



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Departement für Bau, Umwelt und Geomatik
Institut für Kartografie

Diplomarbeit

Interaktive Karte von Methana für mobile Endgeräte

vorgelegt von
Adrian Weber

Betreuung:
Prof. Dr. L. Hurni
Dipl. Ing. I. Iosifescu
Dipl. Ing. T. Dahinden

Vorwort

Die vorliegende Diplomarbeit entstand im Zeitraum von April bis Juli 2006 am Institut für Kartografie an der ETH Zürich und bildet den Abschluss meines Studiums der Geomatikingenieurwissenschaften.

Ich möchte hiermit allen danken, die mich während dieser Diplomarbeit unterstützt haben.

Als erstes möchte ich Prof. Dr. L. Hurni danken, der als Vorsteher des Instituts für Kartografie diese spannende Arbeit ermöglicht, unterstützt und eine hervorragende Infrastruktur zur Verfügung gestellt hat.

Vielen Dank geht an Ionut Iosifescu und Tobias Dahinden, die mich während der Arbeit mit Tipps und Anregungen tatkräftig unterstützt haben.

Danken möchte ich auch Jürgen Weitkämper und Christian Rolfs von der FH Oldenburg für ihre Bemühungen und Erklärungen zum eingesetzten SVG-Viewer.

Weiter geht Dank an meinen Studienkollegen Andreas Gauer, der mir bei der Bearbeitung der Grundlagedaten wertvolle Hilfe geleistet hat.

Ich danke ganz herzlich meinen Eltern, die mir dieses Studium ermöglicht haben, und allen Freunden und Studienkollegen, die mich in dieser Zeit unterstützt haben.

Zürich, Juli 2006
Adrian Weber

Zusammenfassung

In dieser Arbeit wird eine interaktive Karte von Methana für mobile Geräte entwickelt. Anhand dieser Karte werden technische und kartografische Probleme mobiler Geräte erörtert und Lösungsvorschläge erarbeitet.

Nebst einer Basiskarte des Testgebiets beinhaltet die Anwendung POI. Die Karte wird im Vektorgrafikformat SVG entwickelt und zur Umsetzung von Interaktionen werden ECMAScripts verwendet. Zur Ermittlung der eigenen Position wird ein integrierter GPS-Empfänger eingesetzt und die Position in die Karte eingebunden. Für eine massstabsgerechte Darstellung der Karte wird eine adaptive Visualisierung eingesetzt.

Die verschiedenen Interaktionsmöglichkeiten werden ausführlich beschrieben. Dank dem Einsatz von GPS können die momentane Position in der Karte angezeigt und POI gesucht werden. Dabei kann es sich um einen beliebigen POI oder um den nächstgelegenen handeln. Über jeden POI lassen sich Informationen abfragen. Weiter kann der Benutzer neue POI oder neue Routen aufnehmen und in eine Datei schreiben lassen.

Es werden zudem einige Anwendungsfälle aufgezeigt, wie die interaktive mobile Karte eingesetzt werden könnte.

Inhaltsverzeichnis

1	Einführung	1
1.1	Hintergrund	1
1.2	Inhaltliche Ziele	1
1.3	Technische Problemstellung	2
1.4	Testgebiet Methana	2
2	Grundlagen und Theorie	2
2.1	Location Based Services	2
2.2	Positionsbestimmung	3
2.3	Anforderungen an Karten für mobile Geräte	4
2.4	Verwandte Arbeiten	5
2.4.1	Kartografische Arbeiten über Methana	5
2.4.2	Arbeiten und Projekte im Bereich der mobilen Kartografie	5
3	Konzept	7
3.1	Inhalt	7
3.2	Datenorganisation	7
3.3	Kartenentwurf	8
3.4	Technik	8
4	Grundlagedaten und Techniken	9
4.1	Grundlagedaten	9
4.2	Scalable Vector Graphics (SVG) und ECMAScript	9
4.3	Das Satellitennavigationssystem GPS	10
4.4	UTM-Projektion	11
4.5	Evaluation eines mobilen Geräts	11
4.6	SVG-Viewer für mobile Geräte	13
4.6.1	Übersicht über SVG-Viewer	13
4.6.2	SVG-Viewer für mobile Endgeräte der FH Oldenburg	16
4.6.3	SVG±geo	16
4.6.4	Anpassungen des Viewers	17
4.7	Weitere eingesetzte Software	17
4.7.1	ArcGIS	17
4.7.2	PostgreSQL und PostGIS	17
5	Technische Umsetzung	18
5.1	Datenaufbereitung	18
5.1.1	Generalisierung	18
5.1.2	Kachelung	19
5.2	Aufbau der SVG-Anwendung	19
5.3	Einbindung der GPS-Koordinaten	20
5.3.1	Einbindung der GPS-Koordinaten in das SVG-Dokument	20
5.3.2	Erweiterung des SVG-Viewers um UTM-Projektion	20
5.4	Implementierung der Interaktionen	20

5.4.1	Vergrössern und Verschieben	20
5.4.2	Platzierung der Menüleisten	21
5.4.3	Laden der Kacheln	21
5.4.4	Anzeige von Informationen über POI	21
5.4.5	Ein- und Ausschalten der Ebenen	21
5.4.6	Ein- und Ausschalten GPS-Empfänger	22
5.4.7	Anzeige aktueller Position in Koordinaten	22
5.4.8	Finden von POI	22
5.4.9	Finden des nächstgelegenen POI	22
5.4.10	Erfassen neuer POI	22
5.4.11	Erfassen neuer Routen	23
5.4.12	Speichern neuer POI und Routen	23
5.5	Kartografische Darstellung	24
5.5.1	Generalisierung	24
5.5.2	Nomenklatur	24
5.5.3	Symbolisierung POI	25
6	Anwendungsfälle	26
6.1	Einführung in die Anwendungsfälle	26
6.2	Anwendungsfall 1: Tourist will Methana kennen lernen	26
6.3	Anwendungsfall 2: Tourist befindet sich auf Methana am Wandern	28
6.4	Anwendungsfall 3: Reiseleiter will neue Sehenswürdigkeiten und Wanderun- gen aufnehmen	30
7	Schlussfolgerung	33
7.1	Beurteilung der fertigen Anwendung	33
7.2	Ausblick	36
A	Quellcode zur Einbindung der GPS-Koordinaten in das SVG-Dokument	41
B	Funktion CSVGViewerDemoView::OnSaveFile()	42
C	Auszug aus Datei der gespeicherten POI	43

Abbildungsverzeichnis

1	Interaktive Webkarte von Methana	6
2	Kartenentwurf	8
3	Methana in griechischer Schrift	24
4	Symbolisierung POI	25
5	Screenshot: Display nach dem Öffnen der Anwendung	26
6	Screenshot: Eingblendete Titel- und Legendenleisten und Basiskarte	27
7	Screenshot: Auswählen und Finden eines bestimmten POI	28
8	Screenshot: Anzeige von Informationen über POI	29
9	Screenshot: GPS Cursor und Koordinaten	30
10	Screenshot: Auswählen und Finden eines bestimmten POI mit GPS	31
11	Screenshot: Finden des nächstgelegenen POI	32
12	Screenshot: Aufnahme von neuen POI	33
13	Screenshot: Aufnahme neuer Routen	34
14	Screenshot: Aufnahme neuer Wegpunkte	35
15	Screenshot: Speichern neuer POI und Routen	36

Tabellenverzeichnis

1	Kriterienliste zur Evaluation eines Testgerätes	12
2	Eigenschaften des Modells <i>Pocket Loox N520</i>	13
3	Vergleich von SVG-Viewer für mobile Geräte	14

1 Einführung

1.1 Hintergrund

Dank der fortschreitenden Miniaturisierung in der Computertechnik sind Anfangs der 90iger Jahren die ersten kleinen und tragbaren Computer, die sogenannten PDA, auf den Markt gekommen. Die Abkürzung PDA steht für Personal Digital Assistant, da diese mobilen Geräte vor allem der persönlichen Kalender-, Adressen- und Aufgabenverwaltung dienen. Heutige Modelle verfügen über Rechnerleistungen und Speicherplatz wie Heimcomputer vor weniger als zehn Jahren. Da diese Geräte sehr handlich sind und mühelos mitgenommen werden können, ist es naheliegend diese zur Orientierung zu verwenden. Aus diesem Grund verfügen neueste PDA über integrierte GPS-Satellitenempfänger.

Mobile Karten, d.h. digitale Karten auf PDA, kombiniert mit der Möglichkeit der Positionsbestimmung, versprechen vielseitige Anwendungsmöglichkeiten. Eine wichtige Anwendung sind die sogenannten Location Based Services (LBS), die den Nutzern ortsbezogene Informationen anbieten.

Ein wichtiger Unterschied digitaler Karten gegenüber Papierkarten sind die Interaktionsmöglichkeiten. Am Institut für Kartografie gibt es mehrere Arbeiten über interaktive Karten für Desktop-Computer. Ausgehend von diesen zahlreichen Erfahrungen soll nun eine interaktive mobile Karte, d.h. eine interaktive Karte für PDA, realisiert werden. In dieser Arbeit werden nebst dem Umgang mit den technischen Restriktionen von mobilen Geräten die kartografischen Aspekte kleiner Displays untersucht.

1.2 Inhaltliche Ziele

Die Ziele der Arbeit sind die folgenden:

Unter Berücksichtigung sämtlicher Einschränkungen mobiler Geräte soll eine ansprechende, gut lesbare Karte mit Interaktionsmöglichkeiten auf einem mobilen Gerät entwickelt werden. Die kartographische Umsetzung der Karte soll sich deutlich von denjenigen der Navigationssystemen unterscheiden, welche nebst den Strassen kaum Informationen enthalten. Die Umsetzung soll sich vielmehr am Erscheinungsbild einer topographischen Karten orientieren. Die aktuelle Position des Benutzers soll mittels GPS-Empfänger ermittelt und in der Karte angezeigt werden. Diese soll jederzeit innerhalb des angezeigten Kartenausschnitts liegen.

Der Benutzer kann Punkte von speziellem Interesse, sogenannte Points of Interest (POI) suchen und Richtung und Distanz zu diesem anzeigen lassen. Auf diese Weise können POI einfach gefunden werden. Über jeden POI können interaktiv Informationen in Form von Texten, Bildern und evtl. Tönen abgefragt werden. Ebenfalls kann der Benutzer neue, eigene POI definieren, indem er am entsprechenden Ort die Position und Informationen über diesen speichert. Solche Informationen können z.B. eine Beschreibung oder ein Bild sein, welches mit einer integrierten Kamera gemacht wird. Nebst Einzelobjekten sollen auch Linienzüge gespeichert werden können um so z.B. neu erstellte Wege aufzunehmen. Die gesammelten Punkte sollen anschliessend in ein Format exportiert werden können, welches

von Geoinformationssystemen (GIS) verstanden und weiterverarbeitet werden kann. Werden alle oder zumindest einige dieser Ziele erreicht, wird dem Nutzer eine Anwendung zur Verfügung gestellt, die ihm gegenüber einer Papierkarte Mehrwerte bringt wie die Unterstützung bei der Orientierung auf der Insel oder beim Finden von POI.

1.3 Technische Problemstellung

Zur technischen Problemstellung gehört zuerst die Evaluation von Geräten und Software. Zur Evaluation eines Gerätes muss eine Kriterienliste mit den Anforderungen an das Geräte erarbeitet werden. Anschliessend wird an Hand dieser Kriterien ein Gerät gewählt. Ein geeignetes Grafikformat und eine Software für mobile Geräte, welche dieses anzeigen kann, müssen gefunden werden. Mit dem gewählten Grafikformat soll eine hohe qualitative Darstellung möglich sein. Weiter soll das Format wie auch die Software zur Betrachtung Interaktionsmöglichkeiten unterstützen, d.h. die Anwendung soll auf die Eingaben des Nutzers reagieren.

Aus den Grundlagedaten muss die Karte erstellt werden und auf das Gerät übertragen werden. Damit die Berechnungszeit für die Anzeige der Karte nicht zu lang wird, muss auf eine optimale Datenorganisation geachtet werden.

Aufgrund des kleinen Displays soll die Karte in geeigneter Weise generalisiert werden und an die aktuelle Zoomstufe angepasst werden können. Weiter müssen anschliessend die Interaktionen wie Zoomen, Suchen und Anzeigen von POI etc. implementiert werden.

Bei einem Feldtest wird die mobile Karte untersucht um abschliessend noch kleine Verbesserungen vornehmen und ein Fazit ziehen zu können.

1.4 Testgebiet Methana

Die griechische Halbinsel Methana liegt etwa 70 km südwestlich von Athen im Saronischen Golf. Die Ausdehnung von Methana beträgt ungefähr 9 mal 10 Kilometer und die Fläche um die 55 Quadratkilometer. Die Halbinsel ist vulkanischen Ursprungs, woher sie auch ihre ziemlich runde Form hat. Seit der Antike ist der Hauptort mit dem gleichen Namen wie die Halbinsel als Kurstadt bekannt. Bis heute gibt es mehrere Thermal- und Heilbäder und zahlreiche antike Ruinen und eine Vielzahl sehenswerter Kirchen und Kapellen [Hurni, 1995] [Gauer, 2006].

2 Grundlagen und Theorie

2.1 Location Based Services

Location Based Services (LBS) oder zu deutsch *ortsbezogene Dienste* sind Geodienste für mobile Benutzer. Aufgrund der aktuellen Position, Zeitdaten und personalisierten Präferenzprofilen werden dem Benutzer angepasste, relevante Informationen zur Verfügung gestellt bzw. auf die Bedürfnislage zugeschnittene Produkte angeboten [Gasenzer, 2001].

Der wichtigste Informationstyp in LBS sind die POI. Ein POI ist ein Standort oder ein Objekt der für den Benutzer von besonderem Interesse ist. Ein solches Objekt können

Restaurants, Geldautomaten, Sehenswürdigkeiten, Tankstellen etc. sein.

Die wichtigsten Funktionen von LBS beruhen im wesentlichen auf GIS-Funktionalitäten. Dazu gehören nach [Reichenbacher, 2003] die Positionierung, die Suche beliebiger geografischer Objekte, Geocodierungsdienste d.h. das Bestimmen von Koordinaten für POI oder Adressen und umgekehrt. Weitere Dienste sind das Finden der nächstgelegenen POI, Routenplanungen und Führung ans Ziel sowie Kartendienste, welche qualitative Karten anbieten.

Reichenbacher unterscheidet zwei Möglichkeiten, wie die Informationen verbreitet werden können. Die eine Möglichkeit sind push-Dienste, welche dem Nutzer Informationen übermitteln, ohne dass dieser das verlangt. Die andere sind pull-Dienste, wo Informationen nur auf Anfrage des Nutzers geschickt werden.

Ein LBS-System ist oftmals als ein sogenanntes Client-Server-System konzipiert. Ein Server stellt verschiedene Dienste und GIS-Funktionalitäten zur Verfügung. Der Benutzer stellt eine Anfrage an einen Server. Der Server verarbeitet diese Anfrage und erzeugt daraus eine Karte, welche dem Benutzer übermittelt wird. Diese Karte kann ein Rasterbild oder eine Vektorkarte sein.

2.2 Positionsbestimmung

Eine wesentliche Komponente von Location Based Services ist die Bestimmung der Position. Die Informationen über die Position umfassen nicht nur den absoluten bzw. relativen Aufenthaltsort des Benutzers, sondern auch weitere Parameter wie die Blickrichtung, Orientierung, Geschwindigkeit und Beschleunigung.

Zur Bestimmung dieser Parameter gibt es verschiedene Möglichkeiten. Eine weit verbreitete davon ist das GPS. Da GPS Satelliten gestützt ist, ist der Empfang von Satellitensignalen unabdingbar. Das kann aber oft gerade in Städten zu erheblichen Problemen führen, wenn hohe Gebäude die Signale abdecken. Innerhalb von Gebäuden ist ein GPS-Empfang ebenfalls nicht möglich.

Zur Umgehung dieser Nachteile gibt es Untersuchungen, ein GPS-System mit einem inertialen Navigationssystem (INS), welches Beschleunigungen in drei Richtungen misst, zu kombinieren [Gabaglio, 2003].

Ein anderer Ansatz die Positionsinformation zu erhalten ist die Verwendung der Zellen des Mobilfunknetzes. Dieser Ansatz hat den Vorteil, dass er auch innerhalb von Gebäuden funktioniert. Doch ist die Genauigkeit, die direkt von den Zellengrößen abhängig ist, gegenüber GPS oftmals mehr als Faktor zehn schlechter.

Wie in [Rerrer, 2005] dargelegt gibt es im weiteren die Möglichkeit die Positionsinformation dank einem WLAN zu ermitteln. Dank den kleineren Zellgrößen können höhere Genauigkeiten als mit Mobilfunknetzen erreicht werden. Vor allem in Gebäuden und in Stadtgebieten ist die WLAN-Positionierung eine mögliche Alternative zu obengenannten Ansätzen. Momentan ist die dafür nötige Infrastruktur aber noch nicht in genügendem Masse verfügbar [Kray et al., 2002].

