

Entwicklung eines interaktiven Unfallinformationssystems mit SVG

Christian Spanring
Raumplanung und Raumordnung

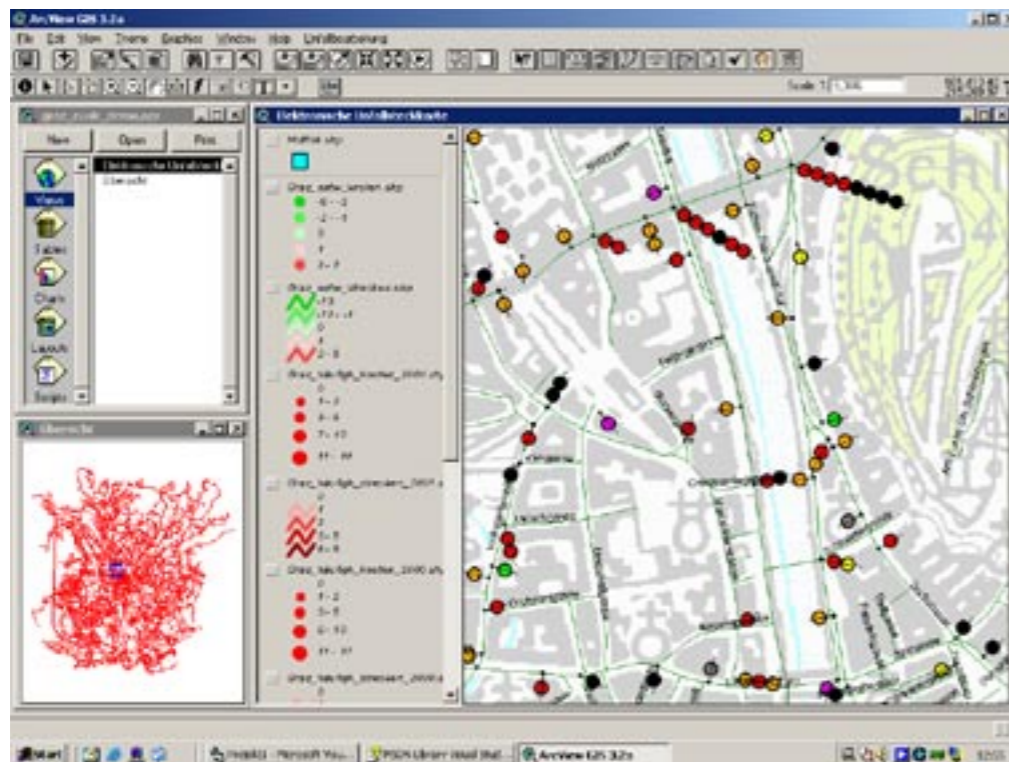
betreut durch:
Ao.Univ.Prof. Mag. Dr. Georg Gartner
Institut für Kartographie und
Geo-Medientechnik
Technische Universität Wien

Entwicklung eines interaktiven Unfallinformationssystems mit SVG

Unfallforschung seit 1964

Unfallsteckkarte - 'Papier und Stecknadeln'

Auf Kartenwerken wurden die Verkehrsunfälle mit Stecknadeln markiert um sich einen besseren Überblick zu verschaffen und sogenannte Unfallhäufungsstellen ausfindig zu machen.



