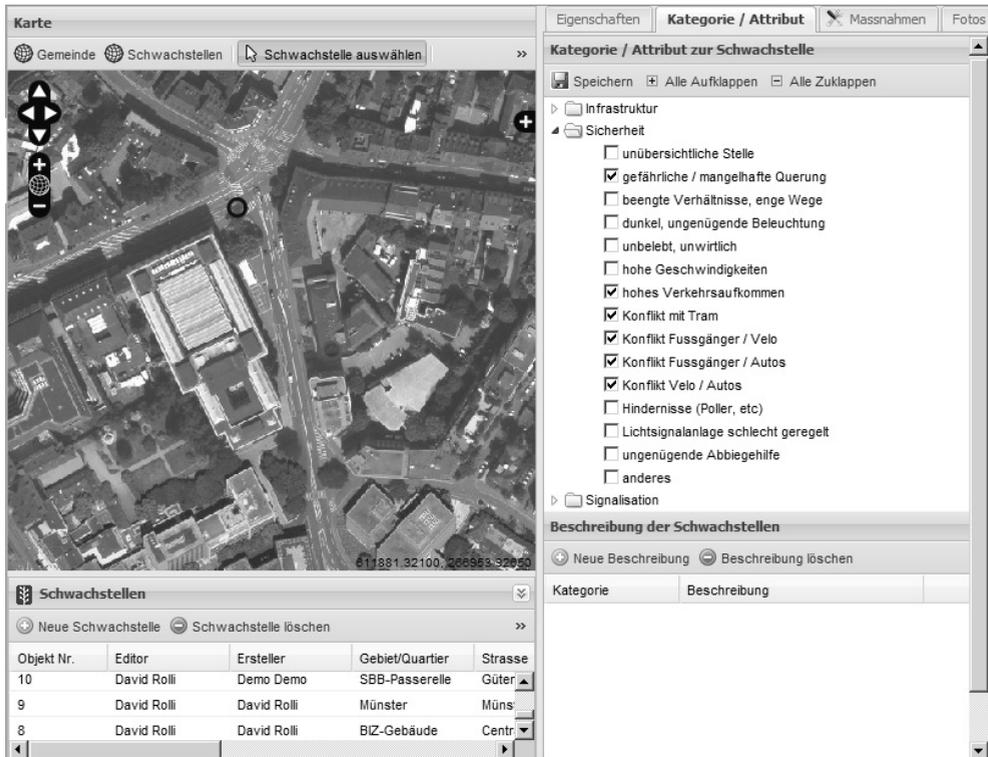


Abb. 2: Webseitenlayout mit ExtJS, GeoExt und OpenLayers

Open-Source-Projekte unterliegen oft aufgrund ihrer aktiven Online-Community starken Entwicklungsschritten und bringen daher mit neuen Softwareversionen neue Funktionalitäten. Daher gelten Entwicklungsversionen im Allgemeinen oft als nicht stabil und werden deshalb nicht auf Produktivsystemen eingesetzt. Allerdings konnte innerhalb der Entwicklungsphase kein Defizit in der Stabilität der Entwicklungsversion von Django entdeckt werden und wurde deshalb auf dem Produktivsystem eingesetzt.

Bei der Migration auf das Produktivsystem musste somit die Rückwärtskompatibilität berücksichtigt werden. Dies bedeutete unter anderem, den Quellcode auf ältere Python-Versionen anzupassen.

Des Weiteren wird für die Integration eines externen, sogenannten cascading WMS in eine bestehende GeoServer-Instanz eine Entwicklungsversion von GeoServer vorausgesetzt (GEOSERVER DOCUMENTATION, 2011). Diese Funktionalität ermöglicht das Exportieren der Vektordaten aus der Datenbank und den Rasterdaten eines externen WMS in ein PDF-Dokument.

3.2 Browserkompatibilität und OpenLayers

Die Nutzbarkeit der Anwendung sollte Browser-unabhängig sein. Diese Unabhängigkeit ist bei modernen Browser, die sich an die W3C-Spezifikationen halten, in der Regel gewähr-

leistet. Allerdings verwenden laut Aussage des Auftraggebers die meisten Nutzer der Anwendung den Internet Explorer von Microsoft, der oft zu Problemen mit der Ausführung von JavaScript Code führt. Dies hat eine fehlerhafte bis keine Darstellung der Web-Oberfläche für Nutzer des Internet Explorers zur Folge. Der Sicherstellung der Browserkompatibilität musste nebst der Implementierung der GUI-Funktionalitäten während der Entwicklungsphase der zweitgrößte Zeitaufwand zugeschrieben werden. Dies könnte durch eine Verbesserung des Programmkerns seitens Microsoft deutlich reduziert werden.

