

## **Eine Architektur zur XML-basierten Repräsentation von bewegten Geo-Objekten (Kurzbeitrag)**

Thomas Brinkhoff und Jürgen Weitkämper

Institut für Angewandte Photogrammetrie und Geoinformatik (IAPG)  
Fachhochschule Oldenburg/Ostfriesland/Wilhelmshaven  
Ofener Str. 16/19, D-26121 Oldenburg  
{Thomas.Brinkhoff, Weitkaemper}@fh-oldenburg.de

**Zusammenfassung.** Das Forschungsgebiet der Geo-Datenbanksysteme wird derzeit durch die Entwicklung geeigneter Datenmodelle und Algorithmen für raum-zeitbezogene Daten geprägt, um zum Beispiel bewegte Geo-Objekte für Anwendungen der Verkehrstelematik verwalten zu können. Einen generellen Trend stellt die Nutzung von XML zur Repräsentation von Daten dar, insbesondere – aber nicht nur – für Internet-Applikationen. In diesem Beitrag sollen die beiden Entwicklungen in einer Architektur zur XML-basierten Repräsentation von bewegten Geo-Objekten zusammengeführt werden. Dies erfolgt speziell in Hinblick auf heterogene Web-Anwendungen. Nach Darstellung der Anforderungen werden die einzelnen Komponenten einer solchen Architektur und deren Zusammenspiel vorgestellt. Dabei zeigt sich, dass der Aufbau einer geeigneten Infrastruktur die Lösung einer Reihe von Fragestellungen erfordert, die in einer konventionellen Architektur nicht auftreten. Davon sind sowohl die Clients als auch die Server- und Datenbank-Komponenten betroffen.

### **1 Einleitung**

Während sich in der letzten Dekade die Forschungsaktivitäten im Bereich von Geo-Daten insbesondere auf den Entwurf und die Entwicklung von Architekturen, Datenstrukturen und Anfragestrategien zur effizienten Behandlung *statischer* Geo-Daten konzentriert haben, ist das derzeitige Forschungsgeschehen sehr stark durch die Betrachtung von *raum-zeitbezogenen Daten* („*Spatiotemporal Data*“) geprägt. Hierzu sei u.a. auf das EU-Projekt "Chorochronos" [15] verwiesen. Motivation für diese Entwicklung ist insbesondere das Aufkommen neuer Anwendungen zum Beispiel aus dem Bereich der Verkehrstelematik ([2], [25]), die die Verwaltung von bewegten Objekten (z.B. von Verkehrsteilnehmern) erfordern.

Ein weiterer Trend im Bereich der Verwaltung von Geo-Daten betrifft deren Präsentation mittels des World Wide Webs. Nahezu jedes *Geoinformationssystem (GIS)* bietet Aufsätze an, die es ermöglichen, Karten und andere raumbezogene Daten im Internet einer breiten Öffentlichkeit bereitzustellen; auch wurden spezielle Geo-Server-Architekturen für die Nutzung von GIS-Funktionalität über das Internet entwickelt [6]. Eine Problematik, die alle Internet-Anwendungen, die mit Geo-Daten arbei-

ten, zu lösen haben, ist die Repräsentation und Darstellung dieser Daten. Da HTML standardmäßig nur Rasterdaten, aber keine Vektorgrafiken unterstützt, führt dies entweder zu funktionalen Einschränkungen oder zu Lösungen, die die Nutzung solcher Anwendungen für große Nutzergruppen erschwert [3].

Eine Lösung dieses Problems zeichnet sich durch die Verwendung von *XML (Extensible Markup Language)* ab, einer Metagrammatik für kontextfreie Grammatiken, mit der sich die Syntax von Auszeichnungssprachen definieren lässt [21]. Ziel von XML ist es, flexibel die Struktur von Dokumenten zu definieren, die im World Wide Web übertragen werden. Durch den Einsatz von XML lassen sich anwendungsspezifische Dokumente beschreiben, für die HTML zu restriktiv ist.

Auch im Geoinformationswesen ist zu erwarten, dass XML eine bedeutende Rolle zur Repräsentation und Übermittlung von Dokumenten im Internet einnehmen wird. Gerade wurde durch das OpenGIS-Konsortium (OGC) eine XML-Spezifikation, die *Geography Markup Language (GML)*, erarbeitet, die zur Repräsentation einfacher geographischer Merkmale dient [13]. Es sind bereits erste Geoinformationssysteme auf dem Markt bzw. angekündigt, die GML unterstützen (z.B. [8], [16]). Im Bereich von Vektorgrafiken wird derzeit der neue, XML-basierte Grafikstandard *SVG (Scalable Vector Graphics)* normiert [24]. SVG eignet sich wesentlich besser zur Darstellung kartografischer Informationen als bisherige Datenformate des WWW [12].

