

ENTWICKLUNG EINES GEOMETRIE-EDITORS FÜR DAS WEBGIS KVWMAP AUF BASIS VON SVG, POSTGIS UND AJAX

Stefan Rahn

Steinbeis Transferzentrum Geoinformatik Rostock

Professur für Geodäsie und Geoinformatik,
Universität Rostock

Kurzfassung: Das Internet-GIS „kvwmap“ wird als Open-Source-Software-Projekt am Steinbeis Transferzentrum Geoinformatik in Rostock entwickelt. Das System wird in mehreren Kreis- und Amtsverwaltungen in Mecklenburg-Vorpommern eingesetzt. Neben der Nutzung als Auskunftssystem für Geobasisdaten wird kvwmap auch für die Erfassung und Bearbeitung von Fachdaten eingesetzt. Durch ein generisches Layerkonzept können Sach- und Geometriedaten beliebiger Layer editiert werden. Für die Bearbeitung der Geometriedaten wurde ein eigener Geometrieeditor entwickelt. Der Editor bietet mehrere Werkzeuge, um Flächen-, Linien- und Punktgeometrien zu bearbeiten. Dazu gehören beispielsweise die Freihanddigitalisierung, geometrische Operationen (Differenz, Vereinigung) auf Basis eigener Polygone oder Geometrien anderer Vektorlayer und die Bearbeitung von Eckpunkten. Umgesetzt wurde der Geometrie-Editor mit SVG, Javascript, PostGIS und PHP.

1 Motivation

Internet-GIS-Lösungen bieten gegenüber lokalen Desktop-GIS sehr viele Vorteile. Durch die redundanzfreie, zentrale Datenhaltung ist sichergestellt, dass jeder Nutzer auf den aktuellsten Daten arbeiten kann. Die Client-Server-Architektur des Internet-GIS hat außerdem den Vorteil, dass der Pflege- und Verwaltungsaufwand ausgelagert wird. Der gesamte Administrationsaufwand konzentriert sich auf einen Server und nicht auf hunderte von Arbeitsplätzen [1].

Viele öffentliche Verwaltungen setzen bereits Internet-GIS-Technologie ein und können ihre Geodaten so einem großen Nutzerkreis individuell zur Verfügung stellen. Durch die fortschreitenden Entwicklungen in der Web-Technologie verfügen diese Systeme über immer mehr Funktionalität und beginnen zunehmend Desktop-GIS zu ersetzen.

Neben der Nutzung als reine Auskunftssysteme von Sach- und Geodaten werden Web-GIS in den Kreis- und Amtsverwaltungen auch zunehmend für die Datenerfassung, Datenbearbeitung und Datenverwaltung eingesetzt. Viele Systeme besitzen Fachschalen [2] für die Erfassung und Bearbeitung von speziellen Fachdaten oder bieten über einen generischen Ansatz die Möglichkeit beliebige Layer zu editieren [3].

Um die oben genannten Prozesse in einem Web-GIS umsetzen zu können, ist es erforderlich, sowohl die Sach- als auch die Geometriedaten von Layern bearbeiten zu können. Eine besondere Herausforderung dabei ist die Entwicklung eines leistungsfähigen Geometrie-Editors. Mit dem Editor muss es möglich sein, Geometrien verschiedener Typen (Fläche, Linie, Punkt) zu erzeugen und vorhandene Datensätze zu bearbeiten. Die meisten webbasierten Geometrie-Editoren bieten jedoch nur die Möglichkeit der einfachen Modifizierung der Geometriedaten auf Eckpunkt-Basis (siehe Abschnitt 3).

Bei der Digitalisierung ist es aber oft erforderlich, geometrische Operationen auf den Geometrien auszuführen, um Geometrien zu vereinigen oder von einander abzuziehen oder die Erfassung auf Basis vorhandener Geometriedaten durchzuführen. Um diese, aus Desktop-GIS bekannten Funktionalitäten auch webbasiert nutzen zu können, wurde der Geometrie-Editor für das Web-GIS kvwmap entwickelt, welcher in diesem Beitrag näher beschrieben werden soll.

2 Grundlagen

Im Folgenden sollen die drei Basistechnologien, auf denen der Geometrie-Editor aufsetzt sowie das WebGIS kvwmap näher beschrieben werden. Für die grafische Darstellung wird der Vektorgrafik-Standard SVG verwendet. Das Datenbanksystem PostGIS dient als Datenhaltungskomponente und Geometrieprozessierungsmodul. Die clientseitige Interaktion und Kommunikation mit der Geometriedatenbank wird mit AJAX realisiert.

