Interoperables 3D Routing auf Basis von OpenLS

Ein Emergency Route Service 3D zur Einbindung in eine 3D Geodateninfrastruktur für das Katastrophenmanagement

Pascal Neis, Arne Schilling, Alexander Zipf

Zusammenfassung

In diesem Beitrag wird das Konzept und die Umsetzung eines *Emergency Route Service 3D* (ERS3D) vorgestellt, der auf Basis des bereits etablierten OGC Standards OpenLS zwei weitere wichtige Aspekte abdeckt: Einerseits die Berücksichtigung von Sperrgebieten in Situationen des Katastrophenmanagements und andererseits die Möglichkeit der Einbindung in 3D Endanwendungen. Die Umsetzung macht keine neuen Standards erforderlich, sondern zeigt, wie durch geschickte Kombination bereits bestehender OGC Dienste anspruchsvollere Dienste realisiert und in 3D Geodateninfrastrukturen integriert werden können.

1 Einleitung

Die aktuellen Forschungsvorhaben zu Geodateninfrastrukturen (GDI) laufen darauf hinaus, offene Standards umzusetzen, um die Kommunikation zwischen Komponenten voneinander unabhängiger Organisationen zu ermöglichen. Das *Open Geospatial Consortium* (OGC) spielt hierbei eine tragende Rolle. Für die Szenarien des Katastrophenmanagements ist unter anderem der OGC OpenLS Route Service von Bedeutung, der Routing Dienste bereitstellt. Open source GDI-Frameworks wie deegree (FITZKE et al. 2004) bieten bereits die wesentlichen Komponenten zum Aufbau von Geodateninfrastrukturen an. In diesem Beitrag soll ein besonderer Aspekt näher betrachtet werden: die Umfahrung von nicht befahrbaren oder abgesperrten Gebieten, beispielsweise durch eine Bombenentschärfung oder sonstige Einwirkungen verursacht. Dies ist sowohl für die Leitung von Einsatzkräften als auch für betroffene Bürger relevant. Ein entsprechender Service wurde für das Projekt OK-GIS (www.ok-gis.de) umgesetzt und in die spezifische GDI für das Katastrophenmanagement eingebunden. Dessen Relevanz wurde durch die durchgeführten Szenario- und Anforderungsanalysen (BIERMANN et al. 2005) explizit bestätigt.

Verschiedene Web Services für das Routing im Katastrophenmanagement wurden bereits bei KEßLER et al. (2003) im Forschungsprojekt *ariadne – GI-Dienste für Notfall-Management-Systeme* prototypisch implementiert. Unter anderem zum Beispiel ein *Emergency Routing Serivce*, allerdings ohne jegliche Berücksichtigung von OGC-Standards.

Weitere Anforderungen für das Routing kommen durch die seit einiger Zeit vorangetriebene Standardisierung von 3D Visualisierungsdiensten und 3D Stadtmodellen hinzu. Vor allem hinsichtlich der Visualisierung und Benutzerführung beim Routing lassen sich die bisherigen Konzepte nicht 1:1 nach 3D übertragen. Je nach Anwendung können ganz unterschiedliche Varianten ideal sein (z. B. Navigation für Fahrzeuge, Fußgänger, Web-

Anwendungen). Der Routing Service kann jedoch durch die lagerichtige Berechnung der Routengeometrie auch 3D-Endanwendungen unterstützen.
Außer den on-board Autonavigationssystemen der bekannten Hersteller gibt es für PDAs eine Reihe kommerzieller Applikationen, die eine kartenbasierte Routenberechnung und Navigationsunterstützung ermöglichen. Die meisten sind vorwiegend auf die Navigation von Fahrzeugen ausgelegt. Viele der bisher angebotenen Geräte bieten mittlerweile einen 3D Modus an, der die Karteninhalte perspektivisch verzerrt darstellt um somit eine bessere räumliche Wahrnehmung zu vermitteln. Die Nutzung echter 3D Modelle für das Routing wird bisher höchstens als prototypische Anwendungen für kleine Bereiche evaluiert.

2 Der OpenLS Route Service

Die *Open Location Services* (OpenLS) Initiative des OGC erarbeitet seit 2000 freie Spezifikationen (Schnittstellen und Protokolle), um für *Location Based Services* (LBS) relevante Dienste zu standardisieren. Das *OpenLS Service-Framework* besteht aus fünf so genannten *Core Services*. Weitere Details finden sich in ZIPF (2001, 2004) und NEIS (2006, 2007). Der OpenLS Core Service 5 - der Route Service (RS) ermöglicht die Routenplanung auf einem Straßennetz. Die OpenLS Spezifikation definiert dabei, wie der RS anzusprechen ist, welche Optionen er bietet und wie die Antworten von ihm aufgebaut sein müssen. Neben Start und Ziel können zahlreiche weitere Kriterien genutzt werden (z. B. Zeit, Distanz, Bewegungstyp etc.). Selbst Start und Ziel lassen sich in einer Fülle verschiedener Varianten angeben (z. B. über Adressen (Freitext oder in vorgegebener Struktur), Koordinaten, Points of Interest - POIs, Geometrien etc.).
Noch umfangreicher sind die Ausgabevarianten:

1. *RouteSummary* - grundsätzliche Informationen, wie Gesamtstrecke und -zeit der Route
2. *RouteGeometry* - Geometrie der ermittelten Route (Linie mit allen Wegpunkten)
3. *RouteInstructions* - "Schritt für Schritt" Fahr- oder Gehanweisungen der ermittelten Route. Dies wurde in einfacher Form in mehreren Sprachen (zurzeit 6) implementiert.
4. *RouteMaps* - Karten, in denen die ermittelte Route dargestellt wird. Es können mehrere RouteMaps gleichzeitig angefragt werden, z. B. eine Übersichtskarte oder detaillierte Start- und Zielpunktansicht der Route.

Der implementierte RS (NEIS 2006) unterstützt die verpflichtenden und die meisten optionalen Parameter der OpenLS RS Spezifikation. Auch Einbahnstraßen wurden berücksichtigt. Für die Routenberechnung wird der bekannte Dijkstra-Algorithmus (DIJKSTRA 1959) angewandt. Um Tippfehler bei Adressen oder POIs zu bereinigen, kommt der Levenshtein Algorithmus (RINGLSTETTER 2003) zum Einsatz. Die OpenLS Spezifikation bietet die Möglichkeit der Routenanfrage zusätzlich auch eine sogenannte *AvoidList* zu übergegeben. In dieser *AvoidList* können Gebiete oder Straßen angegeben werden, durch welche die Route *nicht* verlaufen soll. In diesem Fall ermittelt der RS die "optimale" Route zwischen Start und Ziel durch Umfahrung dieser Ausschlussgebiete. Diese Funktion nutzt der hier vorgestellte *Emergency Route Service*, um Gefahrengebiete oder unpassierbare Straßen im Katastrophenfall zu umfahren.

3 Interoperable Routenplanung im Katastrophenmanagement – ein OpenLS konformer Emergency Route Service (ERS)

3.1 Problemstellung

Leider machen Katastrophen in der Regel nicht vor administrative Grenzen halt, deshalb ist die Entwicklung einer offenen Geodaten Infrastruktur (GDI) im Katastrophenmangement Szenario besonders wichtig. Somit muss es nicht nur mittelfristig sondern auch langfristig gesehen möglich sein, unterschiedliche Systeme aus verschiedenen Ländern nutzen zu können. Für Geodienste sollten daher vorzugsweise offene Schnittstellen eingesetzt werden, wie z. B. die OGC- oder ISO Schnittstellen (Weiser et al. 2006). Darauf wird auch in der EU
Richtlinie 2007/2/EG zur Schaffung einer Geodateninfrastruktur in der Europäischen Gemeinschaft (INSPIRE) hingewiesen.
Rettungskräfte wie z. B. Feuerwehr oder das Technische Hilfswerk müssen schnellst möglich zu ihren Einsatzorten kommen. Die Zeit in der sie von ihrer Einsatzstelle zum Einsatzort unterwegs sind kann von essentieller Bedeutung sein. Behinderungen wie Staus, erhöhtes Verkehrsaufkommen oder gesperrte Straßen/Gebiete können die Effizienz der Einsatzkräfte wesentlich verschlechtern. Aus diesem Grund müssen bei der Routenplanung auch die Behinderungen und Gefahren berücksichtigt und umgangen werden.
Des Weiteren soll der ERS nicht nur im Katastrophenfall Hilfe leisten sondern auch bei der Prävention. Es könnte z. B. schon im Vorfeld möglich sein die Ausmaße einer möglichen Bombenentschärfung auf den Verkehr abschätzen zu können. Die Polizei kann somit bereits im Vorfeld eine mögliche Umleitung des Straßenverkehrs planen.

3.2 Lösung: Automatisches Umfahren von Gefahren durch den ERS

Ein ERS ist ein spezieller RS, der bei der Routenermittlung *automatisch* aktuelle Gefahrengebiete und gesperrte Straßen berücksichtigt und umfährt. Anfragen an den ERS erfolgen genauso wie an den OpenLS RS. Der ERS fügt aktuelle Gefahrengebiete und gesperrte Straßen aus einer zentralen, von der Leitzentrale gepflegten Geodatenbank (bzw. in einem WFS) der Katastrophen-GDI in den ursprünglichen Request als *AvoidList* ein. Die geänderte Anfrage wird dann an den RS weitergeleitet. Dieser berechnet die Route wie gewünscht unter Berücksichtigung gesperrter Bereiche und gibt das Ergebnis zurück. Abb. 1 zeigt das Sequenzdiagramm des ERS. Wie zu sehen ist, werden die von uns entwickelten Komponenten (ERS, OpenLS) mit bereits als Open Source Pakete vorhandenen Komponenten wie *Web Feature Service* (WFS) und *Web Map Service* (WMS) kombiniert und über das Internet Protokoll (HTTP POST) aufgerufen.
Die Berücksichtigung aktueller Gefahrengebiete und gesperrter Strassen lässt sich unterschiedlich umsetzen. In unserem Fall werden diese zentral in einem WFS gepflegt. Dies sollte i. d. R. in der Realität von der Leitstelle durchgeführt werden. Die *GetFeatures* Abfrage erfolgt dabei über eine vergrößerte BoundingBox die durch die Start- & Zielposition des *DetermineRouteRequests* definiert ist. Ist eine Adresse als Start- oder Endpunkt angegeben, muss diese vorher geocodiert werden. Bei uns geschieht dies intern anhand einer Datenbank. Alternativ könnte auch ein *OpenLS Location Utility Service (Geocoder/Reverse Geocoder)* zwischengeschaltet werden. Bei KEßLER et al. (2003) wird für das