2.3 Anforderungen an Karten für mobile Geräte

Karten für mobile Geräte unterliegen nicht den gleichen Anforderungen wie interaktive Webkarten, die für einen Desktop-Computer erstellt werden. Strenge Anforderungen an mobile Karten betreffen aus kartografischer Sicht vor allem die kleine Displaygrösse und aus technischer Sicht in erster Linie die beschränkten Ressourcen mobiler Geräte.

Folgende kartografische Lösungen im Umgang mit kleinen Displays werden in [Nissen et al., 2004] vorgeschlagen:

Eine Karte soll aus vier Ebenen mit abgestuften Symbolisierungen bestehen. Die Farbe Schwarz wird nur für den Text verwendet, Navigationselemente werden ausschliesslich in gelb und rot dargestellt und POI werden dunkelblau symbolisiert. Die Hintergrundkarte soll in hellen Farben gestaltet sein ohne die bereits verwendeten Farben Schwarz, Gelb und Dunkelblau.

Die Karte soll möglichst ohne viele Details auskommen. Die POI-Symbole sollen alltäglichen Piktogrammen ähnlich und so dem Kartenleser vertraut sein. Die Grösse dieser Symbole soll unabhängig des aktuellen Kartenausschnitts immer gleich gross sein.

Die eingesetzte Schriftart soll klar und ohne Verzierungen sein. Dabei muss klar sein, auf welches Objekt sich die Beschriftung bezieht. Diese soll, wie die POI, unabhängig des Ausschnitts eine konstante Grösse haben.

Eine wichtige Technik in der mobilen Kartografie ist die adaptive Visualisierung. Diese passt das Kartenbild an die jeweilige Situation und Position, Benutzer, dessen Aktivitäten etc. an. So soll dem Benutzer geholfen werden, die geografischen Informationen schnell erfassen zu können. Die Darstellung wird den Bedürfnissen des Benutzers und den beschränkten Ressourcen des Gerätes angepasst. Reichenbacher unterscheidet dabei in seiner Dissertation die adaptive Visualisierung in vier verschiedene Bereiche, die an die momentane Lage angepasst werden müssen [Reichenbacher, 2003]:

- Informationsinhalt
- Bedienungsoberfläche
- Visualisierung
- Technologie: Anpassung an Gerät, Displaygrösse, Übertragungsgeschwindigkeit etc.

Eng mit der adaptiven Visualisierung verknüpft ist die Generalisierung. Die Schweizerische Gesellschaft für Kartografie definiert Generalisierung so:

„Die Generalisierung ist die massstabsgerechte inhaltliche und grafische Vereinfachung der komplexen Wirklichkeit auf der Grundlage digitaler Landschaftsmodelle, von Plänen oder Karten grösseren Massstabes.

Die Generalisierung besteht hauptsächlich aus der zweckentsprechenden Auswahl und Zusammenfassung der Objekte sowie der möglichst lagegenauen, charakteristischen, richtigen und eindeutigen grafischen Darstellung“ [SGK, 2002].

Ähnlich wie bei der adaptiven Visualisierung geht es bei der Generalisierung darum, die relevanten Daten hervorzuheben, komplizierte Strukturen zu vereinfachen und Unwesentliches wegzulassen. Die Generalisierung von Daten ist nicht nur für eine gute Lesbarkeit

der Karten sehr wichtig, sondern hilft wesentlich bei der Reduzierung der Datenmenge, die übertragen und angezeigt werden muss.

Der Unterschied zwischen adaptiver Visualisierung und Generalisierung besteht darin, dass die Generalisierung eine möglichst gut lesbare Karte für einen bestimmten Massstab zum Ziel hat, adaptive Visualisierung dagegen stellt die Daten der jeweiligen Situation entsprechend dar.

Interaktionen des Benutzers mit dem System gestalten sich auf mobilen Geräten grundsätzlich anders als Interaktionen auf einem Desktoprechner. Während auf Desktop-Computer oft mit sogenannten „mouseover“-Effekten gearbeitet wird, ist dies bei mobilen Geräten nicht möglich, da keine Maus vorhanden ist. Interaktionen mit Karten auf Mobiltelefonen müssen mit Tastatur-Ereignissen ausgelöst werden. Bei PDA mit berührungssensitiven Displays kann ein Ereignis durch ein Klick des Stifts eintreten. In beiden Fällen gilt zu beachten, dass der Nutzer die Interaktionsmöglichkeiten erkennen kann.

2.4 Verwandte Arbeiten

2.4.1 Kartografische Arbeiten über Methana

Die Halbinsel Methana ist erstmals im Jahre 1993 von Prof. Dr. L. Hurni im Rahmen seiner Dissertation mit hoher Genauigkeit vermessen und kartiert worden. In dieser Arbeit untersuchte er die digitale Erstellung einer topografischen und einer geologischen Papierkarte [Hurni, 1995].

Ausgehend von dieser Arbeit gibt es am Institut für Kartografie zwei weitere Arbeiten. In der ersten entwickelt Heiko Mundle ein dreidimensionales Informationssystem der Insel mit Hilfe der Auszeichnungssprache VRML [Mundle, 1999]. In der zweiten Arbeit realisiert Andreas Gauer eine interaktive Webkarte von Methana mit dem Vektorgrafikformat SVG [Gauer, 2006]. Im folgenden wird die Karte von Gauer *interaktive Webkarte* bezeichnet.

Die interaktive Webkarte enthält zur Navigation Zoom- und Verschiebewerkzeuge und eine Referenzkarte. Zur Abfrage von thematischen Informationen enthält die Karte ein Abfragewerkzeug. Die Karte ist streng nach topografischen Ebenen organisiert. So können einzelne Ebenen je nach Interesse des Betrachters ein- bzw. ausgeschaltet werden. Damit wird die Übersichtlichkeit verbessert und der Benutzer weniger von den für ihn interessanten Informationen abgelenkt. Ein Screenshot der interaktiven Karte von Methana ist in Abbildung 1 zu sehen.

2.4.2 Arbeiten und Projekte im Bereich der mobilen Kartografie

Arbeiten im Bereich der mobilen Kartografie zielen vor allem in die Richtungen Navigationssysteme und Location Based Services.

Zur Gruppe der Navigationssysteme gehört die Arbeit von C. Halbach, in welcher er die Konzeption und Entwicklung mobiler Stadtführer zu touristischen Zwecken untersucht [Halbach, 2005]. R. Würth realisiert einen Navigationsassistent für Fussgänger, in dessen

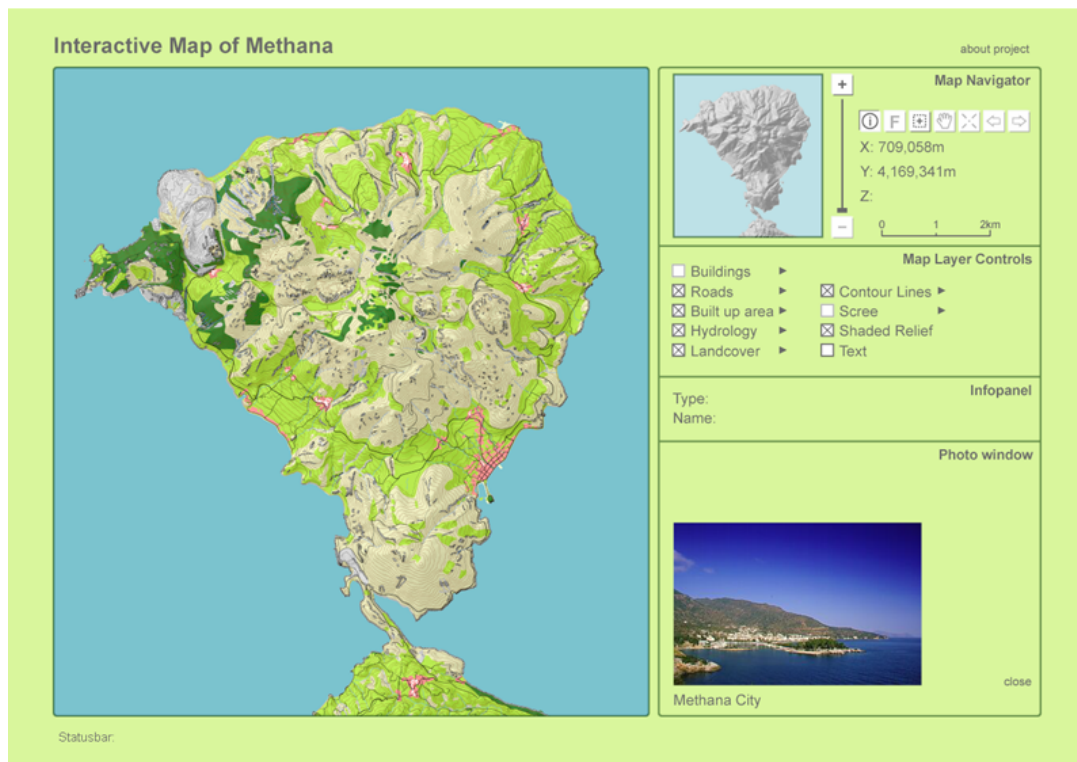


Abbildung 1: Die interaktive Webkarte von Methana von Andreas Gauer

Stadtkarte sogenannte Strukturelemente, anhand derer sich Fussgänger orientieren, betont visualisiert werden [Würth, 2006]. Ein Navigationssystem für Radfahrer entwickelt A. Adam in seiner Arbeit [Adam, 2004].

Die Projekte GiMoDig, Webpark und PARAMOUNT untersuchen vor allem das Thema Location Based Services [gimodig.fgi.fi, 2006] [webparkservices.info, 2006].

PARAMOUNT erforscht wie Bergsteiger und Wanderer mit LBS unterstützt werden können. Das Projekt bietet Dienste in drei Hauptkategorien an: Infotour, Safetour und Datatour.

Die Komponente Infotour besteht im wesentlichen aus dem Abruf ortsgebundener Informationen wie Karten, POI wie Gipfel, Hütten etc., Wetterinformationen und auch Routingfunktionalitäten.

Safetour ist ein Dienst mit sicherheitsrelevanten Funktionalitäten. So können Notfallmeldungen mit Positionsangabe versendet werden, Wanderer in kritischem Gelände zentral verfolgt und Rettungseinsätze unterstützt werden.

Die dritte Komponente Datatour dient dazu, die Datengrundlage laufend aktuell zu halten. So wird der Nutzer direkt in die Datenerfassung und -aktualisierung eingebunden.

Die Kommunikation zwischen dem Client und dem Server erfolgt über GPRS. Die Daten werden mittels XML über das HTTP-Protokoll übermittelt [Sayda et al., 2002].

3 Konzept

3.1 Inhalt

Der Inhalt der Karte soll im wesentlichen derselbe sein wie der Inhalt der interaktiven Webkarte. Dieser umfasst eine genaue topografische Basiskarte und touristische und kulturelle Sehenswürdigkeiten.

Teil der Basiskarte der interaktiven Webkarte sind ein Relief und die Nomenklatur, die beide aber nicht als Vektordaten sondern als Rasterdaten vorliegen. Die Integration dieser Daten muss genauer untersucht werden, falls die Rasterbilder zu langen Ladezeiten führen. Für die Nomenklatur besteht die Möglichkeit, die Schriften zu vektorisieren.

Die Points of Interest (POI) beinhalten ausschliesslich touristische und kulturelle Sehenswürdigkeiten. Sie sollen in der Karte deutlich erkennbar sein und dem Benutzer auffallen. Dazu muss sich deren Symbolisierung in genügendem Masse von der Basiskarte abheben, ohne dass die Symbole ein überladenes Kartenbild zur Folge haben. Zur Betrachtung der Basiskarte kann der Benutzer die POI ausblenden.

3.2 Datenorganisation

Für die Organisation der Daten muss eine Lösung gefunden werden, die mit den Einschränkungen der mobilen Geräte möglichst gut zurechtkommt.

Das Ziel der Datenorganisation muss sein, dass die Software zur Darstellung der Daten, im folgenden Viewer genannt, immer nur gerade die Daten aufbereiten und darstellen muss, die angezeigt werden. Zur Reduzierung der Rechenzeit soll verhindert werden, dass der Viewer Daten aufbereitet, die ausserhalb des sichtbaren Kartenausschnitts liegen. Aus gleichem Grund sollen nur Daten dargestellt werden, die der Benutzer erkennen kann. Daten deren Darstellung kleiner als die Minimaldimensionen sind oder das Kartenbild unübersichtlich machen, sollen weggelassen werden.

Um dieses Ziel zu erreichen werden die beiden Techniken Kachelung und adaptive Visualisierung eingesetzt.

Zur Realisierung von adaptiver Visualisierung muss eine Karte in unterschiedlichen Stufen generalisiert werden. Für die vorliegende Arbeit soll die Karte der Halbinsel Methana in drei Stufen generalisiert werden. Eine stark generalisierte Version soll eine Gesamtübersicht über die Insel ermöglichen. Will der Benutzer Teilregionen der Insel betrachten, steht ihm eine weniger stark generalisierte Karte zur Verfügung. Zum Betrachten von einzelnen Ortschaften oder Objekten wird die schwach generalisierte Karte verwendet. Der Informationsgehalt dieser Karte entspricht demjenigen der interaktiven Webkarte.

Bei der Kachelung wird die Karte in gleich grosse Teilbereiche unterteilt. Die stark generalisierte Karte besteht aus nur einer Kachel, die mittel generalisierte Karte hat Kachelgrössen von $2750\text{ m} \times 3100\text{ m}$. Dies ergibt für das gesamte Gebiet 16 Kacheln. Die schwach generalisierte Karte hat Kacheln von $1000\text{ m} \times 1000\text{ m}$, was 143 Kacheln ergibt.

Je nach Zoomstufe und Koordinatenausschnitt werden die vier entsprechenden Kacheln

geladen und angezeigt. Sobald der aktuelle Ausschnitt kleiner als die Kachel der nächstgrösseren Zoomstufe ist, werden die nächsten, weniger generalisierten Kacheln geladen.

3.3 Kartenentwurf

Um das Display des Gerätes möglichst maximal für die Karte ausnützen zu können, sollen Kartenrandangaben wie Titel und Legende sowie die Bedienungselemente für die Karte ein- und ausgeblendet werden können, so dass der Benutzer bei Bedarf die nötigen Menüleisten einblenden kann.

Dazu sollen zwei Menüleisten erstellt werden. Ein Menü am oberen Rand der Karte enthält den Titel sowie Untertitel. Das zweite Menü wird am unteren Kartenrand platziert und beinhaltet die Legende und Bedienungselemente. Letzteres Menü soll wenn nötig mit mehreren Reitern ausgestattet sein. Der Benutzer kann so die untere Menüleiste einblenden und den entsprechenden Reiter anwählen.



Abbildung 2: Kartenentwurf

3.4 Technik

Die interaktive Karte wird lokal auf dem mobilen Gerät gespeichert, auf eine Client-Server-Architektur wird verzichtet. Heutige PDA verfügen über ausreichend Speicherplatz, so dass auch grosse Datenmengen lokal gespeichert werden können. Der Vorteil der lokalen Speicherung besteht vor allem darin, dass die Übertragung der Daten über das Mobilfunknetz wegfällt und somit für den Nutzer keine weiteren Kosten anfallen. Bei einer Client-Server-Architektur, wo regelmässig eine Verbindung zwischen dem Benutzer und dem Server hergestellt wird, muss die Verfügbarkeit des Mobilfunknetz hoch sein. Für die Halbinsel

Methana ist aber unklar, wie gut sie mit einem Mobilfunknetz versorgt ist. Mit einer Client-Server-Architektur können jederzeit aktuelle Daten zur Verfügung gestellt werden, was ein wichtiger Vorteil gegenüber der lokalen Speicherung ist.

Die grafische Darstellung der Karte soll von hoher Qualität sein, ohne dass die Datenmenge zu gross wird. Weiter sollen anspruchsvolle Interaktionen realisiert werden können. SVG, eine Beschreibungssprache für Vektordaten, erfüllt diese Anforderungen am besten. So können in SVG-Karten mit ECMAScripts Interaktionen implementiert werden.

4 Grundlagedaten und Techniken

4.1 Grundlagedaten

Die Grundlagedaten werden von der Arbeit von Gauer übernommen [Gauer, 2006]. Diese Daten umfassen die Bodenbedeckung, ein Relief und Höhenkurven, Bäche, Strassen, Siedlungsgebiete und Gebäude wie auch Einzelobjekte und die Nomenklatur.

Sämtliche Daten mit Ausnahme des Relief und der Nomenklatur liegen als Vektordaten in Shape-Dateien, einem GIS-Format für Geodaten, vor. Das Relief und die Nomenklatur sind als Rasterbilder vorhanden.