Der Bereich der XML-basierten Web-Anwendungen erfährt zur Zeit eine rasante Entwicklung. Mehrere Frameworks sind im Aufbau begriffen, die die Entwicklung solcher Anwendungen unterstützen sollen (siehe u.a. Sun Microsystems *XML at Sun* [18], das *Apache XML Project* [1], *Microsoft .NET Framework* [11]). Daraus ergibt sich eine breite Basis an Werkzeugen, die jeweils für die speziellen Belange angepasst und erweitert werden können.

Die Verbindung der genannten Trends soll im angelaufenen Projekt "Der Einsatz von XML zur Repräsentation und Visualisierung von bewegten Geo-Objekten" erfolgen. Dazu wurde eine Architektur entwickelt, die eine XML-basierte Repräsentation von bewegten Geo-Objekten erlaubt. Diese Architektur umfasst die Kette vom Client, der für die Visualisierung der Daten und die Nutzerinteraktion verantwortlich ist, bis hin zu einem *spatiotemporalen Datenbanksystem*. Dabei wurden insbesondere die Implikationen betrachtet, die sich aus dem Einsatz von Clients ergeben, die sich in ihrer Präsentations- und Kommunikationstechnologie voneinander unterscheiden, z.B. für die Unterstützung von mobilen Applikationen. Weitere Fragestellungen betreffen die geeignete Repräsentation von bewegten Geo-Objekten und deren Visualisierung.

In diesem Beitrag werden zunächst die Anforderungen an eine solche Architektur präsentiert. Abschnitt 3 stellt die Architektur komponentenweise vor. Den Abschluss bilden eine Zusammenfassung und ein Ausblick auf künftige Aktivitäten.

## **2 Anforderungen an die Architektur**

### **2.1 Der Client**

Hauptaufgaben von Internet-Clients sind die Präsentation der Web-Inhalte und die Interaktion mit dem Nutzer. Aus Sicht eines Datenbanksystems stellt der Client spa-

tiotemporale Anfragen und nimmt die Anfrageergebnisse entgegen. Zusätzlich ist der Client für die Kommunikation mit dem Internet-Server verantwortlich.

Schon bei der Darstellung von reinen HTML-Seiten tritt das Problem auf, dass sich die Darstellung von Client zu Client unterscheidet. Ursachen sind typischerweise der Einsatz unterschiedlicher Browser, Betriebssysteme und Hardware, wobei insbesondere die Bildschirmereigenschaften einen wesentlichen Einflussfaktor darstellen. Gegenwärtig verschärft sich die Problematik durch den Einsatz von Endgeräten, die bislang nicht für die Darstellung von Web-Inhalten geeignet waren. Zu herkömmlichen PCs und Workstations treten nun u.a. Fernsehkonsolen, PDAs und Mobiltelefone in Konkurrenz. Diese Vielfalt von Endgeräten mit signifikant unterschiedlichen Hardware-Eigenschaften erfordert es, unterschiedliche Visualisierungs- und Interaktionstechniken zu unterstützen. Daneben können die Endgeräte auch hinsichtlich der vorhandenen Basis-Software differieren, z.B. bezüglich der Verfügbarkeit von Plugins oder von Java-Editionen; beispielsweise könnte auf PDAs die Java 2 Micro Edition [17] anstelle der üblichen Standard Edition installiert sein.

## **2.2 Internet- und Map-Server, Verwaltungsdatenbank**

Der *Internet-Server* wickelt zunächst das HTTP-Protokoll ab. HTTP als zustandsloses Protokoll bietet keine Möglichkeit, eine Client-Verbindung über HTTP-Requests hinweg zu identifizieren. Für die Verwaltung der aktuell bestehenden offenen Client-abfragen wird daher eine separate *Verwaltungsdatenbank* benötigt.

Daneben ist die Umsetzung der Daten in das vom Client benötigte Datenformat Aufgabe des Internet-Servers. Alle weitergehenden Aufgaben werden üblicherweise einer anderen Komponente, dem Map-Server, übertragen [6].