Erhalten der Gefahrengebiete ein spezieller Dienst definiert, der jedoch keinen OGC Standard berücksichtigt. Wenn eine Straße gesperrt ist oder eine als Fläche definiertes Gebiet nicht durchfahren werden kann oder soll, erfolgt zur Ermittlung der nicht befahrbaren Straßen im RS eine geometrische Verschneidung zwischen dem oder den das Gefahrengebiet beschreibenden Polygon(en) und dem Straßennetz.

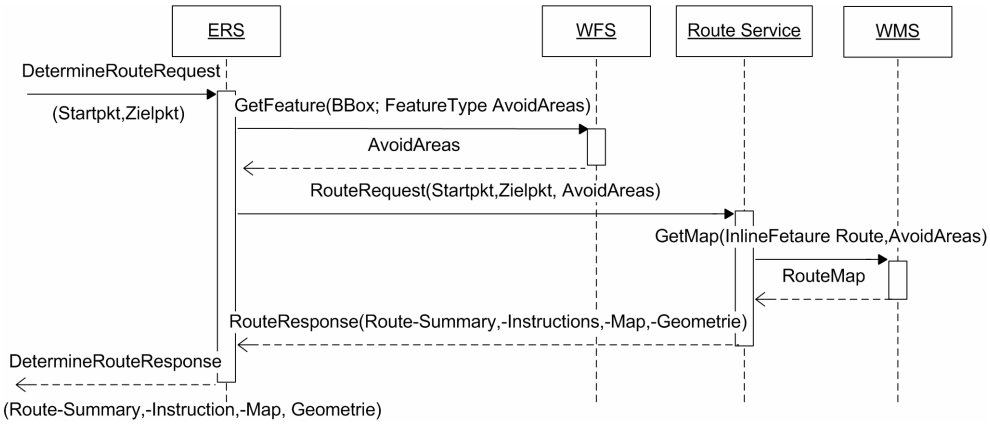

Abb. 1: Emergency Route Service Sequenzdiagramm.

4 Tourenplanung im Rahmen einer 3D-GDI

4.1 3D-GDI - das Beispiel Heidelberg / W3DS

Die vorgestellten Routing-Dienste werden zurzeit in einem größeren Framework integriert und auf ihre Nutzbarkeit für konkrete Anwendungen hin evaluiert. Dabei steht die interoperable Nutzung und Visualisierung von 3D Stadtmodellen im Vordergrund (http://www.3d-gdi.de). Unter den gegenwärtig im OGC diskutierten Vorschlägen spielt hierfür der *Web 3D Service* (W3DS) eine große Rolle, der einen weiteren Schritt darstellt um, auch 3D Daten mittels OGC Diensten verarbeiten zu können. Der W3DS ist ähnlich dem schon standardisierten *Web Terrain Service* (WTS), liefert aber im Gegensatz zu diesem keine perspektivischen Ansichten als geränderte Bilder, sondern 3D Geometrien bzw. Szenen in Standardaustauschformaten, die in jedem Internet Browser mit passendem Plugin angeschaut werden können. Ähnlich auch zum WMS dienen die Bounding Box (räumliche Ausdehnung) und die Namen der Layer als Inputparameter, die dem W3DS übergeben werden. Optional können auch *Point of Interest* (POI), *Point of Camera* (POC), Blickrichtungswinkel (Pitch, Yaw, Roll) und im Service bekannte Styles für die einzelnen Layer übergeben werden. Durch die Kombination von W3DS mit anderen OGC Diensten wie Catalog Service, WFS, WMS, Web Processing Services und anderen wird die Realisierung einer Geodateninfrastruktur für 3D Geodaten (3D-GDI) ermöglicht. Dies ist auch ein Ziel unseres Projektes am Beispiel der Stadt Heidelberg (www.heidelberg-3d.de). Über einen