Weiter stehen der SVG-Quellcode sowie sämtliche für die Interaktion verwendeten ECMASkripts zur Verfügung. Dazu gibt es zahlreiche Bilder jeder Ortschaft und sehenswürdiger Kirchen und Kapellen. Diese bilden die Grundlage für die POI der mobilen Karten.

Die Daten liegen in der UTM-Projektion Zone 34N vor, da wie Gauer schreibt „Wanderer vermehrt mit GPS Geräten unterwegs sind, soll die Karte in der auf dem WGS84 (*Anm. World Geodetic System*) Datum basierten UTM Projektion vorliegen“ [Gauer, 2006]. Dies ist für die vorliegende Arbeit von Vorteil, weil so bei der Umrechnung von GPS-Koordinaten in die lokalen Landeskoordinaten kein Ellipsoidübergang nötig ist.

4.2 Scalable Vector Graphics (SVG) und ECMAScript

SVG ist eine XML-basierte Auszeichnungssprache zur Beschreibung von zweidimensionalen Vektorgrafiken. SVG ist die Abkürzung von Scalable Vector Graphics und ist ein Standard des *World Wide Web*-Konsortiums (W3C). Aktuell ist die Spezifikation 1.1 [w3.org, 2006].

Für die Visualisierung von Vektordaten auf Geräten mit stark limitierter Leistung wie Mobiltelefone und PDA hat das W3C-Konsortium die SVG Mobile Empfehlung erlassen. SVG Mobile besteht aus den beiden Profilen SVG Tiny und SVG Basic, welche beide auf der SVG-Spezifikation 1.1 basieren. SVG Tiny ist in erster Linie für Mobiltelefone entwickelt worden, wogegen SVG Basic sich vor allem für PDA eignet. Die verschiedenen Profile sind streng aufwärtskompatibel, d.h. SVG Tiny ist eine Untermenge von SVG Basic, welches wiederum genau eine Untermenge der SVG Full Spezifikation ist. So ist gewährleistet, dass ein Viewer welcher die SVG Full Spezifikation erfüllt beide SVG Mobile Profile interpretieren kann. Ebenfalls kann eine Applikation, die SVG Basic versteht, eine SVG Tiny Grafik

korrekt darstellen. Die genauen Spezifikationen, d.h. welche Anweisungen SVG Tiny bzw. Basic verstehen, ist auf [w3.org, 2006] tabellarisch aufgeführt.

ECMAScript ist eine Skript-Sprache von *ECMA International*, einem internationalen Interessenverband von Firmen aus der Industrie, der Normen im Bereich der Kommunikationstechnik und Unterhaltungs- und Haushaltselektronik erarbeitet [ecma-international.org, 2006].

Ein SVG-Dokument ist als eine Baumstruktur aufgebaut. Jeder SVG-Tag ist ein Knoten dieses Baums. Diese interne Repräsentation eines Dokuments wird *Document Object Model*, kurz DOM, genannt [Flanagan, 2003]. Zur besseren Anschaulichkeit des Dokumentenbaums werden die Knoten oft auch als Kind- bzw. Elternelemente bezeichnet. Zum Beispiel ist eine Gruppe mit zwei Rechtecken ein Elternelement der beiden Rechtecke. Die beiden Rechtecke hingegen sind Kindelemente der Gruppe.

In ECMAScript wird jeder Knoten des DOM als Objekt dargestellt. So können Skripte auf jedes SVG-Element zugreifen und dessen Inhalt und Attribute verändern. Auf diese Weise werden Interaktionen implementiert.

Die beiden Spezifikation SVG Full und SVG Basic unterstützen das Scripting mit ECMAScript.

4.3 Das Satellitennavigationssystem GPS

NAVSTAR-GPS (NAVigation System width Time And Ranging - Global Positioning System), kurz GPS genannt, ist ein weltweit verfügbares Satellitennavigationssystem des US-amerikanischen Verteidigungsdepartement.

GPS, welches anfänglich für militärische Zwecke eingeführt wird, wird heute vorwiegend und in grossem Umfang zivil für die Positionsbestimmung und Navigation genutzt.

Der Aufbau von GPS besteht konzeptionell aus drei Segmenten. Das Raumsegment besteht aus 24 operationellen Satelliten und einige Reservesatelliten. Das Kontrollsegment umfasst Monitorstationen, die die Distanzen zu den Satelliten messen. Aus diesen Messungen werden die genauen Bahndaten der Satelliten berechnet und diesen übermittelt. Zum Nutzersegment gehören alle, die einen GPS-Empfänger zur Navigation auf See und in der Luftfahrt, zur geodätischen Vermessung, zur Autonavigation etc. einsetzen. Die Anwendungen von GPS sind sehr vielseitig.

Die Signalstruktur von GPS basiert auf den zwei Trägerwellen L1 und L2. Diesen sind Navigationssignale (Codes) und Navigationsnachrichten (Messages) überlagert. Auf der Welle L1 sind der C/A- und der P-Code moduliert und auf L2 nur der P-Code.

Der C/A-Code ist ein allgemein zugänglicher Code, der eine Positionsbestimmung bei eingeschränkter Genauigkeit ermöglicht. Diese Auswertung wird *Code-Lösung* genannt. Hingegen ist der P-Code verschlüsselt und somit nicht frei zugänglich. Dieser Code bleibt also einem eingeschränkten Nutzerbereich, vor allem dem US-Militär, vorbehalten.

Auf die Trägerwellen werden im weiteren noch eine Navigationsnachricht aufmoduliert,

welche eine 30 Sekunden lange Sequenz ist. Die Nachricht enthält die Almanachdaten, welche alle Satellitenbahnen und den aktuellen Systemstatus beinhaltet.

GPS-Empfänger für einfache Navigationsaufgaben sind meist Einfrequenzempfänger, die nur das Signal L1 verarbeiten und also eine Code-Lösung finden.

Dagegen werden für geodätische Anwendungen mit hohen Genauigkeitsanforderungen Zweifrequenzempfänger eingesetzt. Solche Anwendungen erfolgen mit differentiellen Methoden mit Messung der Phasenverschiebung.

GPS-Koordinaten sind geozentrisch und basieren auf dem WGS84-Ellipsoid. WGS84 steht für *World Geodetic System 84*. Oftmals will der Nutzer aber Koordinaten im jeweiligen Landessystem. Dazu braucht es zuerst einen Ellipsoidübergang, d.h. eine Transformation der geozentrischen Koordinaten vom WGS84- zum Zielellipsoid. Die geozentrischen Koordinaten müssen anschliessend auf geografische Koordinaten und dann in Landeskoordinaten umgerechnet werden [Ingensand, 2004].

4.4 UTM-Projektion

Die *Universal Transversal Mercator*-Projektion, kurz UTM, ist eine transversale, winkeltreue Zylinderprojektion, welche seit 1947 weltweit eingesetzt wird.

Die Erde ist in 60 Zonen von je 6° Breite eingeteilt. Diese Zonen sind von 1 bis 60 durchnummeriert, beginnend beim Meridian 180° westlich von Greenwich. Die UTM-Projektion wird nur von 80°N bis 80°S verwendet. In den Polzonen wird eine stereografische Projektion angewandt. Als Bezugsdatum sind verschiedene Ellipsoide möglich. Im Zusammenhang mit GPS ist es sinnvoll, das WGS84-Ellipsoid zu wählen. Auf diese Weise entfällt der Ellipsoidübergang.

In der UTM-Projektion werden kartesische Koordinaten verwendet, die bei bekannter Zonennummer und Hemisphäre, eindeutig sind.

Der Rechtswert, auch *Easting* genannt, besteht aus der Distanz in Metern vom Mittelmeridian der jeweiligen Zone zum gesuchten Punkt. Damit es bei westlich des Mittelmeridians gelegenen Punkten keine negativen Koordinaten gibt, wird zum Rechtswert 500'000 addiert. Der Hochwert (*Northing*) ist der nördliche Abstand des Punktes vom Äquator in Metern. Um in der südlichen Hemisphäre negative Hochwerte zu verhindern, wird auf der Südhalbkugel zum Hochwert 1'000'000 dazugezählt [Elmiger et al., 2003] [Snyder, 1982].

4.5 Evaluation eines mobilen Geräts

Zur Evaluation eines Testgerätes für diese Arbeit wird eine Kriterienliste erarbeitet. Die Kriterien und deren Gewichtung sind in Tabelle 1 ersichtlich.

Das erste und wichtigste Kriterium für ein Testgerät ist ein integrierter GPS-Empfänger. Ist der GPS-Empfänger bereits im Gerät eingebaut, erübrigt sich das Anschliessen einer externen GPS-Antenne. Es ist anzunehmen, dass in naher Zukunft viele mobile Geräte mit GPS-Empfängern ausgerüstet werden. Dies ist ein weiterer Grund, dass für diese Arbeit mit einem solchen Gerät gearbeitet werden soll.

Kriterium	Gewichtung
integrierter GPS-Empfänger	wichtig
Display: Grösse, Kontrast, Farbe	wichtig
<i>touch-screen</i> mit Stift	wichtig
Steckplätze für Speicherkarten	wichtig
Verfügbarkeit	wichtig
Prozessor	mittel
Speicher	mittel
integrierte Kamera	optional

Tabelle 1: Kriterienliste zur Evaluation eines Testgerätes

Damit die Karte möglichst gut lesbar ist, muss das Display im Verhältnis zum Gerät möglichst gross sein. Zu berücksichtigen ist ebenfalls der Kontrast wie auch die Anzahl Farben.

Das Gerät soll über ein berührungssensitives Display verfügen, um so die Interaktionen mit der Karte zu ermöglichen.

Ebenfalls soll das Gerät Steckplätze für Speicherkarten bieten. So kann der interne Speicher erheblich erweitert werden, falls das nötig ist. Diese beiden Kriterien sind in heutigen PDA praktisch ausnahmslos erfüllt.

Ein weiteres wichtiges Kriterium ist die Verfügbarkeit. D.h. das Gerät muss innerhalb von wenigen Tagen beschafft werden können.

Da das Darstellen von grossen SVG-Grafiken erfahrungsgemäss rechenintensiv ist, soll das Gerät einen möglichst schnellen Prozessor wie auch einen vergleichsweise grossen internen Speicher.

Eine integrierte Kamera ist kein ausschlaggebendes Kriterium. Wenn aber der ausgewählte PDA über eine integrierte Kamera besitzt, wäre es interessant, neu aufgenommene Bilder zusammen mit der aktuellen Position als POI zu speichern. Die Bilder würden dann wieder als Information über einen POI angezeigt.

Wie sich bei der Evaluation herausstellt, ist der Markt von PDA relativ klein. Insbesondere gibt es nur paar wenige Geräte mit integriertem GPS-Empfänger, die in der Schweiz verfügbar sind.

Die folgenden Geräte erfüllen die Anforderungen am besten:

- Acer n35se Pocket PC
- Asus MyPal A636 Pocket PC
- Fujitsu-Siemens Pocket LOOX N520
- HP iPAQ hw6516

GPS	integriert
Display	3.5 Zoll <i>touch-screen</i> 240x320 Pixel, 65.536 Farben
Erweiterungs- steckplätze	SD Karte
Prozessor	<i>Intel PXA270</i> 312MHz basierend auf <i>Intel XScale</i> Mikroarchitektur
Speicher	Systemspeicher (RAM) 64 MB Flashspeicher (ROM) 128 MB
Betriebssystem	<i>Microsoft Windows Mobile 5.0</i>

Tabelle 2: Die Eigenschaften des Modells *Pocket Loox N520*

Das Gerät von HP verfügt zwar über eine integrierte Kamera, doch wird es aufgrund der kleineren Displaygröße nicht gewählt. Die restlichen drei Geräte sind praktisch gleichwertig. Der Entscheid fällt auf das Modell Pocket LOOX N520 von Fujitsu-Siemens, weil dieses Gerät am schnellsten verfügbar ist. Die wesentlichen Eigenschaften sind in der Tabelle 2 zu finden.

Der GPS-Empfänger des Gerätes ist ein *Sirf Star 3*. Dieser Chip wird in zahlreichen mobilen GPS-Empfängern verwendet. So setzt z.B. auch Garmin auf die Technologie der Firma Sirf [garmin.com]. Der Sirf Star 3-Chip empfängt den CA-Code der Trägerwelle L1 und berechnet eine Code-Lösung. Der Hersteller gibt als Positionsgenauigkeit weniger als zehn Meter an [sirf.com, 2006].

4.6 SVG-Viewer für mobile Geräte

4.6.1 Übersicht über SVG-Viewer

Zur Darstellung von SVG-Grafiken gibt es für mobile Geräte nicht so zahlreiche Möglichkeiten wie für Desktop-Computer. Daher muss zu Beginn untersucht und entschieden werden, wie die SVG-Karte auf dem PDA dargestellt werden soll.

Die eine Möglichkeit ist einen SVG-Viewer zu finden, welcher mindestens das SVG Basic-Profil und somit Scripting unterstützt.

Da heutige mobile Geräte oftmals über eine Java Virtual Machine (JVM) der Java Micro Edition Platform (J2ME) verfügen, ist eine zweite, aufwändigere Möglichkeit das Programmieren einer eigenen Java-Applikation mit bestehenden Bibliotheken zur Darstellung von SVG.

Wie sich herausstellt trifft dies es auf das Pocket LOOX-Gerät nicht zu. Eine Java Virtual Machine lässt sich zu installieren. Es wird nach einer Java-Umgebung wie auf Desktop-Rechnern üblich, also Java 2 Standard Edition (J2SE), gesucht. Es gibt zwar Projekte zur Entwicklung einer J2SE-Umgebung für mobile Betriebssysteme, doch sind die entweder zu wenig ausgereift oder benutzen eigene Bibliotheken, so dass z.B. die Batik-Bibliothek nicht verwendet werden kann.

	SVG-Spezifikation	ECMA Script	OS	Erhältlich	Internet
BitFlash	Tiny 1.1 & 1.2	Ja	div. für Mobiltelefone	für Hersteller	www.bitflash.com
eSVG	Basic 1.1	Ja ¹	Windows	Gratis Evaluations Paket	www.esvg.com
Ikivo	Tiny 1.1 & 1.2	Nein	div. für Mobiltelefone	für Hersteller	www.ikivo.com
OpenWave	Tiny 1.2	Ja	div. für Mobiltelefone	für Hersteller	www.openwave.com
SVG-Viewer für mobile Endgeräte	SVG ^{±geo}	Ja ²	Windows; Symbian; Qt;	frei	[fh-oow.de, 2006]
TinyLine	Tiny 1.1	Nein ¹	Java MIDP 2.0	frei	www.tinyline.com
X-Smiles	Full 1.0	Nein ¹	Java	frei	www.x-smiles.org

Tabelle 3: Vergleich von SVG-Viewer für mobile Geräte gemäss Herstellerangaben, ausser

¹ eigene Untersuchungen am Desktop-Computer

² eigene Untersuchungen auf mobilem Gerät

Ebenfalls wird keine Möglichkeit gefunden, eine Java 2 Micro Edition (J2ME) Umgebung für Mobiltelefone zu installieren. Womit die Erkenntnis resultiert, dass auf Geräten mit *Windows Mobile*-Betriebssysteme keine Java Virtual Machine installiert werden kann. Trotzdem werden im nachfolgenden der Vollständigkeit halber die Java-basierten Viewer aufgeführt.

Damit Interaktionen möglich sind muss ein SVG-Viewer Ereignisse verarbeiten und ECMA-Script-Code einbinden können. D.h. der Viewer muss auf ein Ereignis wie z.B. einen Mausklick bzw. im Falle von PDA einen Klick des Stifts reagieren können. Anschliessend wird eine ECMAScript-Funktion aufgerufen, die den Event verarbeitet und gegebenenfalls auf bestimmte Attribute von Objekten zugreift und diese verändert.

In der Tabelle 3 ist eine Auflistung und Vergleich der aktuellen mobilen SVG-Viewer. Die Tabelle hat keinen Anspruch auf Vollständigkeit und beruht, mit Ausnahme wo es vermerkt ist, auf Herstellerangaben. Viewer, von denen es eine Version für Desktop-Computer gibt, werden zuerst auf dem Desktop-Computer getestet. Erst anschliessend sollen diese auf dem PDA installiert werden, was aber bis auf eine Ausnahme nicht möglich ist.

BitFlash SVG Tiny Player ist ein SVG Viewer für mobile Geräte. Er unterstützt nach

eigenen Angabe SVG Tiny 1.2 und somit auch Scripting Funktionalitäten. Der Viewer ist für diese Arbeit nicht erhältlich [bitflash.com, 2006].

eSVG ist die Abkürzung für *embedded SVG* und implementiert die SVG Mobile und Teile der SVG 1.1 Full Spezifikation. Auf der Webseite steht eine tabellarische Übersicht zur Verfügung welche Sprachelemente unterstützt werden und welche nicht. Insbesondere versteht eSVG ECMAScript, was für Interaktionen unabdingbar ist. Von eSVG gibt es eine eigenständige Viewer-Applikation, den sogenannten **eSVG Viewer** für den Desktop-PC wie auch für mobile Geräte. Daneben kann eSVG in Anwendungen zur Darstellung von SVG-Grafiken verwendet werden.