2.1 SVG

SVG (Scalable Vector Graphics) [4] ist eine XML-basierte Sprache für die Beschreibung von 2D-Grafik. Mit SVG lassen sich 3 verschiedene Arten von grafischen Elementen darstellen: Vektorgrafikobjekte (Kreise, Rechtecke, Polygone, Linienzüge, etc.), Rastergrafiken und Text. Die Eigenschaften der grafischen Elemente lassen sich durch Objektattribute und Cascading Style Sheets (CSS) [5] beschreiben. SVG unterstützt außerdem die Gruppierung, Transformation und Animation von Objekten.

Eine Besonderheit von SVG ist die Möglichkeit, alle Elemente des Document Object Model (DOM) [6] mit Hilfe von Skriptsprachen (wie ECMAScript [7]) zu verändern. Dadurch werden SVG-Grafiken dynamisch und interaktiv.

SVG ist ein W3C-Standard, dessen erste Version (SVG 1.0) im September 2001 veröffentlicht wurde. Die aktuelle Spezifikation (SVG 1.1) wird seit Januar 2003 vom W3C

empfohlen. Momentan arbeitet die W3C SVG Working Group an der SVG-Version 1.2, welche als Entwurf zur Verfügung steht.

2.2 PostGIS

PostGIS [8] ist eine räumliche Erweiterung für das objektrelationale Datenbankmanagementsystem PostgreSQL [9]. Durch die Erweiterung erhält das System GIS-Funktionalitäten nach der „OpenGIS Simple Features Specification For SQL“ [10]. In einer PostGIS-Datenbank ist z.B. es möglich, geometrische Objekte zu speichern. PostGIS unterstützt die folgenden Geometrietypen:

- OpenGIS Well-Known Text (WKT) bzw. Well-Known Binary (WKB): Point, Linestring, Polygon, Multipoint, Multilinestring, Multipolygon und Geometrycollection
- Extended Well-Known Text bzw. Extended Well-Known Binary: OpenGIS WKT mit Höheninformation
- SQL Multimedia Applications Spatial Specification (nicht vollständig unterstützt): Circularstring, Compoundcurve, Curvepolygon, Multicurve, Multisurface

Außerdem unterstützt PostGIS eine Vielzahl von Funktionen und Operatoren zur Analyse und Verarbeitung von geometrischen Objekten:

- Geometriebeziehungs-Funktionen (z.B. Overlaps, Intersects, Within)
- Geometrieprozessierungs-Funktionen (z.B. Flächenberechnung, Verschneidung)
- Funktionen zur Abfrage von Geometrien (z.B. als WKT, GML, KML, SVG)
- Funktionen zur Geometrie-Erzeugung

PostGIS wurde von der kanadischen Firma Refrations Research entwickelt und unterliegt der GNU General Public License.

2.3 AJAX

AJAX (Asynchronous JavaScript and XML) [11] ist ein Oberbegriff für den Einsatz verschiedener Technologien, die es ermöglichen, eine asynchrone Datenübertragung zwischen einem Server und dem Browser herzustellen, um innerhalb einer HTML-Seite eine HTTP-Anfrage durchzuführen, ohne die Seite komplett neu laden zu müssen. In einer AJAX-Anwendung werden folgende Web-Technologien benutzt.

- XHTML und CSS für die Darstellung der Webinhalte
- das Document Object Model (DOM) für die Repräsentation der Inhalte
- XML für den Austausch von Daten
- das XMLHttpRequest-Objekt für die asynchrone Kommunikation mit dem Webserver
- JavaScript, um das DOM manipulieren zu können und als Schnittstelle zwischen den Komponenten.

Die AJAX-Technologie wird in vielen Web 2.0-Anwendungen eingesetzt, um ein interaktives, an Desktop-Anwendungen angelehntes Verhalten zu realisieren. Das Prinzip besteht darin, bei Nutzeraktionen, die einen Zugriff auf neue Informationen darstellen, mit dem Javascript-XMLHttpRequest-Objekt eine HTTP-Anfrage an den Webserver zu senden. Der Webserver bearbeitet die Anfrage und schickt als Antwort die angeforderten Daten als XML an den Client. Der Client verwendet die Daten, um den aktuellen DOM zu erweitern bzw. zu verändern und kann ihn sofort darstellen. Da diese Prozesse asynchron ausgeführt werden, können auch mehrere XMLHttpRequests nacheinander abgesetzt werden. Die Anwendung muss nicht auf das Ergebnis der Anfragen warten.