selbst entwickelten Java3D-Client, der über Java Webstart gestartet wird, können verschiedene OGC Dienste angesteuert und die Ergebnisse dreidimensional visualisiert werden. Insbesondere durch die Einbindung des Routing Service wird die Planung und Visualisierung virtueller Besichtigungstouren durch das 3D-Stadtmodell möglich, ähnlich wie bei ZIPF UND SCHILLING (2002). Parallel dazu wurden weitere Klienten auf Basis der MS .NET Platform (BOCK 2007) und auf mobilen Geräten (FISCHER et al. 2006) entwickelt, die auf die gleiche Infrastruktur zugreifen. Neben den oft genannten Anwendungsszenarien wie Tourismusinformationssysteme, Stadtplanung und –marketing soll auch die Unterstützung im Katastrophen- und Notfallmanagement ermöglicht werden. Ein erster Schritt stellt die Einbindung des Emergency Route Service und die adäquate Präsentation der Routen in 3D dar. Hierzu wurde der Java3D-Viewer erweitert, so dass er den ERS3D ansprechen kann. Zur Realisierung des ERS3D selbst, musste die vom RS berechnete Routengeometrie an das vorhandene Geländemodell angepasst werden. Dies wird im nächsten Abschnitt erläutert. Weitere Geodaten wie die Gebäude, Landmarken, Ebenen für Wald, Grünflächen, Baublöcke, Gewässer und Straßen, sowie Bäume, Ampeln und Verkehrsschilder werden vom W3DS als VRML Modelle angefordert und können über einen Importer direkt im Java3D Viewer dargestellt werden.

4.2 RS3D – ein OpenLS Routenplaner 3D

Routenplanung basiert bekanntlich auf topologischen Graphen. Dass den Knoten und Kanten eine konkrete Geometrie zugeordnet sein kann, ist lediglich für die Visualisierung relevant. Die hierbei verwendeten Koordinaten sind in der Regel 2D. Wir benötigen aber 3D-Visualisierung, d. h. eine Überlagerung der Ergebnisroute auf ein DGM. Dies kann auf zwei Arten realisiert werden: Entweder nimmt der Client die 2D-Geometrie und passt sie während der Visualisierung an das DGM an. Dies ist aber nur für Thick Clients möglich, da diese Funktion z. B. Standard VRML-Browser nicht bieten. Da wir auf eigenständig entwickelte Clients setzen, wäre dies durchaus möglich, wobei jedoch der Rechenaufwand zunehmen würde. Insbesondere für mobile 3D-Clients auf Smartphones oder PDAs ist dies zu rechenaufwändig und sollte daher schon serverseitig vorberechnet werden. Nun könnte ein RS prinzipiell durchaus 3D Koordinaten zurückgeben. Die GML-Koordinaten in der zurückgegebenen Routengeometrie haben dann einfach 3 statt 2 Koordinatenwerte. Hierzu benötigt der RS3D aber Informationen über die Geländeoberfläche. Nun ist es aber so, dass nicht einfach jedem Punkt des Straßennetzes statisch ein Höhenwert fest zugeordnet sein kann. Vielmehr ist eine dynamische Anpassung an das DGM notwendig. Dies liegt daran, dass das DGM auch dynamisch ist, d. h. je nach Level of Detail bzw. Anwendung, Client-Ressourcen, etc. unterschiedliche Genauigkeiten unterstützt werden müssen (vgl. SCHILLING & ZIPF 2003). Außerdem sind auch die Punkte in der Geometrie einer Route, wie sie vom RS berechnet werden, nicht unbedingt fest, sondern können durch Generalisierungsverfahren verändert werden, oder durch Erweiterungen, wie einer nutzergerechten Segmentierung (KRAY et al. 2003), um ideale Sprachausgaben (*RouteInstructions*) zu generieren. HANSEN et al. 2006 schlagen zudem Erweiterungen der OpenLS RS Spezifikation zur kognitiv adäquate Unterstützung bei der Navigation vor. Ziel bei unserer Umsetzung des RS3D war es, möglichst OGC konform zu bleiben, und eine Veränderung des Standards zu vermeiden. Dies ermöglicht es uns, den RS3D ohne größeren Aufwand in die 3D-GDI von Heidelberg einzubinden. In der Tat weicht die Schnittstelle des RS3D nicht von der Spezi-

fikation für den OpenLS RS ab. Nur intern wird zusätzlich die Anpassung der Routengeometrie durchgeführt. Als Output besitzen die Koordinaten des GML-Linestrings dann zusätzlich Höhenwerte. Da zudem als Datenquelle für das DGM ein WFS verwendet wird, konnten wir dem Ziel möglichst hoher OGC Konformität voll gerecht werden. Eine zukünftig mögliche Erweiterung wäre die Auslagerung der Interpolations-Funktionalität, die die Punkte auf das DGM rechnet in einen *Web Processing Service* (WPS) (vgl. KIEHLE et al. 2006). Hierdurch würde ein noch höherer Grad der Modularität erreicht werden - allerdings auf Kosten der Performanz.