Hauptaufgabe eines *Map-Servers* (auch *Geo-Server* genannt) ist das Zusammenspiel mit einem Geo-Datenbanksystem; in kommerziellen Lösungen wird die Rolle des Geo-Datenbanksystems meist von einem Geoinformationssystem (oft in Zusammenspiel mit Standard-Datenbanksystemen) übernommen [3]. Für ein System, das raum-zeitbezogene Daten verwaltet und bereitstellt, wird die Rolle des Geo-Datenbanksystems durch ein spatiotemporales Datenbanksystem wahrgenommen. Außerdem kann es unabhängig davon auch erforderlich sein, weitere (Standard-) Datenbanksysteme anzusprechen. Der Map-Server leitet die Anfragen an das bzw. die Datenbanksysteme weiter und bereitet die Anfrageresultate auf.

## **2.3 Das spatiotemporale Datenbanksystem**

Ein spatiotemporales Datenbanksystem hat primär zwei Aufgaben: 1. die effektive Speicherung und Aktualisierung der temporalen Geo-Daten und 2. die effiziente Bearbeitung von spatiotemporalen Anfragen. Wir gehen in der nachfolgend vorgestellten Architektur davon aus, dass die Aktualisierung der Datenbank nicht über die Komponenten erfolgt, die dem Internet-basierten Zugriff dienen, sondern über andere (interne) Schnittstellen. In diesem Sinne ist die Architektur auskunftsorientiert.

## 2.4 Zeitliche Aktualisierung

Spatiotemporale Anfragen produzieren über ein – gegebenenfalls unendlich langes – Zeitintervall immer wieder Ergebnisse (s. a. Abschnitt 3.1.2). Prinzipiell können sowohl der Client, als auch der Server die Initiative zur Aktualisierung der Daten übernehmen. Da der Client nicht weiß, wann und in welchem Umfang sich Daten verändert haben, ist es zweckmäßig, dass die Initiative von der Server-Seite ausgeht. (s. Abschnitt 3.1.3).

## 2.5 Datenrepräsentation

Benötigt werden zwei ergänzende XML-Anwendungen. Zunächst muß eine Sprache zur Beschreibung der zeitlichen Aspekte definiert werden („Temporal GML“). Einen Ansatz dafür könnte die für Multimedia-Anwendungen entwickelte XML-Anwendung SMIL (*Synchronized Multimedia Integration Language*) bieten [22]. Ferner müssen Änderungen bestehender XML-Daten geeignet repräsentiert werden.

## 3 Die Architektur

Das in Bild 1 dargestellte Diagramm gibt schematisch die wesentlichen Komponenten und Abläufe in unserer Architektur zur XML-basierten Repräsentation von bewegten Geo-Objekten wieder. Nachfolgend werden die einzelnen Komponenten erläutert.

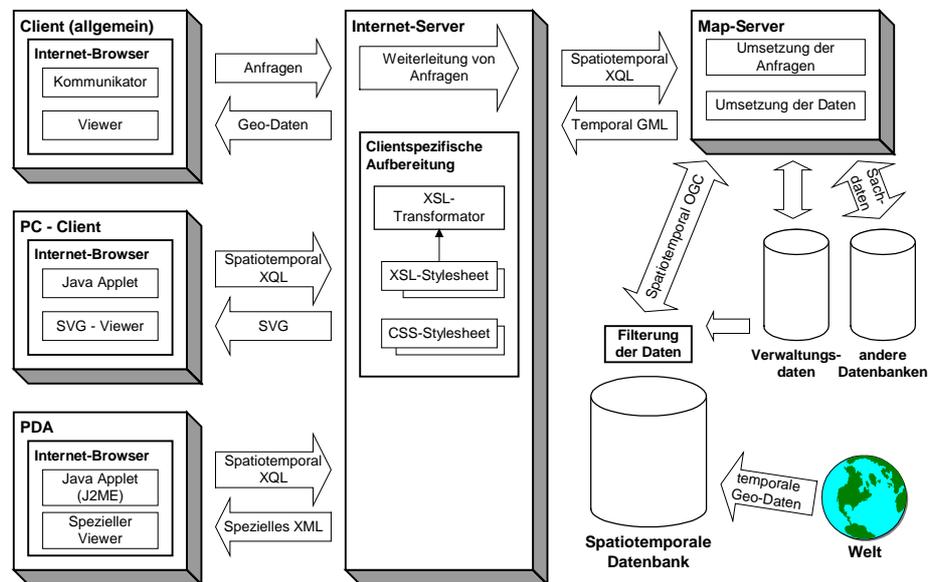


Bild 1. Architektur zur XML-basierten Repräsentation von bewegten Geo-Objekten

### 3.1 Der Client

Betrachtet werden nachfolgend die Visualisierung sich bewegender Geo-Objekte, die typischen Anfragen und die Kommunikation des Clients mit dem Internet-Server.