Das eSVG Paket ist als limitierte Version zur Evaluation frei erhältlich. Das Evaluationspaket kann sowohl für Desktop-Computer wie für mobile Geräte mit dem Windows-Betriebssystem bezogen werden.

Auch nach einigen Versuchen gelingt es nicht, den eSVG-Viewer erfolgreich auf dem PDA zu installieren. Der Installationsvorgang klappt zwar ohne Fehlermeldung, doch lässt sich anschliessend die Anwendung nicht öffnen. Wie im Forum auf [esvg.com, 2006] zu sehen ist, ist dies nicht ein einmaliges Problem. Seitens der Entwickler gibt es keine Reaktionen betreffend den fehlgeschlagenen Installationen. Somit kann eSVG trotz vielversprechender SVG-Unterstützung nicht verwendet werden.

Ikivo SVG Player ist ein SVG-Viewer für Mobiltelefone. Nach eigenen Angaben wird er in zahlreichen, aktuellen Modellen verwendet [ikivo.com, 2006].

OpenWave Mobile Browser ist ein kommerzieller Internet-Browser, der sich vor allem an Hersteller von Mobiltelefone richtet [openwave.com, 2006].

SVG-Viewer für mobile Endgeräte ist eine Entwicklung der FH Oldenburg. Dieser Viewer wird ausführlich in Kap. 4.6.2 behandelt.

TinyLine Die Applikation TinyLine implementiert die SVG Tiny Spezifikation. Von TinyLine gibt es die beiden Produkte TinyLine 2D und TinyLine SVG. Ersteres ist eine J2ME-Klassenbibliothek und kann so in andere J2ME-Projekte integriert werden [tinyline.com, 2006].

X-Smiles ist nach eigenen Angaben ein offener XML-Browser für exotische Geräte. Er unterstützt zahlreiche XML-Formate wie XHTML, XForms, XSL Formatting Objects, X3D aber auch SVG. Zur Anzeige von SVG-Dokumenten verwendet X-Smiles das veraltete CSIRO SVG Toolkit. Auf [x-smiles.org, 2006] wird die Umstellung vom CSIRO SVG Viewer auf die Batik SVG Bibliothek als ein kurzfristiges Ziel angegeben. X-Smiles ist ein Projekt des Telecommunications Software and Multimedia Laboratory der Technischen Universität Helsinki.

Als weitere Möglichkeiten zur SVG-Darstellung werden die mobilen Versionen der beiden frei erhältlichen Internetbrowser *Mozilla Firefox* und *Opera* in Betracht gezogen [mozilla.org, 2006] [opera.com, 2006].

Die Vollversionen beider Applikationen können standardmässig, ohne dass Plug-In installiert werden muss, SVG-Grafiken darstellen. Nach eigenen Untersuchungen stellt sich aber

heraus, dass dies für die mobilen Versionen nicht zutrifft.

Zur Darstellung von SVG-Grafiken im *Microsoft Internet Explorer* muss das Plug-In *SVG Viewer* von *Adobe* installiert werden. Vom diesem Plug-In gibt es zur Zeit keine Version für den *Pocket Internet Explorer*, so dass eine SVG-Darstellung auf diese Weise ebenfalls nicht möglich ist [adobe.com, 2006].

Aufgrund der beschriebenen Probleme bleibt schlussendlich der SVG-Viewer der FH Oldenburg übrig, welcher für die vorliegende Anwendung eingesetzt wird. Dieser wird im nächsten Kapitel ausführlich beschrieben.

4.6.2 SVG-Viewer für mobile Endgeräte der FH Oldenburg

Der SVG-Viewer für mobile Endgeräte ist ein Projekt des Instituts für Angewandte Photogrammetrie und Geoinformatik der Fachhochschule Oldenburg/Ostfriesland/Wilhelmshaven. Nebst der Entwicklung von Viewer für verschiedene Betriebssysteme mobiler Geräte, ist das Ziel des Projekts die Entwicklung eines angepassten SVG-Profil, sogenannt $\text{SVG}^{\pm\text{geo}}$, welches speziell auf die Bedürfnisse der Repräsentation und Verarbeitung von Geodaten passt [fh-oow.de, 2006]. Der Viewer implementiert dieses $\text{SVG}^{\pm\text{geo}}$ -Profil. Der Quellcode des Viewers, kompilierte Versionen für ein Windows Desktop-Betriebssystem und für ein Windows Mobile-System, sowie eine ausführliche Dokumentation ist frei erhältlich [fh-oow.de, 2006].

Die wichtigsten Vorteile dieses Viewers sind der offengelegte Quellcode mit der dazugehörigen Dokumentation und die bereits implementierte Anbindung an einen GPS-Empfänger. Als Nachteil erweisen sich die relativ schlechte grafische Auflösung und die unvollständige Implementierung von ECMAScript, welche zudem nicht dokumentiert wird. Kurvenförmige Linienelemente werden sehr verpixelt und am Rand ausgefranst dargestellt. Ein weiterer Nachteil betrifft die Konfigurationsdatei, welche im Quellcode mit absolutem Dateipfad angegeben werden muss. So muss der Viewer neu kompiliert werden, wenn sich dieser Pfad ändert.

4.6.3 $\text{SVG}^{\pm\text{geo}}$

In [Brinkhoff und Weitkämper, 2004] wird festgestellt, dass sich SVG grundsätzlich für die Visualisierung von Geodaten anbietet. Doch gibt es für den Einsatz von SVG im Geodatenbereich einige Anforderungen, die von der Sprache nicht direkt erfüllt werden. Dazu zählen sie Aspekte der Generalisierung, dynamische Integration von Daten, Integration von Sensordaten und des Ortsbezugs, Anzeige von Sachattributen und die Erfassung von Sach- und Geodaten.

Dazu wird von den Entwickler der FH Oldenburg das $\text{SVG}^{\pm\text{geo}}$ -Profil entwickelt, welches einerseits für die Geodatenverarbeitung unwesentliche SVG-Elemente weglässt, andererseits wichtige Erweiterungen enthält.

Für die vorliegende Arbeit von besonderem Interesse ist die Integration von Sensordaten. Dazu wird die sogenannte Auszeichnungssprache XForms-GI von Brinkhoff und Weitkäm-

per entwickelt. XForms-GI ist eine Erweiterung von XForms. X-Forms dient zur Eingabe von alphanumerischen Daten in Formulare und ist sie eine Weiterentwicklung der Formulartechnik aus HTML [w3.org, 2006].

Mit XForms können Formulare und die Weiterleitung der gesammelten Daten in jedem XML-Dialekt, also auch in SVG, beschrieben werden. Dabei trennt XForms die Darstellung strikt vom Datenmodell.

Die Formulare und deren Darstellung werden aus sogenannten XForms Controls zusammengesetzt. Das Datenmodell wird hingegen in einem XForms Model definiert. *XForms Binding* verbindet schliesslich die in den Kontrollelementen von *XForms Controls* eingebrachten Daten mit den Daten in *XForms Model*.

XForms-GI als Teil von SVG^{±geo} führt weitere Kontrollelemente ein, um oft verwendete Interaktionsmöglichkeiten von Geometrien deklarativ in eine SVG-Karte einbinden zu können. Eines dieser Kontrollelemente ist **sensor**, welches wie in Kap. 5.3.1 in das SVG-Dokument eingebunden wird.

Für die Elemente von XForms-GI wird der Namensraum <http://www.fh-oldenburg.de/xformsgi/2004/> eingeführt [Brinkhoff und Weitkämper, 2004].

4.6.4 Anpassungen des Viewers

Die vorkompilierte Version für Windows Mobile-Betriebssysteme, die die Entwickler zur Verfügung stellen, läuft auf dem mobilen Gerät auf Anhieb.

Doch sollen noch einige Anpassungen und Erweiterungen des Viewers vorgenommen werden. Dazu muss die Entwicklungsumgebung *Mircosoft eMbedded Visual C++* auf dem Desktop-Rechner installiert werden. Diese Umgebung beinhaltet einen Compiler für ARM-Prozessoren, d.h. für mobile Geräte mit dem Windows Mobile-Betriebssystem. Das Projekt sowie die dazugehörigen Bibliotheken können erweitert und anschliessend einzeln kompiliert werden. Diese neu kompilierte Komponente kann einfach auf den PDA geladen werden. Beim Neustart des SVG-Viewers wird diese neuen Komponente sofort verwendet.

Eine nötige Anpassung ist die Änderung des absoluten Pfads der Konfigurationsdatei. Nebst dieser Anpassung wird der SVG-Viewers um die UTM-Projektion erweitert, wie in Kap. 5.3.2 genauer erläutert wird. Eine zweite Erweiterung stellt die Funktion zum Speichern von neuen POI dar, die in Kap. 5.4.12 beschrieben wird.

4.7 Weitere eingesetzte Software

4.7.1 ArcGIS

ArcGIS ist ein weitverbreitetes und umfangreiches Geoinformationssystem von ESRI und umfasst mehrere Software-Produkte für verschiedene Aufgaben im Bereich von Geodaten [esri.com, 2006]. Es wird die Version 9.1 verwendet.

4.7.2 PostgreSQL und PostGIS

PostgreSQL ist eine objektrelationale Datenbank, die frei erhältlich ist. *PostGIS* ist eine Erweiterung zu PostgreSQL zur Unterstützung von geografischen Objekten. Mit dieser

Erweiterung verfügt eine PostgreSQL-Datenbank über räumliche Abfragemöglichkeiten und kann somit als GIS eingesetzt werden. Wie von PostgreSQL ist auch von PostGIS der Quellcode offengelegt [postgresql.org, 2006] [postgis.refractions.net, 2006].

5 Technische Umsetzung

5.1 Datenaufbereitung

5.1.1 Generalisierung

Zur Generalisierung der Karte werden die Grundlagedaten, die in Shape-Dateien vorliegen, verwendet. Die Generalisierung erfolgt mit ArcGIS, von welchem vor allem die Komponenten ArcMap und ArcToolbox verwendet werden.

Die flächenhaften Geometriedaten werden wie folgt generalisiert:

Die Dateien werden in sogenannte Coverage-Dateien konvertiert um dann die Generalisierungsfunktionalitäten der ArcToolbox verwenden zu können.

Aus der ArcToolbox wird zuerst die Funktion *Eliminate* verwendet, mit welcher die kleinsten Flächen gelöscht werden können. Diese Flächen, die danach eigentlich keine Daten mehr enthalten, werden den benachbarten, grösseren Flächen zugeteilt. So wird eine Reduktion der Anzahl von Flächen erzielt. Übrig bleiben nur noch Flächen ab der bestimmten Grösse. Aus diesen Daten werden anschliessend noch sogenannte *Labels* erzeugt, d.h. pro Polygon wird ein Punktobjekt mit den selben Attributen wie das Polygon erzeugt. Dies wird später weiterverwendet, um den generalisierten Flächen diese Attribute wieder hinzuzufügen.

Um die Grösse der späteren SVG-Datei möglichst gering zu halten, wird die Anzahl der Stützpunkte der Polygone und der Linien reduziert. Dies wird wiederum automatisch mit der Funktion *Simplify Line Or Polygon* erreicht. Als Vereinfachungsmethode wird *BEND.SIMPLIFY* gewählt. Diese Methode erhält die Charakteristiken der ursprünglichen Linien besser als die Methode *POINT_REMOVE*, welche auch zur Verfügung stehen würde.

Mit der Reduzierung der Stützpunkte geht die ursprüngliche Topologie sowie die Attributierung verloren. Aus diesem Grund muss die Topologie bereinigt werden. Danach liegen die generalisierten Geometrien wieder als Polygone vor.

Diesen Polygone müssen nun die thematischen Attribute hinzugefügt werden. Dazu wird ein sogenannter *Spatial Join* mit den oben erwähnten Labels durchgeführt. Aus dieser Verbindung geht die generalisierte Shape-Datei hervor.

Aufgrund der zum Teil unbefriedigenden Generalisierung, werden von Hand noch einzelne Flächen gezeichnet. Mehr über kartografische Aspekte der Generalisierung ist in Kap. 5.5.1 zu finden.

Zur Generalisierung der linienhaften Geometriedaten wird einzig eine automatische Reduzierung der Daten durchgeführt. Diese Reduzierung ist direkt auf Shape-Dateien ausführbar, so dass die Konvertierung in eine Coverage-Datei entfällt.

Weiter werden zur Generalisierung Geometrien mit Attributen, die später in der Karte

unwichtig sind, gelöscht. So werden z.B. in der Shape-Datei, die die Strassen beinhaltet, Fusswege und Pfade gelöscht und nur die Haupt- und wichtigsten Nebenstrasse erhalten.

5.1.2 Kachelung

Zur Kachelung werden die generalisierten Shape-Dateien in eine PostgreSQL-Datenbank gespeichert um die Verschneide-Funktionalitäten von PostGIS verwenden zu können. Jede dieser Dateien bildet in der Datenbank eine Tabelle.

Zuerst wird die Kachel, d.h. der Koordinatenausschnitt, als Polygon definiert. Damit es beim späteren Zusammenfügen der Kacheln am Rand keine unschönen Effekte gibt, wird das Polygon mit einer gewissen Überlappung bestimmt. Dann wird die vollständige räumliche Abfrage gebildet. Diese Abfrage verschneidet die Daten und gibt sie als SVG-Code zurück.

Für einige Tabellen werden nebst der Geometrie gewisse Attribute abgefragt. So können die Geometrien gruppiert und symbolisiert werden. Für jede Kachel wird das Resultat anschliessend als SVG-Dokument in eine eigene Datei geschrieben.

5.2 Aufbau der SVG-Anwendung

Die interaktive Karte besteht aus der sogenannten Initialdatei und mehreren einzelnen SVG-Dateien. Die Initialdatei besteht hauptsächlich aus den ECMAScripts, den Points of Interest, der Nomenklatur, den Platzhaltern für die Kacheln mit den Karteninhalte sowie den Elementen für die Menüleisten.

Die Nomenklatur kann nicht wie bei der interaktiven Webkarte als Rasterbild eingebunden werden, da der SVG-Viewer keine Transparenz von PNG oder GIF darstellen kann.

Es ist eine Eigenschaft des SVG-Viewers, dass er auf Skripts in externen Dateien nicht zugreifen kann, dagegen können eingebundene SVG-Dokumente auf die Skripts in der Initialdatei zugreifen. Aus diesem Grund stehen sämtliche Skripts in der Initialdatei, was lässt die SVG-Datei stark wachsen lässt. Jedoch sind keine Nachteile gegenüber dem Einbinden von externen ECMAScript-Dateien ersichtlich.

Die Nomenklatur und die POI müssen direkt in der Initialdatei stehen, damit die Skripts auf die Attribute der Elemente zugreifen können. Ein solcher Zugriff ist nötig, damit die Schriftgrösse und die Grösse der POI-Symbole abhängig der Zoomstufe mitskaliert werden können. So sind die Schrift bzw. die Symbole der POI gegenüber dem Display immer gleich gross ist. Im Kap. 5.5.2 wird ausführlicher auf die Platzierung der Schrift und auf das Setzen der griechischen Schrift eingegangen.

Die SVG-Dokumente mit dem Karteninhalt werden je nach Zoomstufe und Koordinatenausschnitt dynamisch in die dafür vorgesehenen Platzhalter eingefügt. Diese sind getrennt nach ihrer Generalisierungsstufe in unterschiedlichen Verzeichnissen geordnet. In einem weiteren Verzeichnis sind sämtliche Bilder der POI zu finden.

5.3 Einbindung der GPS-Koordinaten

5.3.1 Einbindung der GPS-Koordinaten in das SVG-Dokument

Zur Einbindung von GPS-Daten muss die Initialdatei um mehrere Namensräume erweitert werden. In die Initialdatei wird ein XForms-GI Kontrollelement eingefügt. Dieses Element greift auf eine Gruppe zu, welche den GPS-Cursor symbolisiert. Im Hauptdokument definiert ein XForms Modell die Struktur der einzugebenden Daten. Nebst dieser braucht es Angaben zum Bezugssystem, damit die im WGS84-System empfangenen GPS-Koordinaten in gewünschte System transformiert werden können. Zur eindeutigen Identifikation der Projektionssysteme wird die Kodierung der *European Petroleum Survey Group* (EPSG) verwendet [epsg.org, 2006]. Der Quellcode, der zeigt wie die GPS-Koordinaten in das SVG-Dokument eingebunden werden, ist in Anhang A zu finden.