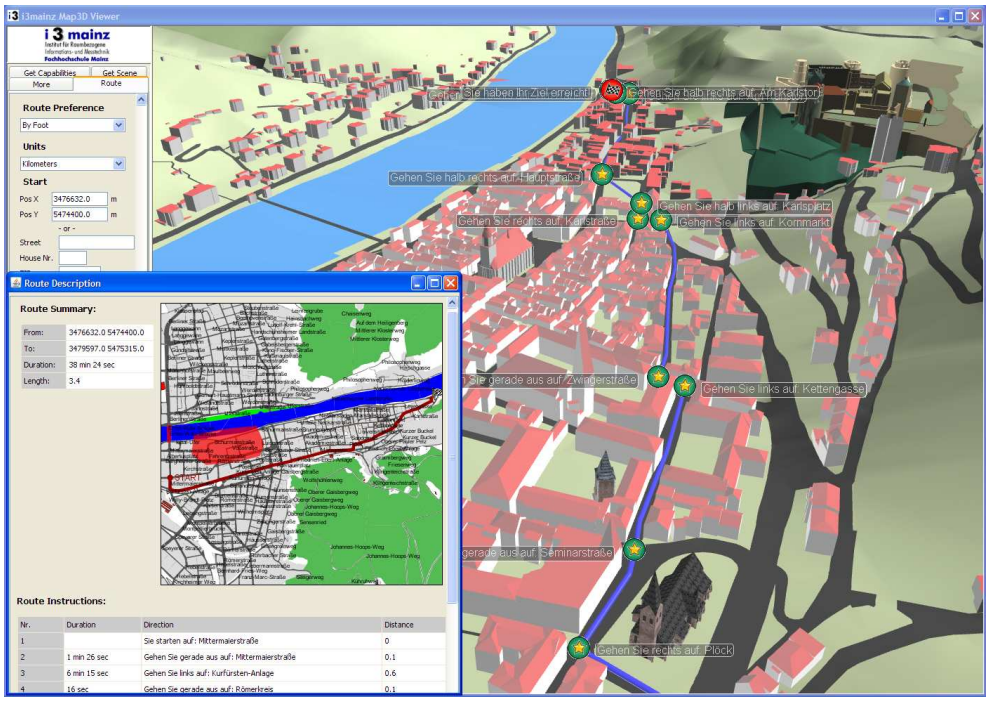

Abb. 2: Darstellung der Route im 3DViewer. In einem separaten Fenster werden die OpenLS typischen Informationen wie *RouteSummary*, *RouteInstructions* und *RouteMap* angezeigt (Daten: Vermessungsamt Heidelberg und EML Heidelberg).

Die Umsetzung des RS3D erfolgt als Kaskade von OpenLS RS, sowie weiteren OGC Diensten. Anfragen an den RS3D erfolgen wie erwähnt genauso wie an den OpenLS Route Service. Sie werden einfach an einen normalen OpenLS Route Service durchgeschleift, der die eigentliche Routenberechnung durchführt. Das in der Umgebung (OGC-Filterfunktion *DWITHIN*) der Geometrie der Routensegemente befindliche DGM wird von einem WFS via *GetFeature* (unter Nutzung einer PostGIS Datenbank) angefordert. Obwohl es möglich wäre, einfach für jede Koordinate einen Höhenwert auf dem DGM zu ermitteln, führt dies bei unebenem Gelände schnell zu sichtbaren Problemen. Beispielweise werden Erhebungen unterschnitten. Daher wird für jedes Routensegment der genaue Verlauf des DGMs ermit-

telt. Das DGM ist ein TIN aus den topologischen Elementen Knoten, Kanten und Dreiecken. Ein räumlicher Index (Quadtree) sorgt für einen schnellen Zugriff für räumliche Vergleichsoperatoren (Touches, Intersects, Contains etc.). Dieser Index wird dazu verwendet, die topologischen Kanten des TINs ausfindig zu machen, die genau unter den Routensegmenten liegen. Daraufhin werden die Höhen der 2D Schnittpunkte des Segments mit der DGM-Kante über die Koordinaten der Knoten interpoliert. Die neuen Punkte werden in der richtigen Reihenfolge in den neu erzeugten GML *Linestring*, der die Routengeometrie beschreibt, eingefügt. Die Datenmenge für die geometrische Beschreibung des Routenverlaufs erhöht sich damit natürlich je nach Feinheit des DGMs erheblich. Der so erzeugte GML *Linestring* wird in die Antwort des OpenLS RS eingearbeitet und mit als Ergebnis des RS3D zurückgegeben. Im Client kann die so erstellte 3D Route leicht zusammen mit dem restlichen Stadtmodell dargestellt werden - in unserem Fall als röhrenförmige Geometrie die mittels eines „Sweeping" Befehls erstellt wird, der ein Profil entlang des Pfades extrudiert und einige Meter vertikal nach oben versetzt (Abb. 2). Weitere Ausgabevarianten wie die *RouteInstructions, RouteSummary* und *RouteMaps* werden in einem separaten Fenster eingeblendet. Zusätzlich wird eine Animationssequenz bzw. Kamerafahrt entlang der Route automatisch erzeugt, so dass sie virtuell abgefahren werden kann. Dabei hat sich jedoch gezeigt, dass die Routensegmente zu feingliedrig für die Animation sind. Für jedes Segment wird ein Keyframe festegelegt. Bei sehr vielen kleinen Segmenten kann es dadurch zu abrupten Richtungsänderungen zwischen den Keyframes kommen. Um dem abzuhelfen wird nachträglich die Routengeometrie, wie sie vom RS3D kommt, generalisiert. Dazu wird ein Douglas Peucker Algorithmus (DIJKSTRA 1959) verwendet, der zusätzlich auch die z Werte mit berücksichtigt. Dadurch wird die Darstellung zusätzlich verbessert, da viele kleine und kaum relevante Segmente herausgefiltert werden. Das Sequenzdiagramm zum RS3D (Abb. 5) ist sehr ähnlich zum im Folgenden vorgestellten ERS3D (Abb. 3).