2.4 Das WebGIS kvwmap

Das Web-GIS „kvwmap“ [12] ist ein Softwaresystem zur Erfassung, Verarbeitung, Analyse und Präsentation von raumbezogenen Informationen. Das System wird als Open-Source-Projekt am Steinbeis Transferzentrum Geoinformatik in Rostock entwickelt. Eine Zielstellung des Projektes ist die Entwicklung eines WebGIS, das den speziellen Aufgaben der Kreis-, Gemeinde- und Amtsverwaltungen gerecht wird. Kvwmap zeichnet sich durch Fachschalen aus, deren Funktionen auf diese Aufgaben zugeschnitten sind [1].

Kern des Systems ist der UMN-MapServer [13], eines der am meisten verwendeten freien Softwaresysteme zur Bereitstellung von Karten über das Internet. Kvwmap ist in PHP [14] implementiert und verwendet PHP MapScript [15], um auf die API des UMN-MapServers zuzugreifen. Ein Apache-Webserver [16] übernimmt die Ausführung des PHP-Codes und die Kommunikation mit dem Client. Kvwmap benutzt eine PostgreSQL-Datenbank mit PostGIS-Aufsatz, um die räumlichen Daten und zugehörigen Sachdaten zu speichern. Um die Anwendung zu benutzen, wird ein herkömmlicher Browser verwendet. Für die clientseitige Interaktion, insbesondere bei der Navigation auf der Karte und der Benutzung des Geometrieeditors, wird SVG in Kombination mit Javascript/AJAX eingesetzt.

Das System wird derzeit in 10 Kreis- und Amtsverwaltungen in Mecklenburg-Vorpommern eingesetzt. Neben der Nutzung als Auskunftssystem für Geobasisdaten wird kvwmap auch zunehmend für die Erfassung und Bearbeitung von Fachdaten verwendet. Dies führte u.a. zur Entwicklung des Geometrie-Editors.

3 Verwandte Arbeiten

Die Webmapping-Anwendung von Neumann und Winter [17] nutzt die gleichen Basis-komponenten wie der hier vorgestellte Geometrie-Editor. SVG dient als Grafik-Ausgabe-Schnittstelle zur Darstellung der Geodaten, die in einer PostGIS-Datenbank gespeichert sind. Bei jeder Navigationsoperation (Zoom, Pan) wird eine Javascript-Funktion ausgeführt, welche die darzustellenden Geometriedaten jedes aktiven Layers per XMLHttpRequest vom Server anfordert. Sobald die angeforderten Geometriedaten im Client vorliegen, werden sie in den DOM eingefügt und dargestellt.

Obwohl diese Anwendung, wie schon erwähnt, die gleichen Techniken verwendet wie das in diesem Beitrag vorgestellte System, ist die Funktionalität eine andere. Die Anwendung ist eine Webmapping-Applikation und dient nur zur reinen Betrachtung der Geodaten. Obwohl die Geometriedaten in Vektorform auf dem Client vorliegen, gibt es keine Möglichkeit zur Editierung.

In einer weiteren Veröffentlichung [18] beschreibt Neumann ein Konzept zur Digitalisierung und Editierung von Geodaten. Auch hier wird eine PostGIS-Datenbank als Datenhaltungskomponente und SVG als Grafikausgabe-Standard verwendet. Das Nachladen der Geodaten erfolgt hier ähnlich wie in [17]. Eine ECMAScript-Funktion stellt bei einer Navigationsaktion eine Anfrage an ein serverseitiges PHP-Skript und übermittelt den neuen Kartenausschnitt und die aktiven Layer. Auf dem Server werden die neuen Geometrie-Daten aus der PostGIS-Datenbank extrahiert und an den Client gesendet. Hier werden die Daten von einem Parser interpretiert und in den DOM eingebunden. Um Geometriedaten zu digitalisieren und zu bearbeiten, stehen mehrere Funktionen zur Verfügung:

- Digitalisierung (Hinzufügen von Eckpunkten)
- Verschieben von Eckpunkten
- Löschen von Eckpunkten
- Löschen von Teilgeometrien

Allerdings beschränken sich alle diese Editier-Funktionen auf das Hinzufügen, Entfernen oder Modifizieren von Koordinaten in den Geometriedaten, die sich auf dem Client befinden. Es können keine geometrischen Operationen wie Vereinigung oder Differenz auf den Geometrien ausgeführt werden. Ebenso ist es nicht möglich, Geometrien aus anderen Layern, die sich nicht auf dem Client befinden, in die Digitalisierung mit einzubeziehen.