5.3.2 Erweiterung des SVG-Viewers um UTM-Projektion

Bis anhin sind im SVG-Viewer die folgenden EPSG-Codes implementiert:

4326	WGS84
31466- 31469	Deutsches Hauptdreiecksnetz (DHDN) / Gauss-Krüger Zone 2- DHDN / Gauss-Krüger Zone 5

Da die vorliegende Karte von Methana in der UTM 34-Projektion vorliegt, muss der SVG-Viewer um die UTM-Projektion erweitert werden. In diesem Fall bezieht sich die Projektion auf das WGS84-Ellipsoid, d.h. auf dasselbe Ellipsoid wie die empfangenen GPS-Koordinaten. Dies vereinfacht die Umrechnung von WGS84-Koordinaten zu UTM 34-Koordinaten.

Da die UTM-Projektion der Gauss-Krüger ähnlich ist, können die bereits für die Gauss-Krüger-Projektion implementierten Formeln übernommen und angepasst werden. Die Formeln der Umrechnung werden anhand [Elmiger et al., 2003] auf ihre Richtigkeit überprüft.

5.4 Implementierung der Interaktionen

5.4.1 Vergrössern und Verschieben

Die Interaktionen Vergrössern bzw. Verkleinern (Zoomen) und Verschieben sind Standardfunktionen in heutigen interaktiven Karten. In interaktiven Karten können diese Funktionen grundsätzlich auf zwei verschiedene Arten gelöst werden. Entweder können die Zoom- und Verschiebefunktionen des Viewers benutzt werden oder sie werden mit Hilfe von Skripts realisiert wie das in [carto.net, 2006] vorgeschlagen wird.

Aufgrund der beschränkten ECMAScript-Unterstützung des SVG-Viewers der FH Oldenburg wird entschieden, dass die Zoom- und Verschiebefunktionen des Viewers übernommen werden.

5.4.2 Platzierung der Menüleisten

Wie in Kap. 5.4.1 beschrieben, werden die Zoom- und Verschiebefunktionen des Viewers übernommen. Dies bringt den Nachteil mit sich, dass die Kartenrandangaben, in diesem Fall also die Titel- und Bedienungsleiste, ebenfalls vergrößert bzw. verkleinert und verschoben werden. Damit die Menüleisten also jederzeit sichtbar sind, müssen diese nach jeder Zoom- und Verschiebe-Aktion neu platziert werden. Dazu muss nach jeder Aktion ein entsprechendes Skript aufgerufen werden. Für die Zoom-Aktionen ist die im Gegenteil zu den Verschiebe-Aktionen möglich.

Alle Element, die neu platziert werden müssen, werden in einer Gruppe zusammengefasst und gemeinsam skaliert und verschoben.

5.4.3 Laden der Kacheln

Abhängig der Zoomstufe werden die Kacheln der entsprechenden Generalisierungsstufe geladen. Dazu wird nach jeder Zoom-Aktion eine Funktion aufgerufen, welche aufgrund der Höhe und Breite des aktuellen Kartenausschnitts die Generalisierungsstufe und die nötigen Kacheln bestimmt. Dazu wird auch die Platzierung der Kacheln berechnet werden. Die Dateinamen der Kacheln werden den Platzhaltern in der Initialdatei übergeben und eingeblendet. Die Platzhalter für die Karten der anderen Generalisierungsstufen werden ausgeblendet.

5.4.4 Anzeige von Informationen über POI

Bei einem Klick auf einen Point of Interest sollen Informationen über diesen erscheinen. Vorläufig liegen für die POI nur ein Bild sowie dessen Name vor.

Es wird ein Fenster erstellt, in welchem die Informationen über POI angezeigt werden. In diesem Fenster gibt es je einen Platzhalter für das Bild und den Namen des POI. Ein Klick auf einen POI ruft eine Funktion auf, der als Parameter der Dateiname des dazugehörigen Bildes und der Name des POI übergeben werden. Die Funktion setzt das Bild und den Text in die dafür vorgesehenen Platzhalter und blendet diese ein.

5.4.5 Ein- und Ausschalten der Ebenen

SVG-Elemente, die inhaltlich zusammengehören, wie die Nomenklatur und die POI werden je in einer Gruppe zusammengefasst. Eine solche Gruppen kann als eine thematische Ebene aufgefasst werden. Alle Elemente einer solchen Ebene können dann zusammen angesprochen und verändert werden. Beim Ein- und Ausschalten der Nomenklatur bzw. der POI wird die entsprechende Gruppe angesprochen und aus dem DOM entfernt.

Wird eine dieser beiden Ebenen eingeschaltet, müssen zusätzlich die Symbole für die POI bzw. die Schriftgrösse skaliert werden, damit sie relativ zur Displaygrösse gleich gross bleiben.

5.4.6 Ein- und Ausschalten GPS-Empfänger

Der GPS-Empfänger und somit die Anzeige der eigenen Position auf der Karte kann ein- bzw. ausgeschaltet werden. Beim Start der interaktiven Karte ist der Empfänger ausgeschaltet. Wird der Empfänger eingeschaltet, wird nach der GPS-Initialisierung die momentane Position mit einem Cursor angezeigt.

5.4.7 Anzeige aktueller Position in Koordinaten

Die aktuelle Position kann mit einem Cursor in der Karte angezeigt werden. Der Cursor ist ein roter Kreis mit einem Pfeil, der die Bewegungsrichtung anzeigt. Es können auch die Koordinaten der Position ausgegeben werden.

Dazu greift eine Funktion auf XForms Modell zu, in welchem zur Laufzeit die aktuellen GPS-Koordinaten sind.

5.4.8 Finden von POI

Damit der Benutzer den gewünschten POI auswählen kann, muss ihm eine Liste mit sämtlichen POI präsentiert werden. Da eine Auswahlliste, wie sie auf [carto.net, 2006] vorgeschlagen wird, für den SVG-Viewer zu kompliziert ist, wird eine statische Liste implementiert.

Beim Öffnen der Liste schreibt eine Funktion sämtliche Identitäten der POI in die Elemente der Liste. Je nach dem ob der GPS-Empfänger eingeschaltet ist oder nicht, wird mit einem Klick eine unterschiedliche Funktion aufgerufen. Ist der GPS-Empfänger an, wird eine Linie zwischen der aktuellen Position und dem gesuchten POI gezeichnet. Falls der GPS-Empfänger ausgeschaltet ist, wird der POI mit einem Kreis hervorgehoben.

5.4.9 Finden des nächstgelegenen POI

Die Funktion zum Finden des nächstgelegenen POI kann im Gegensatz zum Finden von beliebigen POI nur durchgeführt werden, wenn der GPS-Empfänger eingeschaltet ist. Zuerst wird die aktuelle Position ermittelt und gespeichert. Anschliessend wird die Distanz zu jedem POI berechnet. So wird die kürzeste Distanz und der dazugehörige POI erhalten. Anschliessend erscheint ein Nachrichtenfenster mit dem Namen und der Distanz zum nächstgelegenen POI aus und eine Linie vom aktuellen Standort zum POI wird gezeichnet.

5.4.10 Erfassen neuer POI

Mit dem Aufnehmen von neuen POI ist gemeint, dass ein neues Element in das DOM eingebunden wird. Dieses Element wird nicht in eine Datei geschrieben, sondern bleibt nur solange im DOM erhalten wie die SVG-Datei geöffnet ist. Das Aufnehmen neuer POI ist also zu unterscheiden vom Speichern, wie das in Kap. 5.4.12 beschrieben wird.

In der SVG-Initialdatei gibt es einen Platzhalter für neu erfasste Elemente. Beim Erfassen eines POI, wird ein neues SVG-Element erstellt, das diesen Punkt repräsentiert. Diesem

Element werden als Attribute die Position und eine Symbolisierung übergeben. Anschliessend wird es zum Platzhalter hinzugefügt.

Damit den neuen POI eigene Namen vergeben werden können, muss der Benutzer einen solchen eingeben können. Da der Viewer keine Tastaturereignisse empfangen kann, muss nach einer anderen Lösung gesucht werden. So wird eigens ein Tastaturfeld in SVG gezeichnet. Jedes Feld löst bei einem Klick ein Ereignis aus, welches eine Funktion aufruft, die der entsprechende Buchstabe in das Textfeld schreibt. Dabei liest die Funktion den bestehenden Inhalt des Textfeldes ein, und hängt den neuen Buchstaben am Ende an. Wird die `BackSpace`-Taste gedrückt, wird ebenfalls der Inhalt des Textfeldes eingelesen, aber anschliessend der letzte Buchstaben abgetrennt.

Beim Setzen des neuen POI wird dann das Textfeld eingelesen und dessen Inhalt als ID verwendet. Gibt der Benutzer keinen neuen Namen ein, wird ein Namen aus einer fünfstelligen Zufallszahl generiert.

5.4.11 Erfassen neuer Routen

Eine Route ist ein Linienelement, das aus mehreren Wegpunkten besteht. Beim Erfassen einer neuen Route, wird eine neue Gruppe angelegt. Die Wegpunkte sind Elemente mit der Position als Attribut, welche der Gruppe hinzugefügt werden.

Wie beim Erfassen der POI, wird einer Route ein Name vergeben. Die einzelnen Wegpunkte können dagegen nicht benannt werden.

5.4.12 Speichern neuer POI und Routen

SVG besitzt nur lesenden, aber keinen schreibenden Zugriff auf das Dateisystem. ECMAScripts können aus Sicherheitsgründen nicht auf das Dateisystem schreiben. Da diese Skriptsprache oft in Webanwendungen verwendet wird, wäre die Gefahr zu gross, dass Dateien missbräuchlich verändert und gelöscht würden.

Aus diesem Grund kann das Speichern von POI und Routen nicht mit SVG oder ECMAScripts realisiert werden. Das Schreiben einer Datei muss vom Viewer selber übernommen werden. Dazu muss dieser um eine Funktion erweitert werden.

Im wesentlichen funktioniert die Methode folgendermassen: Zuerst wird ein Zeiger auf das `root`-Element des SVG-Dokuments benötigt. Ist dieser vorhanden, kann im gesamten DOM nach einem Element gesucht werden. In diesem Fall wird nach der Gruppe mit den neuen POI und Routen gesucht. Anschliessend wird überprüft, ob das Element gefunden worden ist. Ist dies nicht der Fall, wird dies in einem Nachrichtenfenster mitgeteilt und die Funktion beendet. Ansonsten wird eine neue Datei angelegt, in welche die Gruppe dann geschrieben wird. Zum Schluss wird noch ein Nachrichtenfenster angezeigt zur Bestätigung für den Benutzer.

Die vollständige Funktion ist im Anhang B wiedergegeben.

Die geschriebene Datei ist eine Text-Datei, die in beliebigen Editoren betrachtet werden kann. Ein Beispiel einer solchen Datei ist in Anhang C zu finden.

5.5 Kartografische Darstellung

5.5.1 Generalisierung

Die automatische Generalisierung in ArcGIS (Kap. 5.1.1) ist nicht unproblematisch. Wie in Kap. 2.3 ausgeführt, geht es bei der Generalisierung darum die charakteristischen und relevanten Daten hervorzuheben und unwesentliches wegzulassen. Die in ArcGIS verwendeten Methoden, insbesondere die beschriebene Funktion *Eliminate* löscht, ohne den Kontext zu berücksichtigen, die Flächen unterhalb der angegebenen Grösse.

Konkret stellt sich das Problem, dass bei einer starken Generalisierung sämtliche Felsflächen gelöscht werden, weil es allesamt kleine Flächen sind. Betrachtet man hingegen diese Felsflächen als Ganzes im Zusammenhang, wird die Form eines Vulkankraters deutlich. Diese charakteristische Geländeform muss bei einer Generalisierung berücksichtigt werden, da sie dem Kartenleser als wichtige Orientierungshilfe im Gelände dienen kann und ausserdem typisch für die Halbinsel Methana ist.

So werden nebst dem Löschen der kleinsten Flächen, mehrere dieser Felsflächen in ArcGIS manuell zu einer kraterförmigen Fläche zusammengefasst. Aufgrund der mangelnden Ortskenntnisse des Autors wird das Zusammenfassen nur beispielhaft gemacht und wird daher einer exakten Generalisierung nicht gerecht.

5.5.2 Nomenklatur

Μεθανα

Abbildung 3: Der Name Methana in griechischer Schrift in SVG codiert.

Wie sich zeigt, kann die griechische Schrift auf dem mobilen Gerät dargestellt werden. Dazu muss jeder Buchstabe einzeln codiert werden. Die Kodierung besteht aus dem Präfix `&#x` und der hexadezimalen Unicode-Nummer, abgeschlossen wird jeder Buchstabe mit einem Semikolon. Zum Beispiel wird der Buchstabe Alpha so codiert: `α`. Die Codierung des Namens Methana sieht dann folgendermassen aus:

`Μεθανα`

Wie die Ausgabe in SVG aussieht ist in Abbildung 3 ersichtlich.

Die Schriftgrösse der Nomenklatur soll sich, wie bereits erwähnt, relativ zum Display nicht ändern.

In SVG müssen einem `<text>`-Element immer die beiden Attribute `x` und `y` gegeben werden, welche die Platzierung des Textes bestimmen. Dabei sind dies immer die Koordinaten der linken unteren Ecke des Textes. Wird die Schriftgrösse skaliert, was ja in der vorliegenden Karte der Fall ist, ändert sich die Textbreite und -höhe, hingegen verschiebt sich der linke, untere Bezugspunkt nicht.

Folglich muss entweder der Bezugspunkt einer Beschriftung immer rechts oberhalb der Ortschaft liegen, damit für den Kartenleser jederzeit offensichtlich ist, zu welcher Ortschaft eine Beschriftung gehört.

Die Platzierung der Schrift rechts oberhalb entspricht zwar den kartografischen Regeln nach Imhof, doch ergibt sich im Falle der vorliegenden Karte nicht in jedem Fall eine optimale Lösung.

Damit die Karte bei kleinerem Massstab nicht unübersichtlich wird, wird die Kartenbeschriftung adaptiv visualisiert. Die vier grösseren Ortschaften werden jederzeit und mit einer grösseren Schrift angezeigt als die weiteren kleineren Orte, die erst bei einem grösseren Massstab erscheinen.

5.5.3 Symbolisierung POI

Die Symbolisierung der Points of Interest (POI) ist in Abb. 4 abgebildet. Sie lehnt sich in der Farbgebung sehr stark an die Farbgebung der weitverbreiteten Informationssymbole, die einen weissen Buchstaben i auf einem blauen Hintergrund zeigen. Damit soll sich für den Kartenleser einen hohen Wiedererkennungseffekt ergeben.

Die Ortschaften werden analog zu den erwähnten Informationssymbole mit einem weissen i symbolisiert. Die Symbolisierung der Kirchen und Kapellen ist der Signatur „Kapelle“ der swisstopo ähnlich [swisstopo, 2003].

Der blaue Hintergrund der Symbole hebt sich sehr gut von der restlichen Karte ab, so dass diese POI vom Leser sehr schnell erkannt werden können.

Die Symbole werden für diese Karte entworfen und erstellt. Dabei werden die Grössen-



Abbildung 4: Symbolisierung der POI Ortschaft, Kirche, Kapelle

verhältnisse so gewählt, dass die Darstellung im SVG-Viewer möglichst gut ist. Zum Beispiel wird das weisse i als Symbol für die Ortschaften im Viewer im Verhältnis viel breiter dargestellt. Beim Symbol für die Kapellen und Kirchen wird der Kreis, auf dem das Kreuz steht, im Verhältnis zu klein gezeichnet.

6 Anwendungsfälle

6.1 Einführung in die Anwendungsfälle

In diesem Kapitel werden verschiedene Anwendungsfälle der interaktiven mobilen Karte vorgestellt. Diese sollen einige Anwendungen aufzeigen wie die Karte eingesetzt werden kann. Die Liste ist nicht abschliessend.

Für jeden Anwendungsfall wird eine Zielgruppe definiert. Mögliche Nutzer und ihre Absichten, ihre Vorkenntnisse etc. werden beschrieben.

Anschliessend wird das Anwendungsmodell erläutert. In diesem wird detailliert erklärt, was der Nutzer als nächstes wissen bzw. machen will. Dazu wird jede Aktion beschrieben, die er dazu durchführen muss.