5 Zusammenführung von RS3D & ERS: Der ERS3D

Die oben beschriebenen Routing Dienste RS3D und ERS und die darin verwirklichten Konzepte können parallel genutzt werden, da sie wie für eine GDI zu erwarten auf separaten Rechnern installiert werden können. Oder aber sie werden als letzter Schritt noch einmal aggregiert zu einem *Emergency Route Service 3D* (ERS3D).
Abb. 3 zeigt das Sequenzdiagramm für diesen Service. Einziger Unterschied zwischen einem ERS3D und ERS ist das der Request mit der eingearbeiteten *AvoidLists* nicht an einen normalen RS sondern an einen RS3D weitergeleitet wird. Dadurch ist später die *RouteGeometry* in 3D statt in 2D. Ansonsten läuft die Prozesskette wie schon weiter oben beschrieben ab. Somit stehen beide hier vorgeschlagenen Features, die über ein normales Routing weit hinausgehen, in einem einzigen Service zur Verfügung, der sich sehr gut in einer 3D GDI einbinden und in einem konkreten Anwendungsszenario einsetzen lässt. Die einzig wünschenswerte Erweiterung wäre die Erzeugung der 3D Geometrien für die Darstellung der *AvoidAreas* direkt durch den ERS3D. Dies ist jedoch nicht mehr standardkonform, so dass darauf verzichtet wurde, um der OpenLS Spezifikation zu entsprechen. Daher muss der Java Client selbst den WFS ansteuern und die *AvoidAreas* entsprechend visualisieren, da die 2D Karte dafür kaum zu gebrauchen ist. Zurzeit werden per Picking einige Höhenpunkte ermittelt und die Polygone einige Meter nach oben extrudiert (Abb. 4).

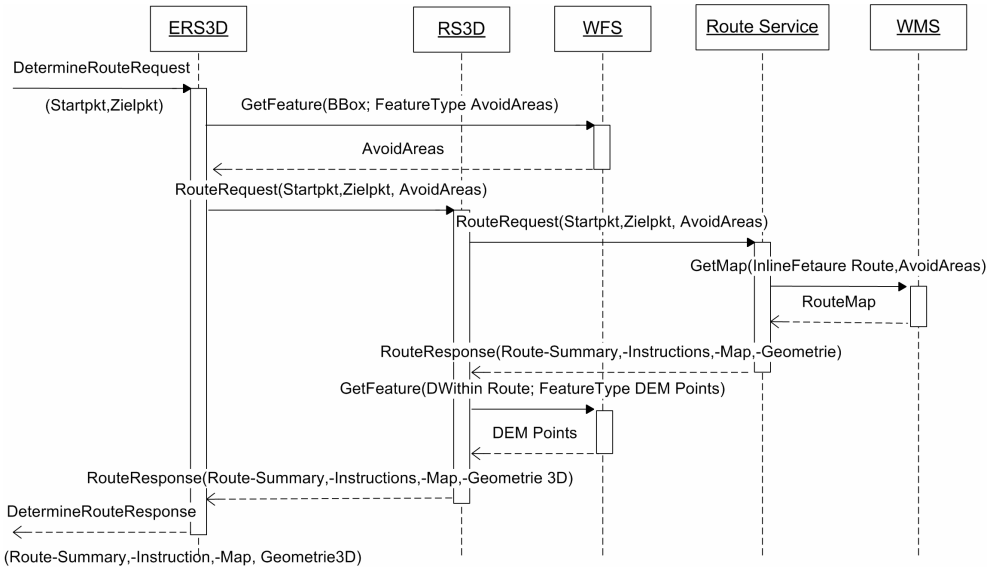

Abb. 3: Emergency Route Service 3D Sequenzdiagramm (Detail).