4 Umsetzung

4.1 Voraussetzungen

Wie schon erwähnt, unterstützt der Editor die Geometrietypen Fläche, Linie und Punkt. Da die Daten in einer PostGIS-Datenbank gespeichert werden, muss der Geometrietyp der entsprechenden Tabelle „MULTIPOLYGON“, „MULTILINESTRING“ oder „POINT“ sein. Je nach Geometrietyp hat der Editor andere Werkzeuge mit unterschiedlichen Funktionen. Welche Werkzeuge geladen werden wird entschieden, wenn ein Datensatz eines PostGIS-Layers abgefragt wird bzw. ein neuer Datensatz erfasst werden soll. Außerdem muss die entsprechende Tabelle oids [19] besitzen, um die einzelnen Datensätze bei Abfrage oder Änderung der Geometriedaten eindeutig identifizieren zu können.

4.2 Laden und Darstellen der Geometriedaten auf dem Client

Bei Bearbeitung der Geometrie eines bestehenden Datensatzes werden die PostGIS-Funktionen „AsSvg()“ und „AsText()“ verwendet, um die Geometriedaten sowohl als SVG als auch in WKT-Form auf dem Client zu laden. Diese Zweiteilung der geometrischen Repräsentation wird deswegen vorgenommen, damit die Geometriedaten einerseits sofort mit SVG dargestellt werden können und andererseits in der für weitere Geometrieoperationen geeigneten WKT-Form vorliegen.

Für die SVG-Darstellung von Polygonen und Linienzügen werden Path-Objekte verwendet. Obwohl bei der Bearbeitung der Geometrien beliebig viele Ringe bzw. Teilgeometrien hinzugefügt werden können (siehe Abschnitt 4.4), werden nur zwei Path-Objekte benötigt, um die Geometrien zu repräsentieren. Das Prinzip besteht darin, die aktuelle Geometrie durch das eine Path-Objekt (path1) darzustellen und die Geometrie, die hinzugefügt werden soll, durch das andere Path-Objekt (path2).

4.3 Digitalisierung und Bearbeitung von Eckpunkten

Ein Basiswerkzeug für die Neuerfassung oder das Hinzufügen von Teilgeometrien ist die Freihanddigitalisierung. Mit diesem Werkzeug lassen sich sowohl Polygone als auch Linienzüge erzeugen, indem die Eckpunkte mit der Maus in der Karte gesetzt werden.

Um bereits vorhandene Eckpunkte von Polygonen oder Linienzügen zu bearbeiten, gibt es ein weiteres Werkzeug. Ist es aktiviert, werden alle Eckpunkte der aktuellen Geometrie hervorgehoben. Jeder der Eckpunkte kann nun verschoben oder gelöscht werden. Außerdem ist es möglich, neue Eckpunkte auf den Kanten der Geometrie bzw. am Anfang und Ende von Linienzügen zu erzeugen.

4.4 Hinzufügen und Löschen von Polygonen und Linienzügen

Die bisher vorgestellten Werkzeuge basieren auf der einfachen Veränderung von Eckpunkt-Koordinaten der auf dem Client geladenen Geometriedaten. Oft ist es bei der Geometriebearbeitung aber auch notwendig, Flächen zu vereinigen, mehrere disjunkte Polygone oder Löcher in Polygonen zu erzeugen.

Um dies zu realisieren, werden geometrische Operationen wie Differenz und Vereinigung benötigt, die die korrekte Ergebnisgeometrie auf dem Client berechnen. Eine Möglichkeit zur Umsetzung wäre, diese Operationen in Javascript neu zu implementieren. Dies hätte den Vorteil, dass die Berechnung direkt auf dem Client ausgeführt werden würde und keine Datenübertragung mit einem Server erfolgen müsste. Durch die Komplexität der Operationen ist eine komplette Neu-Implementierung jedoch sehr aufwendig und auf Grund der eingeschränkten Möglichkeiten von Javascript als clientseitige Scriptsprache auch nicht unbedingt geeignet.

In diesem Beitrag wird ein anderer Ansatz vorgestellt. Wie in Abschnitt 2.2 beschrieben, bietet die PostGIS-Datenbank bereits einen umfangreichen Satz an geometrischen

Funktionen. U.a. werden die kompletten OpenGIS-Funktionen unterstützt. Das Prinzip besteht darin, die geometrischen Operationen nicht direkt im Client auszuführen, sondern die Berechnung auf den PostGIS-Datenbankserver auszulagern.