Elektronische Unfallsteckkarte ArcView GIS > Webapplikation

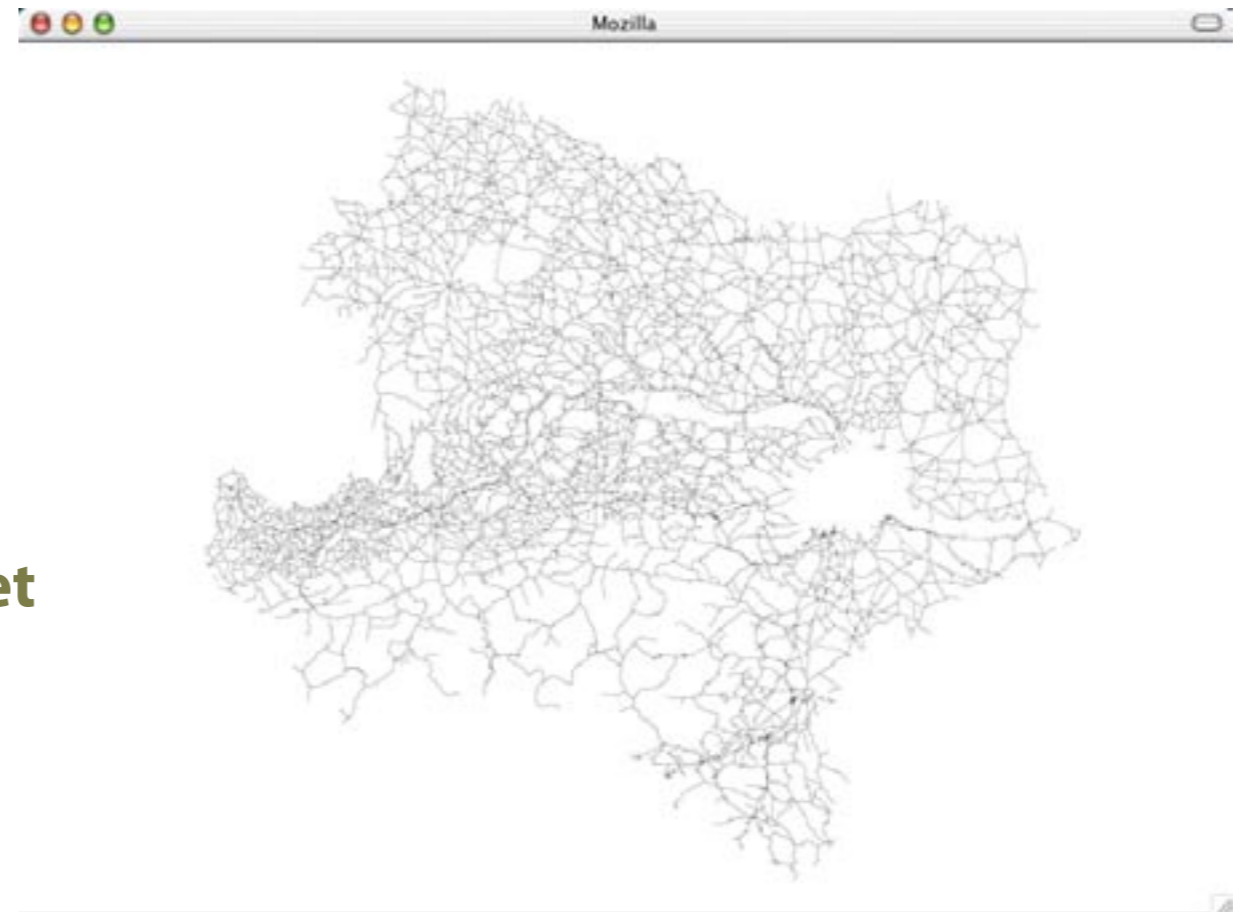
Das aktuelle Unfallinformationssystem bestehend aus der Unfalldatenbank und der Elektronischen Unfallsteckkarte (GIS) ist ohne spezielle Vorkenntnisse (Expertenwissen) kaum bedienbar. Diese Diplomarbeit soll versuchen die beiden Systeme miteinander zu verknüpfen und sie ohne Expertenwissen bzw. spezielle technische Voraussetzungen zugänglich zu machen.