6 Ausblick

Das Konzept des ERS3D baut aus einzelnen standardisierten Diensten als Bausteinen immer komplexerer Anwendungen und Funktionalitäten auf. Dies geschieht durch Aggregation bzw. Verkettung der Dienste. Dies zeigt die Nützlichkeit standardisierter modularer Bausteine und beweist, dass OGC heute viel mehr bietet als reines Web-Mapping. Routenplanung und Navigationsunterstützung wird schon in absehbarer Zeit verstärkt dreidimensional visualisiert werden – mehrere Hersteller von Autonavigationssystemen arbeiten an diesem Thema. Spezielle Aspekte wie z.B. der mobilen 3D-Routennavigation mittels Landmarken etc. werden im Rahmen des neuen BMBF-Projektes „MONA3D – Mobile Navigation 3D“ untersucht werden. Zukünftig ist die Performanz des RS3D zu evaluieren, wenn er sehr lange Routen auf ein DGM anpassen muss. Zu einem müssen hierzu erhebliche Daten für das DGM von einem WFS abgerufen werden, zum anderen muss das DGM erstellt und die Höheninformationen der Route auf dem DGM berechnet werden. Eine Implementierung der restlichen OpenLS Services wird aktuell umgesetzt. So kann die Implementierung des OpenLS Location Utility Service (Geocoder/ReverseGeocoder) im ERS3D und ERS zum Geocodieren der Start- und/oder Zieladresse zum Einsatz kommen. Zusätzliche Erweiterungen aller OpenLS Core Services (nicht nur die des RS) in Richtung 3D wären zusätzlich sehr interessant und werden auch im OGC diskutiert. Vorarbeiten hierfür erfolgten bereits durch ZLATANOVA & VERBREE (2005). Allerdings waren diese nur konzeptionell und ohne genaue Untersuchungen und konkrete Vorschläge zur Erweiterungen der Schnittstellen bzw. gar einer Implementierung. Zudem ist zu evaluieren, ab welcher Granularität es sinnvoll sein kann, interne Berechnungen wie zum Beispiel die Ermittlung der Höheninformationen der Routengeometrie von einem *Web Processing Service* (WPS)

berechnen zu lassen. Weitere interessante Erweiterungsmöglichkeiten betreffen die Unterstützung von Indoor-Navigation und deren 3D-Visualisierung (Mohan and Zipf 2007).

Danksagung
Wir danken allen Kollegen für ihre Hilfe und Inspiration. Wir danken dem Vermessungsamt Heidelberg und dem European Media Laboratory für die Bereitstellung von Datengrundlagen. Die Arbeit wurde von der Klaus-Tschira-Stiftung (KTS) gGmbH und über das BMBF-Projekt „OK-GIS“ gefördert.

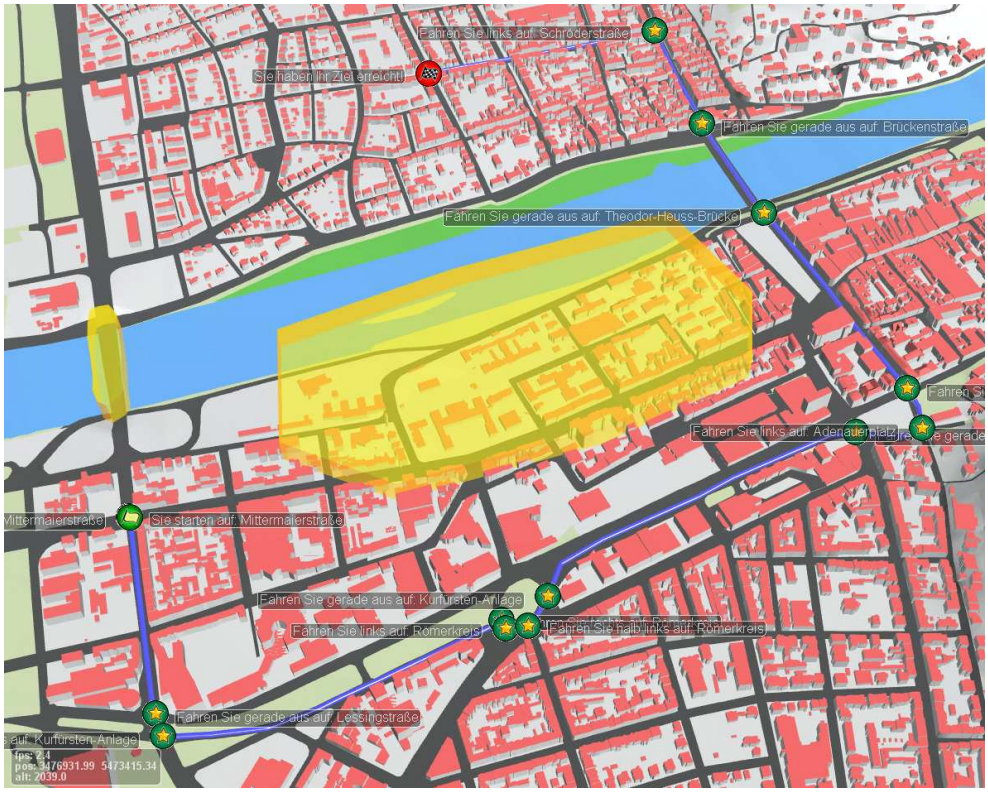

Abb. 4: Graphischer Output des ERS3D im 3DViewer. Die Gefahrengebiete werden als halbtransparente extrudierte Polygone dargestellt.