Bei der Bearbeitung von Polygonen oder Linienzügen gibt es ein Werkzeug, mit dem neue Polygone bzw. Linienzüge zur aktuellen Geometrie hinzugefügt werden können. Während der Digitalisierung der hinzuzufügenden neuen Geometrie wird bei jedem gesetzten Eckpunkt ein XMLHttpRequest mit dem Geometrie-Operator der bereits vorhandenen Geometrie (path1) und der neuen Geometrie (path2) - jeweils in WKT-Form - an den Server geschickt. Auf dem Server wird die entsprechende Operation „Union()“ auf den beiden Geometrien durchgeführt und die Ergebnisgeometrie zurück an den Client gesandt. Sobald die Geometrie auf dem Client vorliegt, wird die aktuelle Geometrie (path1) durch diese ersetzt. Dieser Vorgang kann sich beliebig oft wiederholen. Analog erfolgt die Subtraktion von Polygonen, wobei hier die Funktion „Difference“ verwendet wird.

In Abb. 1 sind die einzelnen Schritte beim Vereinigen von zwei Polygonen dargestellt. Abb. 1a zeigt das Ausgangspolygon path1 (schraffiert). In Abb. 1b wird mit der Digitalisierung des zweiten Polygons path2 begonnen. Das Polygon path2 besitzt in Abb. 1c drei Eckpunkte, ist damit ein gültiges Polygon und die Vereinigungsoperation wird per XMLHttpRequest ausgelöst. Sobald das Ergebnis der Operation vorliegt, wird die Geometrie von path1 durch die Ergebnisgeometrie ersetzt (Abb. 1d). In Abb. 1e – h wird die Digitalisierung des Polygons path2 fortgesetzt und weitere Vereinigungsoperationen durchgeführt. Das Endergebnis ist in Abb. 1i dargestellt.

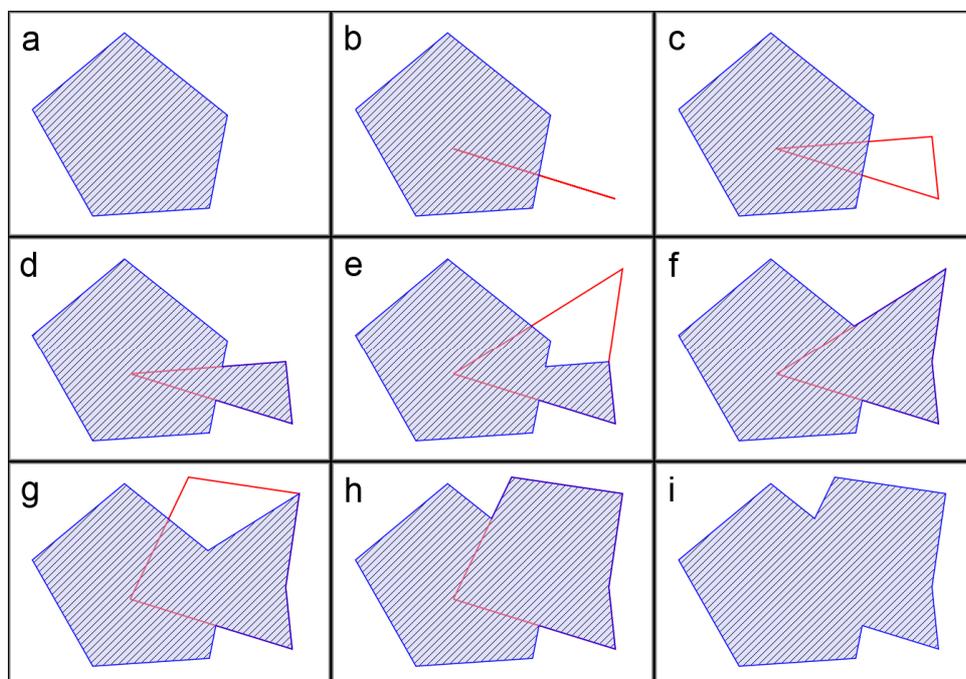


Abb. 1: Darstellung der einzelnen Schritte beim Vereinigen von 2 Polygonen

Durch die Verwendung von XMLHttpRequests läuft die Kommunikation mit dem Server für den Benutzer des Geometrie-Editors im Hintergrund ab. Außerdem erfolgt die Datenübertragung asynchron, so dass der Client nicht auf die Antwort des Servers warten muss, wodurch sich keine Verzögerung beim Zeichnen ergibt.