Entwicklung eines interaktiven Unfallinformationssystems mit SVG

Konvertierung GIS > SVG

Problem:
SVG-Dateigröße

Auswirkungen:
> Datenübertragung im Internet
> SVG Viewer Performance



Eine erste Konvertierung der ArcView GIS Daten nach SVG war mit Hilfe des Avenue-Scripts 'shp2svg' (<http://www.carto.net>) sehr einfach möglich. Damit wurde eine sehr saubere und leicht editierbare SVG-Struktur erzeugt. Das Problem bestand nun darin, dass der erste und wichtigste Layer, der 'Strassengraph' bereits 3,5mb Speicherplatz beanspruchte.

Mit dieser Dateigröße müßte man eine relativ große (trotz immer höher werdenden Bandbreiten im Internet) Wartezeit in Kauf nehmen, bis sich die Karte vollständig aufgebaut hat. Auch hatte der Adobe SVG Viewer selbst bei lokalem Aufruf der Daten Probleme, diese noch fehlerfrei darzustellen und die verfügbaren Funktionen wie 'Zoom' und 'Pan' in einer erträglichen Geschwindigkeit (flüssig) auszuführen.

Entwicklung eines interaktiven Unfallinformationssystems mit SVG

Konvertierung GIS > SVG

Lösung:

'load on demand'

Unter 'load on demand' wird nur jene Datenmenge laden die auch benötigt wird verstanden. In diesem konkreten Fall heißt das, nur jene Elemente der Karte in den Viewer zu laden, die in diesem Moment sichtbar sind ('viewbox'). Bei Änderung des Kartenausschnitts wird der Vorgang wiederholt und neue sichtbare Daten geladen.

'zoom to full extent' unterbinden

Die Performance-Probleme des Adobe SVG Viewer bei Darstellung der vollen Datenmenge kann man auch mit Hilfe des 'load on demand' Schemas nicht umgehen. Daher besteht die einzige Möglichkeit darin dieses Problem zu lösen, nicht die volle Datenmenge anzeigen zu lassen.

in Betracht gezogen:

> Oracle Spatial / XSQL

Über XSQL ('XML SQL' von Oracle) werden geometrische Daten aus der Oracle Spatial Datenbank direkt als XML zurückgegeben und via XSLT (Extensible Stylesheet Language Translator) in SVG übersetzt.

Ein Beispiel dazu ist unter <http://webmap.geog.byu.edu/webmap/atlas.svg> zu finden.

Diese Lösung wurde aus Kostengründen und aufgrund der proprietären Sprach XSQL nicht weiterverfolgt. Ein wesentliches Ziel dieser Arbeit stellt die Offenheit des Systems dar.

> PostgreSQL / PostGIS

Für den OpenSource-Datenbankserver PostgreSQL gibt es das sogenannte PostGIS-Zusatzmodul. Diese Modul dient der Handhabung geometrischer Information, ermöglicht beispielsweise den Import von Shape-Files und geometrische Berechnungen (Länge, Fläche, find within, etc.).

Diese Variante wurde wieder aufgegeben, da der geschätzte benötigte Zeitaufwand zur Aneignung der benötigten Unix-Kenntnisse um den Datenbankserver sowie das PostGIS-Modul zu administrieren und in weiterer Folge die Anlernung der PostgreSQL/PostGIS-Syntax als zu hoch erschien.

Ein Beispiel dazu ist unter <http://tirolatlas.uibk.ac.at/papers/svgopen2002/isogen/index.html> zu finden.

Entwicklung eines interaktiven Unfallinformationssystems mit SVG

Konvertierung GIS > SVG

in Betracht gezogen:

> MySQL

Aufgrund bereits vorhandener, relativ guter Vorkenntnisse in MySQL, einem installierten und funktionierenden MySQL-Datenbankserver und der Abschätzung, dass die benötigten Funktionen zur dynamischen Erzeugung der SVG-Daten, die die oben genannten Systeme teilweise onboard liefern, auch unter nicht allzu großem Entwicklungsaufwand selbst erzeugt werden können, fiel die Entscheidung letztendlich auf MySQL.

weiteres Vorgehen:

> MySQL-Datenbank des Strassengraphen

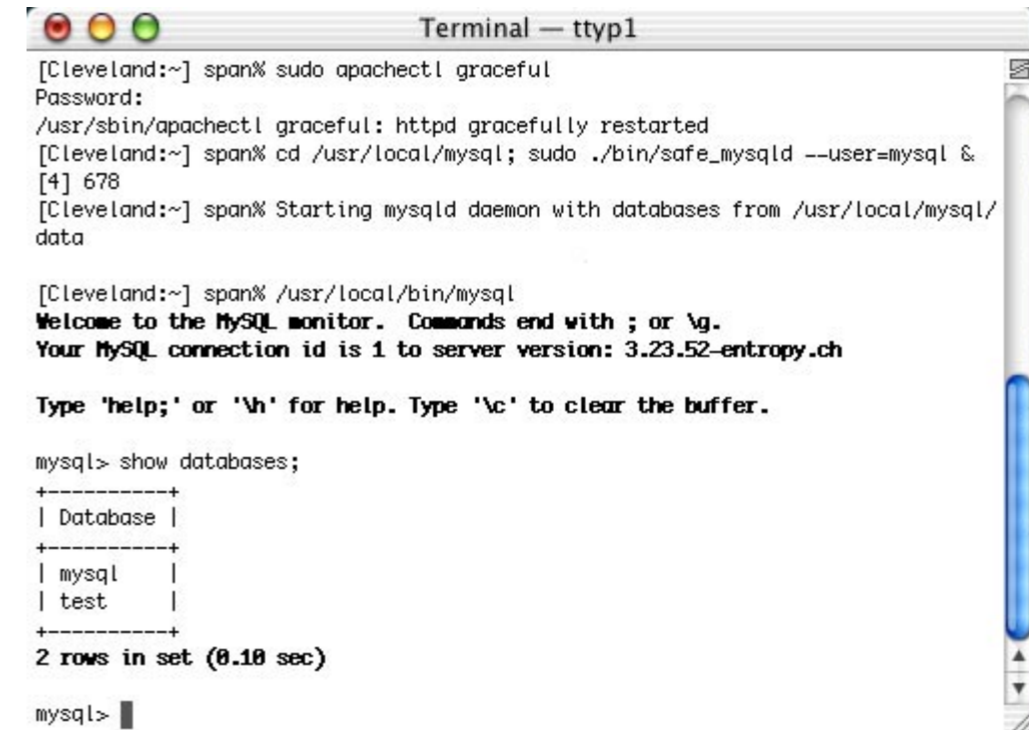
Exportieren der geometrischen Informationen aus ArcView GIS mittels eines Avenue-Scripts und in den MySQL-Server übertragen. Prinzipiell ist dieser Vorgang auf 2 Varianten lösbar:

a) mittels einer 'Zwischendatei', also ein in MySQL importierbares Format (z.B. Text- oder Komma-getrennter Text, MySQL-Dumpfile)

b) direkte Herstellung einer Verbindung von ArcView GIS zum MySQL-Server und direktes Schreiben der Datenbankeinträge (ähnlich dem OpenSVGServer auf <http://www.carto.net>)

> dynamische Erzeugung des Strassengraphen mit PHP/JavaScript/SVG

...aus der MySQL-Datenbank nach dem 'load on demand' Schema.



```
Terminal — tty1
[Cleveland:~] span% sudo apachectl graceful
Password:
/usr/sbin/apachectl graceful: httpd gracefully restarted
[Cleveland:~] span% cd /usr/local/mysql; sudo ./bin/safe_mysqld --user=mysql &
[4] 678
[Cleveland:~] span% Starting mysqld daemon with databases from /usr/local/mysql/
data

[Cleveland:~] span% /usr/local/bin/mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1 to server version: 3.23.52-entropy.ch

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> show databases;
+-----+
| Database |
+-----+
| mysql    |
| test     |
+-----+
2 rows in set (0.10 sec)

mysql>
```