#### 3.1.1 Visualisierung

Bei temporalen Geo-Daten handelt es sich primär um Vektordaten. Wie erwähnt, steht für die Darstellung von Vektorgrafiken bislang kein einheitlicher, allgemein anerkannter Standard zur Verfügung. Typische Lösungen für diese Problematik im GIS-Umfeld sind folgende:

- Die Verwendung von Rasterformaten, die dynamisch auf der Server-Seite generiert werden (z.B. [4]). Die Folge ist eine eingeschränkte Funktionalität auf der Client-Seite und ein hohes Datenübertragungsvolumen.
- Eine zweite Lösung ist der Einsatz von Zusatz-Software, die die Funktionalität des Internet-Browsers erweitert. Je nach Einsatzfeld liegt diese Software als Java-Applet oder als Active-X-Komponente vor. Insbesondere ad-hoc-Nutzer von Internet-Anwendungen mit Geo-Daten werden durch solche Lösungen abgeschreckt.
- Vergleichbar zur zweiten Lösung ist der Einsatz von proprietären Formaten wie z.B. Macromedia Flash [10], die aber Einschränkungen für kartografische Anwendungen aufweisen [12] und in keiner Weise standardisiert sind.

Ziel ist also ein gleichermaßen flexibles wie auch standardisiertes Format für Vektorgrafiken. Damit bietet sich der Einsatz eines XML-basierten Datenformats an. Ein guter Kandidat dafür ist der Grafikstandard SVG (Scalable Vector Graphics), der zur Zeit in der Version 1.0 den Status einer Candidate Recommendation des WWW-Konsortiums hat [24]. SVG bietet eine Reihe von Eigenschaften, die die Darstellung kartografischer Informationen unterstützen [12].

Um ein Grafikformat (und die zugehörigen Viewer) zur Darstellung von sich bewegenden Geo-Objekten verwenden zu können, ist es erforderlich, dass nachträglich Änderungen an den bislang übertragenen Daten möglich sind; das Einfügen neuer Geo-Objekte, die Veränderung existierender Objekte primär hinsichtlich ihrer Position, aber prinzipiell auch hinsichtlich ihrer Gestalt sowie das Löschen von vorhandenen Geo-Objekten ist clientseitig zu unterstützen. SVG erlaubt diese Modifikationen, indem es einen vollständigen Zugriff auf das hierarchische *Document Object Model (DOM)* zulässt und über Skripte Veränderungen in diesem Objektbaum unterstützt.

Während bei Standard-Browsern zu erwarten steht, dass sie in absehbarer Zeit SVG auch ohne zusätzliche Plug-ins unterstützen, kann dagegen bei Endgeräten wie PDAs oder Mobilfunktelefonen zur Zeit keine Aussage zur Verfügbarkeit und Mächtigkeit von SVG-Viewern gemacht werden. Daher muss die Architektur in der Lage sein, flexibel andere (möglichst auch XML-basierte) Grafikformate zu unterstützen.

#### 3.1.2 Anfragen

Prinzipiell können wir verschiedene Typen von Anfragen unterscheiden. Von besonderer Bedeutung sind *spatiotemporale Anfragen*. Diese haben sowohl einen temporalen als auch einen raumbezogenen Anteil in der Anfragebedingung und können somit vom spatiotemporalen Datenbanksystem optimiert bearbeitet werden. Der Client erhält die augenblicklich oder für einen beliebigen Zeitpunkt oder -raum gültigen

Geo-Objekte, die beispielsweise in einem bestimmten Anfragerechteck liegen. Sollten Objekte für einen Zeitraum angefragt worden sein, ist die Aufbereitung der Ergebnisdaten, um die Bewegung der Daten darzustellen, Aufgabe des Clients bzw. des Viewers. *Temporale Anfragen* ohne eine raumbezogene Anfragebedingung können auch auftreten, sind aber für spatiotemporale Datenbanksysteme eher untypisch.

Für die Darstellung sich bewegender Geo-Objekte gibt es eine wichtige Variante von spatiotemporalen Anfragen, bei denen das Zeitintervall alle Daten vom aktuellen Zeitpunkt bis in die unbegrenzte Zukunft umfasst. In diesem Fall werden zunächst alle aktuellen Daten (z.B. eines Gebiets) an den Client übertragen und dort angezeigt. Danach findet eine *kontinuierliche Aktualisierung* statt: Sobald es geboten ist (dazu mehr in Abschnitt 3.4), werden Informationen über neue, geänderte und gelöschte Daten von dem spatiotemporalen Datenbanksystem entdeckt und übertragen.