4.5 Geometrische Operationen auf Basis anderer PostGIS-Layer

Bei der Geometrie-Erfassung von Fachdaten ist es oft erforderlich, die Digitalisierung auf Basis bereits vorhandener Geometriedaten aus anderen Layern durchzuführen. Dies ist beispielsweise der Fall, wenn die Geometrien flurstücksscharf erfasst werden sollen. Auch diese Funktionalität lässt sich mit dem oben beschriebenen Prinzip umsetzen.

Um die Geometrie eines anderen Layers zur aktuellen Geometrie hinzuzufügen, muss dieser Layer zunächst aus einer Liste ausgewählt werden. Voraussetzung ist, dass der zweite Layer ein PostGIS-Layer ist und den gleichen Geometriotyp besitzt, wie der gerade bearbeitete Layer. Jetzt ist es möglich, mit einem Klick in die Karte oder durch Aufziehen eines Rechtecks ein oder mehrere Datensätze des zweiten Layers abzufragen. Auch hier wird wieder ein XMLHttpRequest verwendet, der die aktuelle Geometrie (path1), die Abfragekoordinaten und die Abfrage-Parameter des zweiten Layers zum Server sendet. Auf dem Server wird die Geometrie des Layers in dem angegebenen Bereich abgefragt und danach mit der aktuellen Geometrie vereinigt. Die Ergebnisgeometrie wird zurückgeschickt, path1 aktualisiert und dargestellt. Analog kann Geometrie von anderen Layern auch von der aktuell bearbeiteten Geometrie abgezogen werden.