6.2 Anwendungsfall 1: Tourist will Methana kennen lernen

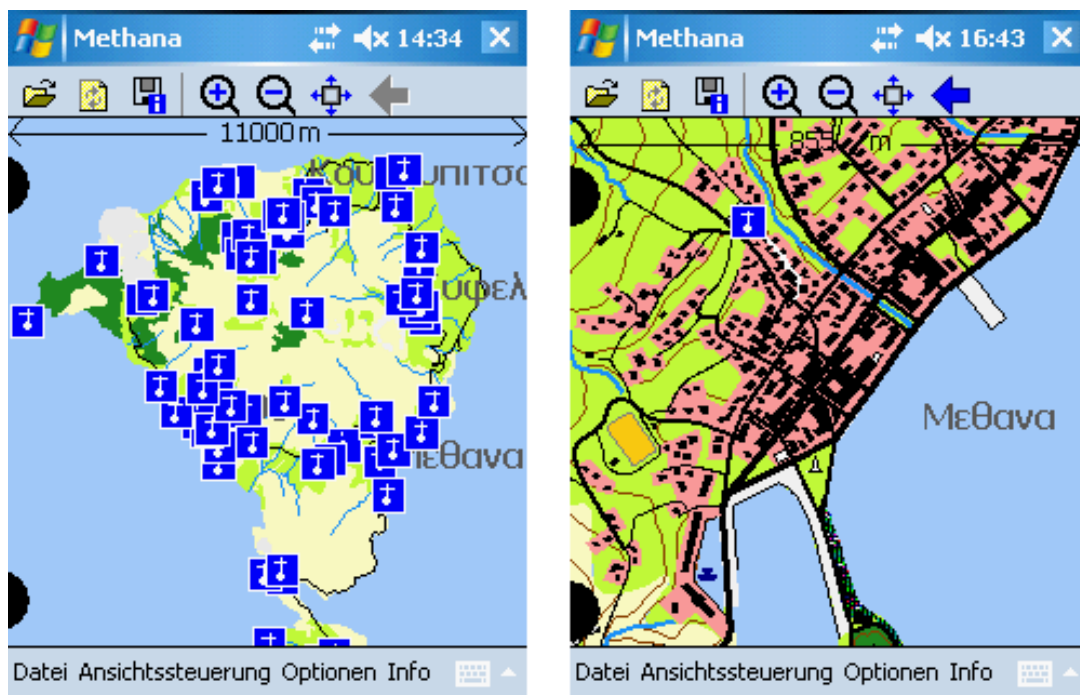


Abbildung 5: Display nach dem Öffnen der Anwendung und ein grösserer Kartenausschnitt von Methana Stadt

Zielgruppe

Zielgruppe dieses Anwendungsfall sind Touristen, die nach Methana in den Urlaub fahren wollen. Im Vorfeld dieses Urlaubs wollen sie die Insel etwas kennen lernen und ihren Urlaub planen.

Anwendungsmodell

Der Tourist will zuerst eine Gesamtübersicht der Insel bekommen. Dazu öffnet er die Anwendung und es erscheint eine Übersichtskarte der gesamten Insel. Sämtliche POI werden angezeigt wie auch die Namen der wichtigsten Ortschaften. Der Benutzer sieht sofort wo die POI liegen, siehe dazu linker Screenshot der Abbildung 5.

Interessiert ihn ein bestimmter Ort wie z.B. Methana, benutzt er die Zoom- und Verschiebe-Funktionen des SVG-Viewers. Mit dem Einzoomen wird ihm eine weniger stark generalisierte Karte zur Verfügung gestellt, und er erkennt nebst den POI weitere interessante Objekte wie Thermalbäder, Denkmäler, Sportplätze etc. [Abb. 5, rechter Screenshot].

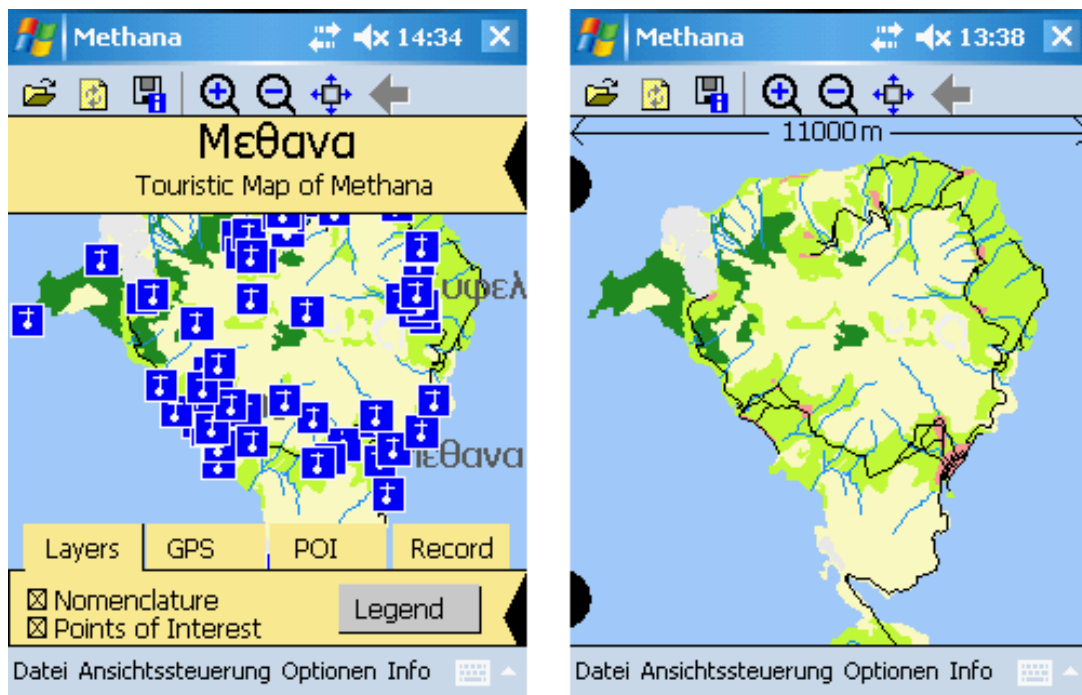


Abbildung 6: Karte mit eingblendeten Titel- und Legendenleisten und Basiskarte

Am linken Rand der Karte sind die beiden Reiter zu erkennen, die mit einem schwarzen, halbrunden Kreis symbolisiert sind. Mit einem Klick des Stifts auf diese Reiter öffnet der Benutzer die Titelleiste bzw. die Menüleiste mit der Legende und den Bedienungselementen [Abb. 6, linker Screenshot].

In der oberen Leiste liest er den Titel und Untertitel der Karte und in der unteren Leiste sieht er die vier Reiter Layers, GPS, POI, Record. Nach dem Öffnen der Anwendung ist immer der Reiter Layers zuvorderst. In diesem Reiter kann er die Nomenklatur und die Points of Interest ein- und ausschalten. Schaltet er diese beiden Themen aus, so erhält er die Basiskarte wie sie in Abbildung 6, rechter Screenshot zu sehen ist.

Der Tourist interessiert sich nun für die Sehenswürdigkeiten auf dieser Insel. In einem

Reiseführer hat er von der Kapelle Ag. Athanassios gelesen, doch weiss er leider nicht, wo die sich befindet. Also öffnet er den Reiter POI und klickt den Button Find POI. Es erscheint eine Liste mit sämtlichen POI. Aus dieser Liste wählt er die erwähnte Kapelle aus [Abb. 7, linker Screenshot].

Nach dem Auswählen schliesst sich die Liste und der gesuchte POI wird mit einem roten Kreis hervorgehoben, so dass der Benutzer den POI sofort findet [Abb. 7, rechter Screenshot]. Vergrössert er den Kartenausschnitt, kann er so anschliessend herausfinden, wie er zu diesem POI kommt.

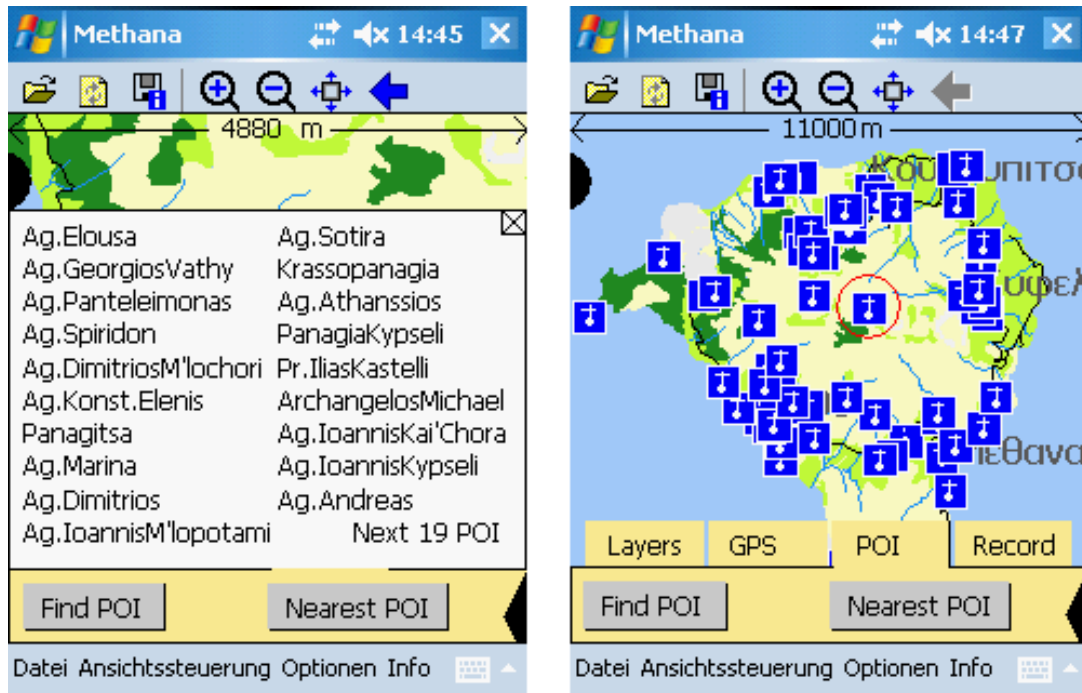


Abbildung 7: Auswählen und Finden eines bestimmten POI

Jetzt kennt der Tourist zwar die genaue Lage des POI und dessen unmittelbare Umgebung, doch weiss er noch nichts über diese Sehenswürdigkeit. Um etwas darüber zu erfahren, klickt er mit dem Stift auf das entsprechende Symbol. Sofort öffnet sich ein Fenster, welches ein Bild wie auch den Namen der Sehenswürdigkeit anzeigt. Handelt es sich beim POI um ein Dorf, wird der Name in griechischer und lateinischer Schrift angezeigt.

Der Benutzer kann das Fenster mit einem Klick auf das Kästchen mit dem Kreuz in der rechten oberen Ecke schliessen [Abb. 8].

6.3 Anwendungsfall 2: Tourist befindet sich auf Methana am Wandern

Zielgruppe

Ein Tourist befindet sich auf der Insel Methana im Urlaub. Er ist ein ungeübter Kartenleser und ist darum froh um Orientierungshilfe. Auf einer Wanderung, die er plant, will er

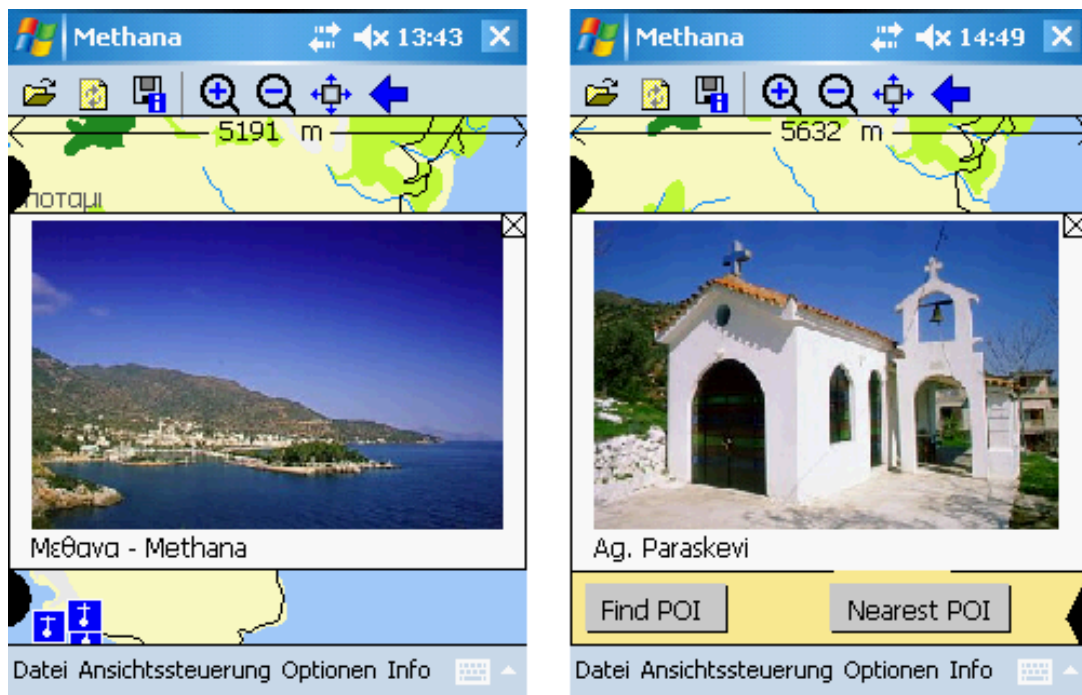


Abbildung 8: Anzeige von Informationen über POI

bestimmte Sehenswürdigkeiten besuchen.

Anwendungsmodell

Der Tourist ist unterwegs auf einer Wanderung. Er schaltet sein mobiles Gerät ein und öffnet die interaktive Karte von Methana. Um zu wissen, wo er sich gerade befindet, öffnet er den Reiter GPS und klickt den Button Run GPS. Nach der Initialisierung des GPS wird seine aktuelle Position und seine Bewegungsrichtung mit einem roten Cursor angezeigt [Abb. 9, linker Screenshot]. Bewegt sich der Benutzer fort, wird die Karte automatisch verschoben, sobald der Cursor in den Randbereich kommt. So bleibt der Cursor innerhalb des aktuellen Kartenausschnitts.

Ebenfalls im gleichen Reiter kann der Benutzer die Koordinaten seiner aktuellen Koordinaten abfragen. Dazu steht die Funktion Position zur Verfügung. Die Koordinaten werden in dem System der Karte, also in diesem Fall in UTM, angezeigt [Abb. 9, rechter Screenshot].

Der Benutzer möchte nun zu einer bestimmten Sehenswürdigkeit gehen. Im Vorfeld seiner Wanderung hat er sich deren Namen notiert. Er öffnet den Reiter POI und klickt den Button Find POI. Eine Liste mit allen POI wird geöffnet. Er sucht aus der Liste den Namen des gesuchten POI. Mit Next 19 POI werden ihm jeweils die nächsten 19 POI angezeigt. Hat er den Namen der Sehenswürdigkeit gefunden, klickt er diesen an. Die Liste wird ausgeblendet und auf der Karte erscheint eine rote Linie zwischen dem GPS-Cursor, d.h. der aktuellen Position, und dem gesuchten POI, die dem Benutzer zeigt, wo dieser POI

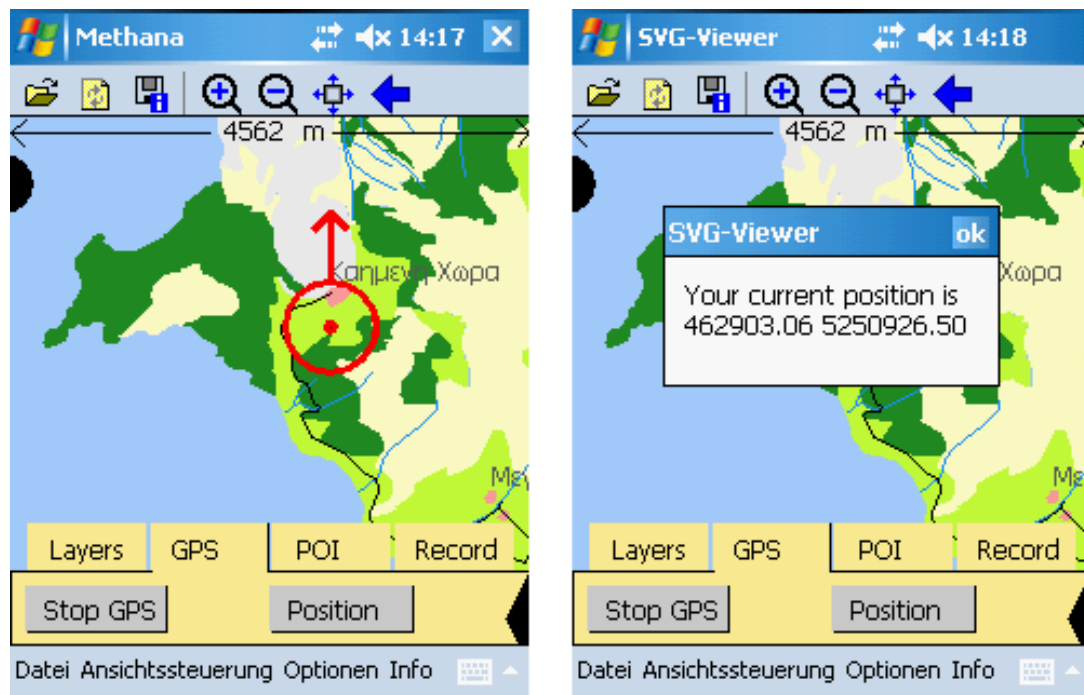


Abbildung 9: GPS Cursor und Koordinaten der aktuellen Position

liegt. So kann er die ungefähre Richtung und Distanz dorthin abschätzen [Abb. 10].