Literatur

BIERMANN, J., GERVENS, T. & HENKE, S. (2005): Analyse der Nutzungszenarien und Aktivitäten der Feuerwehr. Internes Projektdokument. Projekt OK-GIS.

BOCK, N. (2007 in Arbeit): Entwicklung eines W3DS Client auf Basis von MS .Net. Diplomarbeit. Fachhochschule Mainz.

DIJKSTRA E.-W. (1959): Note On Two Problems In Connexion With Graphs, In: Numerische Mathematik, Springer Verlag pp. 269-271, vol. 1, Berlin.

FISCHER, M., BASANOW, J. & ZIPF, A. (2006): Mainz Mobile 3D - A PDA based client for the OGC Web 3D Service and corresponding server. International Workshop on 3D Geoinformation 2006 (3DGeoInfo'06). Kuala Lumpur. Malaysia.

FITZKE, J; GREVE, K; MÜLLER, M. & POTH, A. (2004): Building SDIs with Free Software - the deegree Project. In: Proceedings of GSDI- 7, Bangalore, India.

HANSEN, S., RICHTER, K.-F., & KLIPPEL, A. (2006): Landmarks in OpenLS - A Data Structure for Cognitive Ergonomic Route Directions. In M. RAUBAL, H. MILLER, A. U. FRANK, M. F. GOODCHILD (Eds.): Geographic Information Science - Fourth International Conference, GIScience 2006, pp. 128–144. Springer, Berlin.

HEIER, C. & KIEHLE, C. (2006): Automatisierte Liegenschaftsauskunft mittels OGC Web Processing Service. GeoBit – GIS 2006, 7: 12-16.

KEßLER, C., S. SCHADE, A. STARKE, S. TEGTMEYER & WALKOWSKI, A. (2003): ariadne – GI-Dienste für Notfall-Management-Systeme. In: BERNARD, L., A. SLIWINSKI & K. SENKLER : Geodaten- und Geodienste-Infrastrukturen – von der Forschung zur praktischen Anwendung. Münster: 297-309. (= IfGIprints, Bd. 18)

KRAY, C., ELTING, C., LAAKSO, K. & COORS, V. (2003): *Presenting Route Instructions on Mobile Devices*. Proceedings of IUI'03, ACM Press, New York, NY, pp. 117-124.

MOHAN, S. K. & ZIPF, A. (2007 submitted): Improving the support of indoor evacuation through landmarks on mobile displays.

NEIS, P., (2006): Routenplaner für einen Emergency Route Service auf Basis der OpenLS Spezifikation. Diplomarbeit FH Mainz.

NEIS, P. (2007 in Arbeit): Die OpenLS Core Services. In ANDRAE, C., FITZKE, J., ZIPF, A., (Hrsg.): OpenGIS Essentials - Spezifikationen des OGC im Überblick, Wichmann Hüthig Verlag. Heidelberg.

RINGLSTETTER, C. (2003): OCR-Korrektur und Bestimmung von Levenshtein-Gewichten. Magisterarbeit im Studiengang Computerlinguistik. LMU, München.

WEISER, A., NEIS, P. & ZIPF, A. (2006): Orchestrierung von OGC Web Diensten im Katastrophenmanagement - am Beispiel eines Emergency Route Service auf Basis der OpenLS Spezifikation. In: GIS - Zeitschrift für Geoinformatik. Abc-Verlag. 09/2006. pp. 35-41.

ZLATANOVA, S. & VERBREE, E. (2005): The third dimension in LBS: the steps to go. In: Geowissenschaftliche Mitteilungen, Heft Nr. 74, 2005, pp. 185-190.

ZIPF, A. (2001): Interoperable GIS-Infrastruktur für Location-Based Services (LBS) - M-Commerce und GIS im Spannungsfeld zwischen Standardisierung und Forschung. In: GIS Geo-Informations-Systeme. Zeitschrift für raumbezogene Information und Entscheidungen. 09/2001. 37-43.

ZIPF, A. & SCHILLING, A. (2002): Automatisierte Integration und Visualisierung von verteilten 2D- und 3D-Geodaten am Beispiel einer virtuellen Stadttour. AGIT 2002. Symposium für Angewandte Geographische Informationsverarbeitung. 03.-05.07.2002. Salzburg. Austria.

SCHILLING, A. & ZIPF, A. (2003): Generation of VRML City Models for Focus Based Tour Animations. Integration, Modeling and Presentation of Heterogeneous Geo-Data Sources. 8th Int. Symp. on Web 3D Technology. Web3D 2003. 03/2003 Saint Malo, France.

ZIPF, A. (2004): Mobile Anwendungen auf Basis von Geodateninfrastrukturen - von LBS zu UbiGIS. In: Bernard, L.; FITZKE, J. & WAGNER, R. (eds): Geodateninfrastrukturen. Wichmann Verlag. Heidelberg.