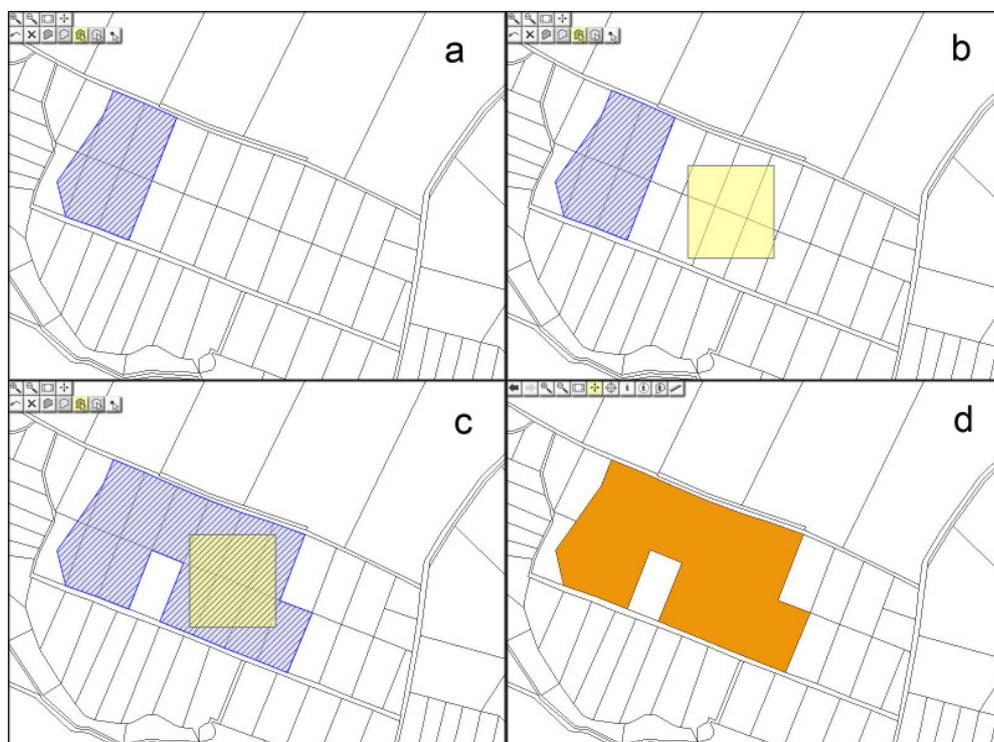


Abb. 2: Hinzufügen von Geometrie eines anderen Layers

In Abb. 2 ist die Digitalisierung eines neuen Polygonlayer-Datensatzes auf Basis eines Flurstückslayers dargestellt. Der Flurstückslayer ist aktiviert und in der Karte sichtbar. Die schraffierte Fläche in Abb. 2a ist die bereits digitalisierte Geometrie. Jetzt soll diese Geometrie auf Basis der Flurstücke erweitert werden. Dazu wird ein Abfragerechteck über den gewünschten Flurstücken aufgezogen (Abb. 2b) und der XMLHttpRequest mit der aktuellen Geometrie und den Abfrageparametern abgesetzt. Liegt das Ergebnis der Anfrage auf dem Client vor, wird die aktuelle Geometrie durch die neu berechnete ersetzt und dargestellt (Abb. 2c). Ist die gewünschte Fläche erfasst worden, kann der neue Datensatz gespeichert werden. Die auf Basis der Flurstücksgeometrie erzeugten Bereiche des Datensatzes stimmen exakt mit den Flurstücken überein (Abb. 2d).

5 Zusammenfassung und Ausblick

In diesem Beitrag wurde die Entwicklung eines Geometrie-Editors für das WebGIS kvwmap vorgestellt. Mit dem Editor ist es möglich, Punkt-, Linien- und Flächengeometrien von PostGIS-Layern zu bearbeiten und neu zu erfassen. Im Gegensatz zu anderen Arbeiten aus dem Bereich der webbasierten Geometriebearbeitung ist es mit dem hier vorgestellten System möglich, geometrische Operationen auf den Geometrien auszuführen. So lassen sich z.B. Flächen vereinigen oder von einander abziehen. Die Ausführung der Operationen findet dabei auf dem PostGIS-Datenbankserver statt, welcher über XMLHttpRequests angesprochen wird. Durch die Möglichkeit, während der Geometriebearbeitung im Hintergrund mit dem Server zu kommunizieren, lassen sich die geometrischen Operationen auch auf Basis anderer Layer durchführen, deren Geometriedaten sich nicht auf dem Client befinden.

Der entwickelte Geometrieeditor hat sich durch den Einsatz in den Kreisverwaltungen bereits in der Praxis bewährt. Die verschiedenen Fachämter einer Kreisverwaltung können ihre eigenen Daten selbständig und direkt im WebGIS erfassen und pflegen. Das hat zur Folge, dass für die Erfassung und Präsentation keine zusätzlichen Desktop-GIS-Arbeitsplätze mehr erforderlich sind. Außerdem werden die Änderungen für alle anderen Nutzer sofort sichtbar.

Trotz der guten Ergebnisse lässt sich der Geometrie-Editor in einigen Bereichen erweitern und verbessern. Eine sehr einfach umzusetzende Erweiterung ist die Unterstützung von weiteren PostGIS-Funktionen. Im Moment werden nur die Geometrieoperationen Union() und Difference() und für die Flächenberechnung Area() verwendet. Prinzipiell lassen sich aber alle von PostGIS unterstützten Funktionen in den Editor einbinden.

Zusätzliche Erweiterungsmöglichkeiten ergeben sich durch die Unterstützung von OGC-konformen Web-Diensten. So könnte z.B. die Digitalisierung auf Basis anderer Layer um eine Unterstützung des Web Feature Service (WFS) [20] erweitert werden. Dadurch ist es dann möglich, neben den eigenen PostGIS-Layern auch fremde WFS-Layer abzufragen und für die geometrischen Operationen zu verwenden.

Auch die Abfrage und Speicherung der zu bearbeitenden Layer ließe sich um eine WFS-Unterstützung erweitern. Hier müsste ein transaktionaler Web Feature Service (WFS-T) implementiert werden, um die Geometriedaten von WFS-Layern OGC-konform lesen und schreiben zu können.

Die Schnittstelle zwischen prozessierendem Server und Client könnte durch Umsetzung in einem Web Processing Service (WPS) [21] ebenfalls Standard-konform erfolgen. Allerdings wäre dies nur sinnvoll, wenn entweder die Zeichenfunktionen des Clients und/oder die Geometrieberechnungsfunktionen des Servers interoperabel genutzt werden sollen.

Das größte Problem, welches sich beim Einsatz des Geometrie-Editors zeigte, ist die Bearbeitung von sehr großen Geometrien. Bis zu einer Größenordnung von ca. 20000 Eckpunkten liegt die Wartezeit bei geometrischen Operationen in einem akzeptablen Bereich von einigen Sekunden. Bei komplexeren Geometrien sinkt die Geschwindigkeit entsprechend. Die Eckpunkt-Bearbeitung zeigt allerdings schon bei ca. 10000 Eckpunkten Performance-Schwächen.