Ein anderes Mal ist der Tourist vielleicht etwas weniger zielgerichtet unterwegs, und er weiss nicht schon im voraus welche Sehenswürdigkeiten er anschauen gehen will. Während er irgendwo am Wandern ist, möchte er in der Nähe einen POI besuchen. Da er nicht weiss, wo der nächste POI liegt und wie dieser heisst, steht ihm die Funktion **Nearest POI** im Reiter **POI** zur Verfügung. Klickt er diese, erscheint ein Fenster mit dem Namen des POI und der Entfernung vom aktuellen Standort. Auf der Karte wird zudem eine rote Linie von der aktuellen Position zu diesem POI gezeichnet. So kann der Benutzer einfach eine Sehenswürdigkeit auffinden [Abb. 11]

6.4 Anwendungsfall 3: Reiseleiter will neue Sehenswürdigkeiten und Wanderungen aufnehmen

Zielgruppe

Ein Reiseleiter aus der lokalen Tourismusorganisation will den Besuchern der Insel neue POI und Routen anbieten können.

Zur Zielgruppe zählen auch Touristen, die bestimmte Orte bzw. Wanderungen, die ihnen besonders gut gefallen haben, aufnehmen und später wieder finden möchten.

Anwendungsmodell

Der Reiseleiter, der neue POI aufnehmen will, geht dorthin, wo dieser POI sich befindet.

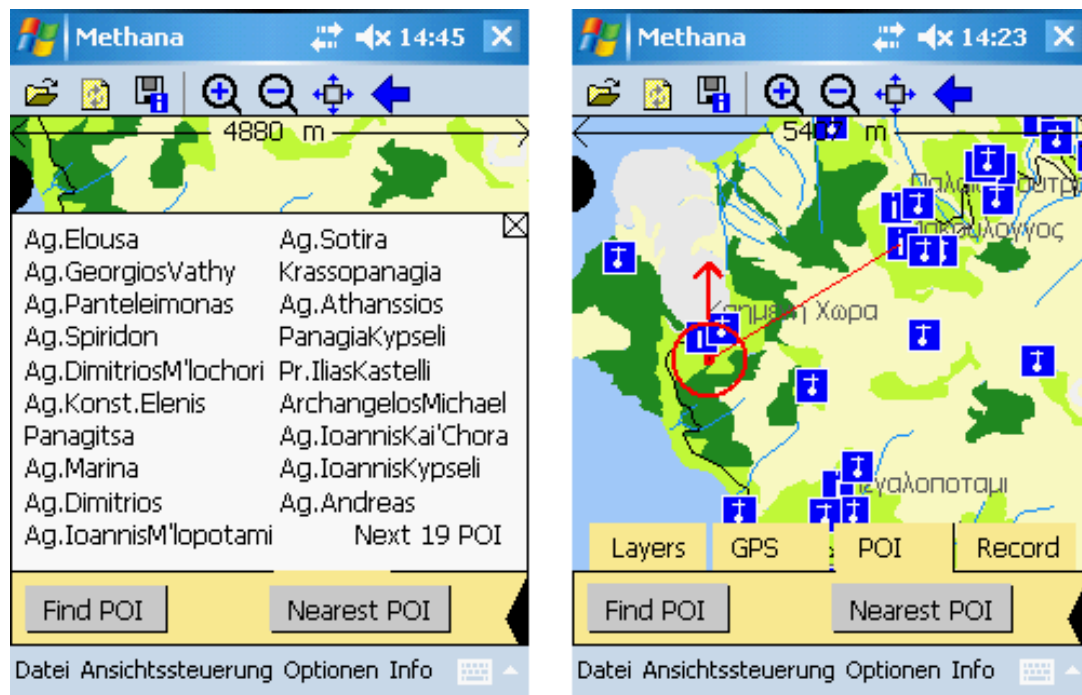


Abbildung 10: Auswählen und Finden eines bestimmten POI mit GPS

Er öffnet die Anwendung und schaltet den GPS-Empfänger ein. Nach der Initialisierung wird ihm die aktuelle Position mit einem Cursor angezeigt.

Er wählt den Reiter Record, wo er zwischen Record Route und Set New POI wählen kann. Da er nur einzelne Sehenswürdigkeiten aufnehmen will, wählt er Set New POI [Abb. 12, linker Screenshot].

Danach öffnet sich ein Fenster, wo der Benutzer den Namen des neuen POI eingeben kann. Wie in Abb. 12, rechter Screenshot zu sehen ist, werden in diesem Fenster ebenfalls die Koordinaten des neuen POI angezeigt. Um später Verwechslungen auszuschliessen, empfiehlt es sich eindeutige Namen zu vergeben. Mit einem Klick auf Set wird der POI in die Karte übernommen, wo er sogleich dargestellt wird. Die Symbolisierung des neuen POI entspricht derjenigen der Ortschaften.

Angenommen, der Reiseleiter will nicht nur einzelne Objekte aufnehmen, sondern will seinen Kunden Wanderungsvorschläge anbieten können. Nachdem der Aufnahme der Wanderung, sollen seine Kunden mit der Anwendung diesselbe Tour begehen können.

Dazu begibt sich der Reiseleiter an den Startort der Wanderung. Die Anwendung läuft bereits und die eigene Position wird angezeigt. Unter dem Reiter Record wählt er die Funktion Record Route. Es folgt einen kurzen Beschrieb, wie neue Routen aufgenommen werden [Abb. 13, linker Screenshot]. In diesem Fenster wählt er anschliessend Begin Rte und es erscheint ein Eingabefeld für den Namen der Route [Abb. 13, rechter Screenshot]. Mit Set wird die neue Route begonnen, d.h. an der aktuellen Position wird der erste Wegpunkt

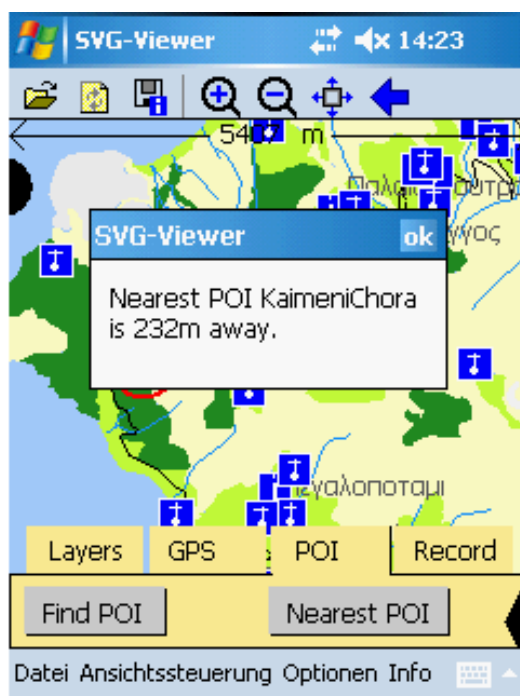


Abbildung 11: Finden des nächstgelegenen POI

gesetzt.

Danach schliesst sich das Fenster und an Stelle der Leiste für Legende und Bedienungselementen erscheint ein weisser Balken [Abb. 14] mit den Funktionen **Begin Rte**, **Set Wpt** und **End Rte**. Der Benutzer kann jetzt seine vorgesehene Route begehen und an Kreuzungen oder charakteristischen Strassenverläufe einen neuen Wegpunkt setzen.

Jedesmal wenn er einen neuen Wegpunkt setzt, erscheint ein Nachrichtenfenster mit dem Hinweis, dass ein neuer Wegpunkt gesetzt worden ist sowie dessen Koordinaten [Abb. 14]. Der Reiseleiter macht dies solange, bis er am Ende seiner Wanderung angekommen ist. Dann beendet er die Route mit **End Rte**, womit der letzte Wegpunkt aufgenommen wird. Anschliessend wird wieder die übliche Leiste am unteren Rand angezeigt. Die aufgenommenen Wegpunkte werden in der Karte als rote Kreise dargestellt.

Damit die neu aufgenommenen Routen beim Beenden der Anwendung nicht verloren gehen, kann der Benutzer diese in eine Datei schreiben. Dazu verwendet er die Funktion **Save POI** des SVG-Viewers. Links der Zoom- und Verschiebe-Schaltflächen befindet sich das entsprechende Icon [Abb. 15, linker Screenshot].

Damit der Benutzer sicher weiss, dass die Speicherung erfolgreich ist, erscheint danach ein Nachrichtenfenster, welches ihm diese bestätigt [Abb. 15, rechter Screenshot].

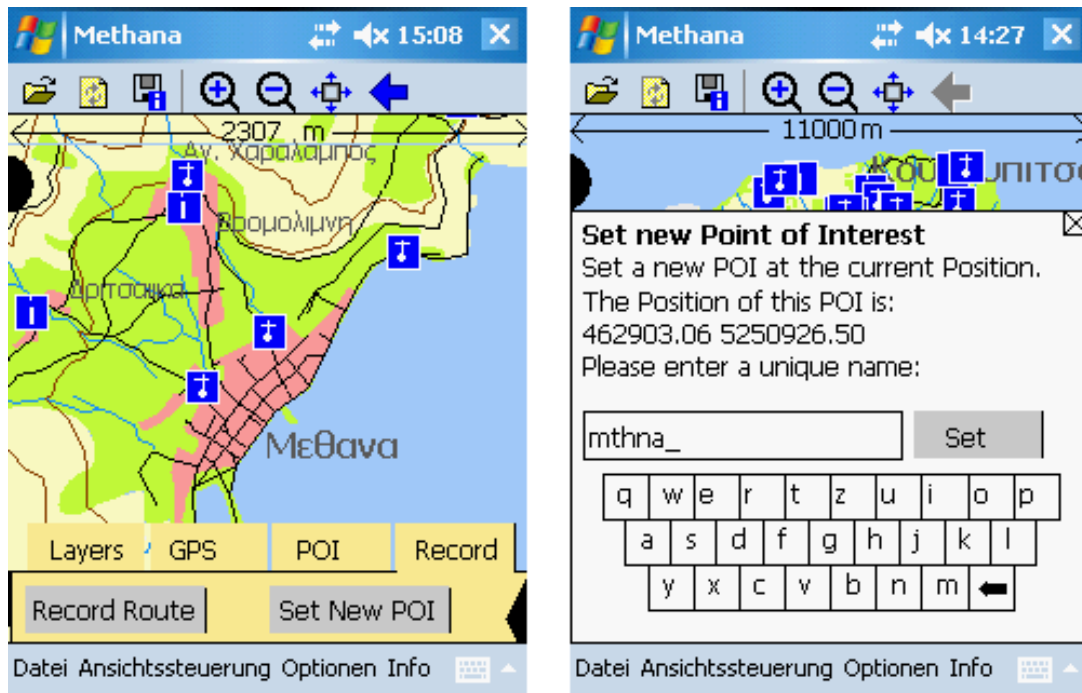


Abbildung 12: Aufnahme von neuen Points of Interest

7 Schlussfolgerung

7.1 Beurteilung der fertigen Anwendung

Die gesetzten Ziele werden bis auf wenige Ausnahmen erfüllt:

- Das Ziel, eine gute Karte mit touristischen und kulturellen Sehenswürdigkeiten zu erstellen, wird erreicht.

Die topografische Basiskarte ist ansprechend und die POI sind so symbolisiert, dass sie sich gut von der Karte abheben.

Die Qualität der grafischen Darstellung ist nicht befriedigend. Wie bereits bei der Evaluation der Viewer feststand, ist diese eine Schwäche des eingesetzten SVG-Viewer. Linien und Kurven sehen verpixelt aus, und die Symbole der POI werden nicht verhältnismäßig angezeigt. Das stört zwar das Erscheinungsbild der Karte, doch verschlechtert sich die Lesbarkeit dank der Einhaltung der Minimaldimensionen nur minim.

Die implementierte adaptive Visualisierung erhöht die Lesbarkeit der Karte auf dem kleinen Display. Drei Generalisierungsstufen, wie in der Karte implementiert, sind eher zu wenig. Vier oder sogar fünf verschiedene Stufen würden bessere Übergänge zwischen den Stufen ergeben.

Beim Öffnen der Anwendung wird eine stark generalisierte Gesamtkarte der Halbinsel gezeigt. Auf diese Weise gewinnt der Benutzer einen Überblick über das Gebiet. Bei starkem Vergrößern der Karte besteht aber die Gefahr, dass der Benutzer die

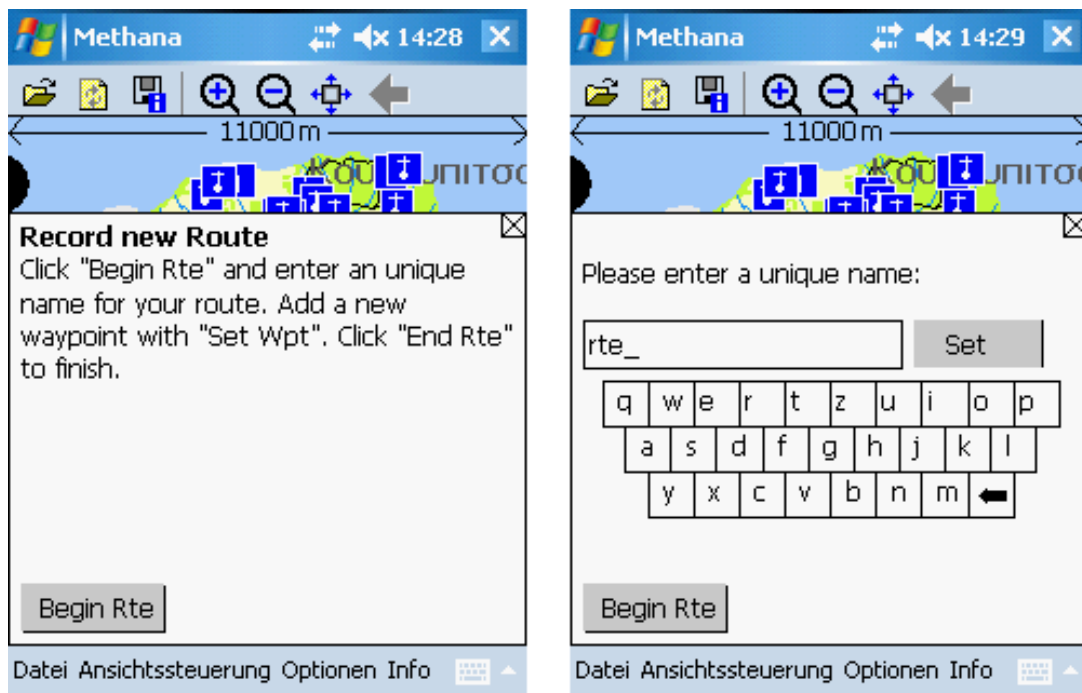


Abbildung 13: Aufnahme neuer Routen

Übersicht verliert. Um dies zu verhindern wäre eine Referenzkarte von Vorteil, die den aktuellen Kartenausschnitt anzeigt.

- Die Einbindung eines GPS-Empfängers in die Anwendung funktioniert. So können die Bestimmung der eigenen Position und deren Anzeige in der Karte realisiert werden. Der rote Cursor, der die aktuelle Position anzeigt, kann nicht skaliert werden. Das hat zur Folge, dass der Cursor beim Vergrössern der Karte auch grösser wird. Dies wirkt sich sehr nachteilig auf das Kartenbild aus. Der Cursor sollte, wie die Symbole der POI, relativ zum Display stets gleich gross sein.
- Das Suchen und Finden von POI ist ein weiteres Ziel, das erreicht wird. Der Benutzer wählt aus einer Liste einen gesuchten POI und kann sich diesen anzeigen lassen.
- Ein Ziel, das nicht vollständig erreicht wird, betrifft die Informationen über die POI. In der jetzigen Karte werden für jeden POI der Name und ein Bild angezeigt. Es können keine ausführlicheren Informationen in Form von Texten oder Tönen abgerufen werden.
- Die Erfassung von POI und Routen kann realisiert werden. Doch fehlen in der jetzigen Karte Möglichkeiten zur Klassifikation neuer POI und Routen sowie zur Erfassung von Informationen über diese.
- Das Ziel, die exportierten Punkte in einem Format zu exportieren, welches von GIS gelesen werden können, wird nicht erfüllt. Doch es ist möglich die abgespeicherten

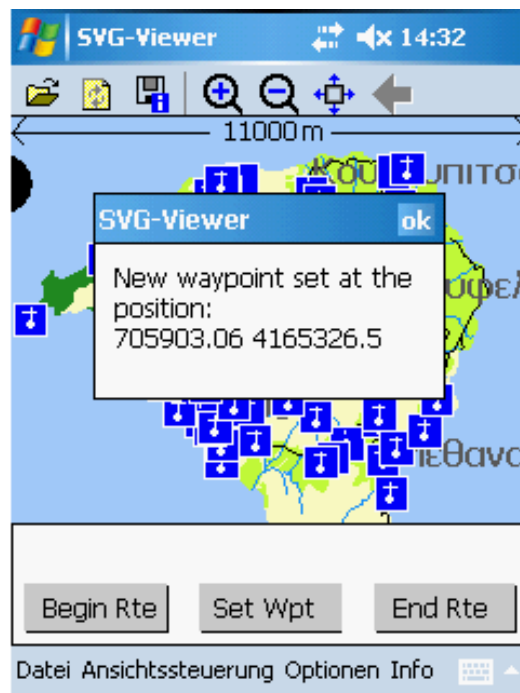


Abbildung 14: Aufnahme neuer Wegpunkte

Aufnahmen von Hand weiterzubearbeiten. Auf diese Weise können die Aufnahmen in ein GIS eingefügt werden.