Trotz bestehender Funktionalität bezüglich Darstellung (Styling) von Geometrieobjekten (Features) in OpenLayers, ist die dynamische Veränderung der Darstellung während der Laufzeit wenig dokumentiert und undurchsichtig. Hier wäre es wünschenswert, wenn die Dokumentation erweitert würde.

Eine weitere Problematik mit OpenLayers bestand in der Verarbeitung von CRUD-Anfragen (Create, Update, Delete Funktionen bezüglich Features) mittels AJAX, die innerhalb einer internen Funktion von OpenLayers durchgeführt werden. Somit kann nur über Umwege auf Fehlermeldungen, wie beispielsweise wenn eine Geometrie außerhalb einer Gemeinde liegt, oder auf zurückkommende Daten (neue ID einer Geometrie) reagiert werden.

3.3 Exportfunktionalität

Der Export von Schwachstellen in das PDF-Format wird durch die Integration von Vektordaten und Rasterdaten problematisch, da diese kombiniert dargestellt und aus verschiedenen Datenquellen zusammengesetzt werden müssen. Das Projekt „Schwachstellen“ setzt jedoch den Export der kompletten Kartendarstellung (entsprechend der Darstellung in OpenLayers und in Abb. 2) voraus und somit die Kombination von Vektordaten und Rasterdaten. Weder OpenLayers noch GeoServer unterstützen zu diesem Zeitpunkt in ihren stabilen Versionen den Export dieser Kombination. Aus diesem Grund wurde eine Entwicklungsversion des GeoServers eingesetzt, die die Integration eines externen WMS erlaubt. Dadurch können mittels einer Layergruppe die Vektor- und WMS-Daten kombiniert mit einer GetMap-Anfrage zwischengespeichert und in einem PDF-Dokument dargestellt werden.

Für den Export als Microsoft Excel Datei wurde die Python-Bibliothek xlwt eingesetzt. Allerdings ist dort die Integration von Geodaten nicht vorgesehen, da es sich um ein tabellarisches Format handelt.

3.4 Nutzung der Anwendung

Da zum Zeitpunkt der Erstellung dieses Beitrags die Anwendung gerade erst freigeschaltet wird, können gegenwärtig noch keine Aussagen zur Nutzung und zur Zufriedenheit der Nutzer mit der Anwendung gemacht werden. Es wird sich daher zeigen, ob die Komplexität der Thematik und der Umsetzung so weit vereinfacht werden konnte, dass auch Laien sich leicht mit der Anwendung beschäftigen und korrekt Schwachstellen erfassen können.

4 Fazit und Ausblick

In Zukunft wäre die Entwicklung einer mobilen Oberfläche für Smartphones (z. B. Android oder iPhone) denkbar, um die Erfassung von Schwachstellen direkt vor Ort mit Photo und Koordinaten zu ermöglichen. Gegebenenfalls werden aber aus ersten Anwenderrückmeldungen weitere Anpassungen sowohl der Nutzeroberfläche als auch der Funktionalität an der Anwendung einfließen.

Literatur

- ASTRA (2007): Der Langsamverkehr in den Agglomerationsprogrammen. <http://www.astra.admin.ch/themen/langsamverkehr/00483/index.html> (01.02.2011).
- BAU- UND UMWELTSCHUTZDIREKTION KANTON BASEL-LANDSCHAFT (2010): Fuss- und Veloverkehr in der Agglomeration Basel. Wegleitung für die Gemeinde.
- DONNELLY, F. (2009): Evaluating open source GIS for libraries. http://gothos.info/resource_files/fpd_libhitech_foss_gis_march2010.pdf (01.02.2011).
- GEOSEVER DOCUMENTATION (2011): External Web Map Server – GeoServer 2.2-SNAPSHOT User Manual. <http://docs.geoserver.org/latest/en/user/data/wms.html> (01.02.2011).
- RAMASUBRAMANIAN, L. (2008): PPGIS as Critical Reflective Practice. *Geographic Information Science and Public Participation*. Springer, Berlin/Heidelberg, S. 135-149.
- TULLOCH, D. (2008): Public Participation GIS (PPGIS). In: KEMP, K. (Ed.): *Encyclopedia of GIS*. Sage Publications, S. 351-353.