Für die grafische Darstellung von komplexen Geometrien gibt es Methoden zur Generalisierung, um die Datenmenge zu reduzieren [22]. Diese Ansätze eignen sich jedoch nicht für den Einsatz in einem Geometrie-Editor, da die Geometriedaten hier bearbeitet werden sollen.

Ein relativ einfach umzusetzender Ansatz diesem Problem zu begegnen, wäre ein Algorithmus, der die Geometrie-Bearbeitung auf die sichtbaren Bereiche der Geometrie einschränkt. Dadurch würde sich die Anzahl an darzustellenden und zu verarbeitenden Eckpunkten reduzieren. Der veränderte Teil der Gesamtgeometrie müsste dann nach der Editier-Operation mit der restlichen Geometrie vereinigt werden.

Literatur

- [1] P. Korduan, M.L. Zehner: Geoinformation im Internet. Technologien zur Nutzung raumbezogener Informationen im WWW, Wichmann Verlag, Heidelberg, 2007
- [2] R. Bill, M.L. Zehner: Lexikon der Geoinformatik, Wichmann Verlag, Heidelberg, 2001
- [3] P. Korduan, S. Rahn: Generischer Layereditor und attributgenaue Rechteverwaltung von Internet-GIS-Layern in kvwmap, 3. GeoForum MV, 2007
- [4] World Wide Web Consortium, Scalable Vector Graphics (SVG) 1.1 Specification, <http://www.w3.org/TR/SVG>, 2003
- [5] World Wide Web Consortium, Cascading Style Sheets Level 2 CSS2 Specification, <http://www.w3.org/TR/REC-CSS2>, 1998
- [6] World Wide Web Consortium, Document Object Model (DOM) Level 3 Core Specification, <http://www.w3.org/TR/2004/REC-DOM-Level-3-Core-20040407>, 2004

- [7] Ecma International, ECMA-262 ECMAScript Language Specification, <http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf>, 1999
- [8] PostGIS, <http://www.postgis.org/>, letzter Zugriff: 06.06.2008
- [9] PostgreSQL, <http://www.postgresql.org/>, letzter Zugriff: 06.06.2008
- [10] Open Geospatial Consortium, OpenGIS Simple Features Specification For SQL, OGC 06-104r3, <http://www.opengeospatial.org/standards/sfs>, 2006
- [11] J. J. Garrett, Ajax: A New Approach to Web Applications, <http://www.adaptivepath.com/ideas/essays/archives/000385.php>, 2005
- [12] Kvwmap, <http://kvwmap.geoinformatik.uni-rostock.de/index.php>, letzter Zugriff: 06.06.2008
- [13] Mapserver, <http://mapserver.gis.umn.edu/>, letzter Zugriff: 06.06.2008
- [14] PHP, <http://www.php.net/>, letzter Zugriff: 06.06.2008
- [15] PHP MapScript, http://www.maptools.org/php_mapscript/, letzter Zugriff: 06.06.2008
- [16] Apache HTTP Server Project, <http://httpd.apache.org/>, letzter Zugriff: 06.06.2008
- [17] A. Neumann, A. Winter, Dynamic Loading of Vector Geodata for SVG Mapping Applications Using Postgis, PHP and `getURL()/XMLHttpRequest()`, http://www.carto.net/papers/svg/postgis_geturl_xmlhttprequest/index.shtml, 2006
- [18] A. Neumann, Using SVG for Online Digitizing and Editing of Geographic Data, SVG Open conference, Tokyo, http://www.svgopen.org/2004/papers/abstract_neumann_svgopen_2004_svg_a_s_editing_environment/, 2004
- [19] Object Identifier Types, PostgreSQL 8.3.1 Documentation, <http://www.postgresql.org/docs/8.3/interactive/datatype-oid.html>, 2008
- [20] Open Geospatial Consortium, OpenGIS Web Feature Service (WFS) Implementation Specification, <http://www.opengeospatial.org/standards/wfs>, 2005
- [21] Open Geospatial Consortium, Web Processing Service, <http://www.opengeospatial.org/standards/wps>, 2007
- [22] D. Douglas, T. K. Peucker, Algorithms for the reduction of the number of points required to represent a digitized line or its caricature, *The Canadian Cartographer*, 10(2):112–122, 1973

Kontakt

Stefan Rahn
STZ Geoinformatik
c/o. Professur für Geodäsie und Geoinformatik
Agrar- und Umweltwissenschaftliche Fakultät
Institut für Management ländlicher Räume
Justus-von-Liebig-Weg 6
18059 Rostock
e-mail: stefan.rahn@uni-rostock.de