Das erste grosse Problem war die Installation eines SVG-Viewers. Es gibt zwar mehrere mobile SVG-Viewer, doch sind diese vor allem für Mobiltelefone entwickelt und für Endkunden zum Teil gar nicht erhältlich. Zudem implementieren diese Viewer grösstenteils die SVG Tiny 1.1-Spezifikation, die kein Scripting umfasst. Es scheint, dass im Umfeld der Mobiltelefonie SVG vor allem für animierte Grusskarten eingesetzt wird. Da sich zudem keine Möglichkeit finden lässt, eine Java-Umgebung auf dem Windows Mobile-Betriebssystem zu installieren, fallen weitere Möglichkeiten der SVG-Darstellung weg. Schlussendlich blieb der SVG-Viewer der FH Oldenburg übrig, welcher installiert, angepasst und erweitert wurde.

Die fertige mobile Karte kann nur mit dem SVG-Viewer der FH Oldenburg richtig dargestellt werden. Das kommt daher, weil gewisse Eigenschaften des Viewers anders als in anderen Viewer sind. Die Portabilität der Karte zwischen verschiedenen Viewer ist nicht gegeben. So kann die Karte z.B. nicht auf dem Adobe SVG Viewer betrachtet werden. Wie in den Anwendungsfällen aufgezeigt, richtet sich die Karte in erster Linie an Touristen. Diese Ausrichtung liegt an der Auswahl der POI, welche touristische Sehenswürdigkeiten beinhaltet. Um eine grössere Zielgruppe anzusprechen, müssten mehrere Kategorien von POI in der Karte angezeigt werden können.

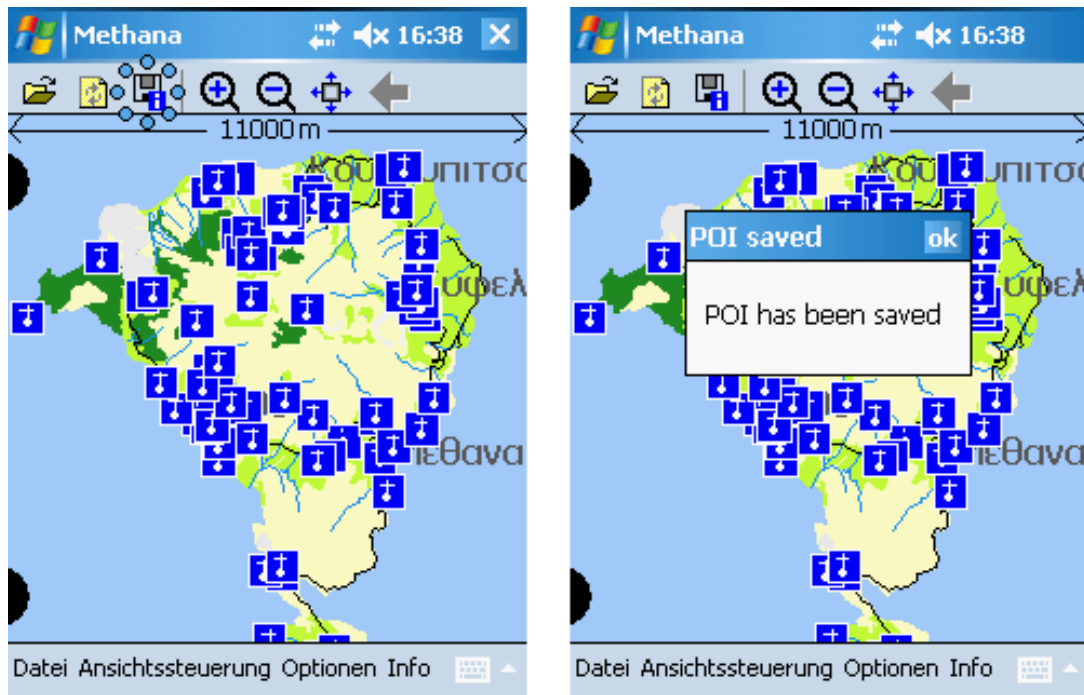


Abbildung 15: Speichern neuer POI und Routen

7.2 Ausblick

Eine sinnvolle Ergänzung der interaktiven mobilen Karte wäre ein erweiterter Umgang der neu aufgenommenen Punkt- und Linienelementen. Nebst dem Aufnehmen und Speichern eigener Elemente wäre es wünschenswert, dass diese Elemente klassifiziert und beschrieben und später jederzeit wieder in die Karte geladen werden könnten. Dazu müsste es möglich sein, die Datei beim Speichern zu benennen. Im Zusammenhang mit POI wäre es zudem interessant, die Navigation zu den POI zu verbessern und Hinweise auf POI in der Nähe zu geben.

Nebst der touristischen Ausrichtung dieser Anwendung, könnte sie auch etwas mehr in Richtung Vermessung entwickelt werden. So wären vermessungstechnische Arbeiten, die nicht zu hohe Genauigkeiten erfordern, wie z.B. die Aufnahme neuer Wege, sehr gut vorstellbar.

Andere Verbesserungsmöglichkeiten betreffen vor allem den SVG-Viewer. So wäre in erster Linie die grafische Darstellung zu verbessern z.B. mit einer Anti-Aliasing-Funktion, welche die verpixelte Anzeige von Objekten reduziert. Weiter wäre eine erweiterte Unterstützung von Skripten und der SVG Spezifikation wünschenswert. Aus Sicht des Autors ist eine verbesserte Unterstützung von Skripten wichtiger als diejenige der SVG Spezifikation.

Mobile Geräte kombiniert mit GPS-Empfängern eröffnen der Kartografie neue Perspek-

tiven indem digitale, interaktive Karten für den Benutzer portabel werden. Es entstehen zahlreiche neue Anwendungsmöglichkeiten für ortsbezogene Dienste.

Bis LBS alltäglich genutzt werden, gibt es noch einige Fragen im Bereich Marktpotentiale, Kundenbedürfnisse sowie Finanzierung zu klären. Ein wichtiger Erfolgsfaktor für eine hohe Marktakzeptanz, insbesondere für kommerzielle LBS, wird das Inhalteangebot sein. [Enichlmair und Staufer-Steinnocher, 2002]

So müsste auch für den vorliegenden Fall von Methana untersucht werden, wie gross ein möglicher Anwenderkreis ist, und welche Bedürfnisse die Benutzer haben.

In dieser Arbeit wird gezeigt, dass die technischen Voraussetzungen und das kartografische Wissen vorhanden sind, und Karten mit anspruchsvollen Interaktionen für mobile Geräte entwickelt werden können.

Adrian Weber

Literatur

- [Adam, 2004] Adam, A. (2004). Entwicklung eines Fahrradtouren-Navigationssystems für Pocket PCs. Universität Hannover, Hannover.
- [adobe.com, 2006] <http://www.adobe.com/svg/viewer/install/> (Juli 2006).
- [Brinkhoff und Weitkämper, 2004] Brinkhoff, T. und Weitkämper, J. (2004). Mobile Viewers based on SVG^{±geo} and XFormsGI
- [Brinkhoff und Weitkämper, 2004] Brinkhoff, T. und Weitkämper, J. (2004). Visualisierung und interaktive Bearbeitung von Geodaten mit SVG^{±geo}
- [bitflash.com, 2006] <http://www.bitflash.com/> (Juli 2006).
- [carto.net, 2006] <http://www.carto.net/> (Juli 2006).
- [ecma-international.org, 2006] <http://www.ecma-international.org/> (Juli 2006).
- [Elmiger et al., 2003] Elmiger, A., Chaperon, F. und Bürki B. (2003). Physikalische Geodäsie und Landesvermessung, Notizen zur Vorlesung. ETH Zürich, Zürich.
- [Enichlmair und Stauer-Steinnocher, 2002] Enichlmair, C. und Stauer-Steinnocher, P. (2002). Location Based Services – marktrelevante Inhalte als Erfolgsfaktor für mobile GIS- und Telekomtechnologien. In Zipf A. und Strobl J. (Hrsg.). Geoinformation mobil. Wichmann, Heidelberg.
- [epsg.org, 2006] <http://www.epsg.org/> (Juli 2006).
- [esri.com, 2006] <http://www.esri.com/> (Juli 2006).
- [esvg.com, 2006] <http://www.esvg.com/> (Juli 2006).
- [fh-oow.de, 2006] <http://www.fh-oow/institute/iapg/projekte/svgmobil/> (Juli 2006).
- [Flanagan, 2003] Flanagan D. (2003). JavaScript, kurz & gut. O'Reilly, Köln.
- [Gabaglio, 2003] Gabaglio, V. (2003). GPS/INS Integration for Pedestrian Navigation. EPF Lausanne, Lausanne.
- [garmin.com] <http://www.garmin.com/> (Juli 2006).
- [Gasenzer, 2001] Gasenzer R. (2001). Mobile Commerce und Location Based Services: Positions-basierte Leistungsangebote für den mobilen Handel. HMD - Praxis der Wirtschaftsinformatik Nr. 8/2001.
- [Gauer, 2006] Gauer, A. (2006). Interaktive Karte von Methana.ETH Zürich, Zürich.
- [gimodig.fgi.fi, 2006] <http://gimodig.fgi.fi/> (Juli 2006).
- [Halbach, 2005] Mobiler Stadtführer für das Finden und Nutzen touristischer und kultureller Sehenswürdigkeiten. FHTW Berlin, Berlin.

- [Hurni, 1995] Hurni, L. (1995). Modellhafte Arbeitsabläufe zur digitalen Erstellung von topographischen und geologischen Karten und dreidimensionalen Visualisierungen. ETH Zürich, Zürich.
- [ikivo.com, 2006] <http://www.ikivo.com/> (Juli 2006).
- [Ingensand, 2004] Ingensand, H. (2004). Einführung in die Geodätische Messtechnik. ETH Zürich, Zürich.
- [Kray et al., 2002] Kray, C., Baus J. und Krüger A. (2002). Positionsinformation und Navigationsaufgaben. In Zipf A. und Strobl J. (Hrsg.). Geoinformation mobil. Wichmann, Heidelberg.
- [mozilla.org, 2006] <http://www.mozilla.org/projects/minimo/> (Juli 2006)
- [Mundle, 1999] Mundle, H. (1999). Methana 3D – Eine kartografische Online-3D-Plattform mit Datenbankbindung der Halbinsel Methana (Griechenland). FH Karlsruhe, Karlsruhe.
- [Nissen et al., 2004] Nissen, F., Hvas, A., Münster-Swendsen, J. und Brodersen, L. (2003). Small-Display Cartography. National Survey and Cadastre, Dänemark.
- [openwave.com, 2006] http://www.openwave.com/us/products/device_products/index.htm (Juli 2006).
- [opera.com, 2006] <http://www.opera.com/products/mobile/> (Juli 2006).
- [paramount-tours.com, 2006] <http://www.paramount-tours.com/> (Juli 2006).
- [postgis.refrations.net, 2006] <http://postgis.refrations.net/> (Juli 2006).
- [postgres.org, 2006] <http://www.postgresql.org/> (Juli 2006).
- [Reichenbacher, 2003] Reichenbacher, T. (2003). Mobile Cartography - Adaptive Visualisation of Geographic Information on Mobile Devices
- [Rerrer, 2005] Rerrer, U. (2005). Location-awareness for a Service-oriented Architecture using WLAN Positioning. Universität Paderborn, Paderborn.
- [Sayda et al., 2002] Sayda, F., Reinhardt, W. und Wittmann, E. (2002). Positionsbezogene Dienste zur Unterstützung von Bergsteigern und Wanderern. In Zipf A. und Strobl J. (Hrsg.). Geoinformation mobil. Wichmann, Heidelberg.
- [SGK, 2002] Schweizerische Gesellschaft für Kartografie (2002). Topografische Karten - Kartengrafik und Generalisierung, Kartografische Publikationsreihe Nr. 16. S. 41.
- [sirf.com, 2006] [http://www.sirf.com/Downloads/Collateral/GSC3\(f\)_6.20.05.pdf](http://www.sirf.com/Downloads/Collateral/GSC3(f)_6.20.05.pdf) (Juli 2006).
- [Snyder, 1982] Snyder, J. P. (1982). Map Projections Used by the U.S. Geological Survey. U.S. Government Printing Office, Washington.

[swisstopo, 2003] swisstopo (2003). Zeichenerklärung und weitere Informationen zu den Landeskarten. Bern.

[tinyline.com, 2006] <http://www.tinyline.com/> (Juli 2006).

[topografix.com, 2006] <http://www.topografix.com/gpx/> (Juli 2006).

[w3.org, 2006] <http://www.w3.org/> (Juli 2006)

[webparkservices.info, 2006] <http://www.webparkservices.info/> (Juli 2006).

[Würth, 2006] Würth, R. (2006). Ein Navigationsassistent für Fussgänger in der Stadt. Universität Zürich, Zürich.

[x-smiles.org, 2006] <http://www.x-smiles.org/> (Juli 2006).

A Quellcode zur Einbindung der GPS-Koordinaten in das SVG-Dokument

Nachfolgend der Quellcode zur Einbindung der GPS-Koordinaten in das SVG-Dokument. Zur Einbindung von GPS-Daten wird in das SVG-Hauptdokument ein *XForms-GI* Kontrollelement des Typs `sensor` eingefügt:

```
<xfmgi:sensor type="gps"
  id="gps-sensor"
  model="#form1"
  ref="/daten"
  cursor="#gps-cursor"
  mode="trackview trackangle"
  state="running"/>
```

Die Visualisierung des GPS-Cursors kann mit dem Attribut `cursor` bestimmt werden. Diesem Attribut wird als Wert die Identität einer beliebigen SVG-Gruppe, die den Cursor darstellen soll, übergeben.

Das *XForms Model* `form1` definiert im Hauptdokument die Struktur der einzugebenden Daten:

```
<xfm:model id="form1">
  <xfm:instance>
    <daten>
      <xfmgi:svgcoords id="svgcoords"/>
      <xfmgi:globalcoords id="globalcoords"/>
    </daten>
  </xfm:instance>
</xfm:model>
```

Nebst dieser Datenstruktur braucht es Angaben zum Bezugssystem, damit die im WGS84-System empfangenen GPS-Koordinaten ins gewünschte System transformiert werden können. Dazu wird das neue Element `coordinatereferencesystem` definiert:

```
<xfmgi:coordinatereferencesystem
  id="coordinatereferencesystem"
  crs-id="EPSG:32632"
  transform="scale(1,-1)"/>
```

Das Attribut `crs-id` speichert den EPSG-Code des Referenzsystems und `transform` beschreibt die Transformation der geodätischen Koordinaten in SVG-Koordinaten.

B Die Funktion `CSVGViewerDemoView::OnSaveFile()`

Dies ist die Funktion `OnSaveFile()` aus der Klasse `CSVGViewerDemoView`, die beim Speichern von aufgenommenen POI und Routen verwendet wird. Die verwendete Programmiersprache ist Visual C++.

```
void CSVGViewerDemoView::OnSaveFile()
{
    CSVGViewerDemoDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    if (!pDoc) return;

    SVGElementSvg &root = pDoc->GetSVGDocument().GetRootElement();

    DOMNode *groupNode = root.GetElementById("groupPersonalPOI");

    if(groupNode == 0)
    {
        MessageBox( L"Group 'groupPersonalPOI' was not found",
            L"Not found", MB_OK);
        return;
    }

    TFileBuffer *tFileBuffer =
        new TFileBuffer(L"/My Documents/personalPOI.svg", true);
    if (!tFileBuffer->IsOpen()) return;

    XMLWriter buffer(tFileBuffer);

    groupNode->OutputAsXMLString(buffer);

    MessageBox(L"POI has been saved",L"POI saved", MB_OK);
}
```


C Auszug aus Datei der gespeicherten POI

Nachfolgend ein beispielhafter Auszug aus einer Datei, in welche die vom Benutzer aufgenommenen POI und Routen geschrieben worden sind:

```
<g id="groupPersonalPOI" >
  <use x="705788.50" y="-4165044.50" id="mthna_test"
    xlink:href="villageS" >
  </use>
  <g id="rte_test" >
    <use x="705788.50" y="-4165044.50" xlink:href="wptS" >
    </use>
    <use x="705801.94" y="-4165005.00" xlink:href="wptS" >
    </use>
    <use x="705801.94" y="-4165005.00" xlink:href="wptS" >
    </use>
    <use x="705801.94" y="-4165005.00" xlink:href="wptS" >
    </use>
  </g>
</g>
```