Um spatiotemporale Anfragen formulieren zu können, bedarf es einer Grundlage, auf der die bewegten Objekte repräsentiert und angefragt werden können. Eine grundlegende Arbeit in dieser Richtung stellt der Artikel von Güting et al. [7] dar, in dem spatiotemporale Datentypen und Operationen vorgestellt und in eine Anfragesprache eingebettet werden. Eine Umsetzung in ein diskretes Datenmodell findet sich in [5]. In einem XML-basierten Umfeld ist eine Übertragung dieser oder vergleichbarer Ergebnisse in eine XML-Notation zweckmäßig. Dies ist in Bild 1 durch den Begriff „Spatiotemporal XQL“ angedeutet.

### 3.1.3 Kommunikation mit dem Internet-Server

Für das Bearbeiten der Anfragen ist eine Kommunikation mit dem Internet-Server erforderlich. Dies ist Standardfunktionalität. Zu betrachten ist aber die kontinuierliche Aktualisierung von Geo-Objekten, die sich so nicht in vergleichbaren Geo-Server-Architekturen findet. Da der Client nicht weiß, wann und in welchem Umfang sich Daten verändert haben, ist es zweckmäßig, dass die Initiative dazu von der Server-Seite ausgeht. Auf abstrakter Ebene wird in diesem Fall also die Rolle zwischen Server und Client ausgetauscht.

Für die serverseitig initiierte Auffrischung der Daten steht die Server-Push-Technologie zur Verfügung, die auf Basis einer festen TCP-Verbindung beruht und bei der zur Aktualisierung die komplette HTML-Seite oder -Seiten neu übertragen werden. Diese Technik belastet durch die vielen aktiven Verbindungen den Server sehr. Dazu kommt, dass für die hier betrachteten Aufgaben i.d.R. nur eine kleine Teilmenge der bereits beim Client vorhandenen Daten verändert werden muss. Eine komplette Neuübertragung wäre damit eine Verschwendung von Übertragungskapazität.

Im Rahmen der hier vorgestellten Architektur wird statt dessen innerhalb eines Applets ein "clientseitiger" Server gestartet, der beim Web-Server registriert wird. Dieser Mechanismus ermöglicht zusätzlich eine Identifizierung der Clients über einzelne HTTP-Requests hinweg: in der oben bereits erwähnten Verwaltungsdatenbank werden zur Identifizierung die IP- und Portnummer dieses Servers registriert. In Bild 1 ist dies durch die Komponente „Kommunikator“ beim Client angedeutet. Zu untersuchen bleibt die Fragestellung, ob diese Lösung für das gewünschte Anwendungsumfeld hinreichend skalierbar ist.

### 3.2 Der Internet-Server

Der Internet-Server ist ein handelsüblicher HTTP-Server zur Behandlung der Seitenanforderungen der Clients. Darüber hinaus werden die Anfragen der Clients an den Map-Server weitergeleitet und die Resultate der Anfragen in die clientspezifischen Daten- und Stilformate umgesetzt.

#### 3.2.1 Aufbereitung der Anfrageresultate

Wie bereits erwähnt, muss davon ausgegangen werden, dass die Clients unterschiedliche Grafikformate benötigen. Dabei nehmen wir an, dass diese Formate über XML definiert sind. Andererseits sollte der Map-Server zur Minderung der Komplexität mit nur einem Datenformat arbeiten. Auch hier ist es zweckmäßig, XML zur Beschreibung der Daten zu verwenden. Für die Repräsentation von Geo-Daten wurde durch das OpenGIS-Konsortium (OGC) eine XML-Spezifikation, die *Geography Markup Language GML*, erarbeitet [13]. Damit steht eine Kerndefinition zur Verfügung, die für eine einheitliche Darstellung von spatiotemporalen Daten (unter Verwendung der gleichen Datentypen wie bei der Definition der Anfragesprache) angepasst und erweitert werden kann. Diese „Temporal GML“ wäre damit das Datenformat, in dem der Map-Server die Daten zur Verfügung stellt.

Für eine Umsetzung zwischen zwei XML-Datenformaten ist der Einsatz von XSL-Transformationen angebracht. Der Transformationsteil von *XSL (Extensible Style Language)* [23] überführt ein XML-Dokument unter Verwendung von XSL-Stylesheets in ein anderes XML-Dokument. Um unterschiedliche Datenformate der Clients zu unterstützen, müssen unterschiedliche XSL-Stylesheets beim Internet-Server bereitgehalten werden. Die Integration des XSL-Transformators in den Server kann als Java-Servlet erfolgen. Prinzipiell ist es auch möglich, die XSL-Transformation erst beim Client auszuführen. Allerdings ist insbesondere für leistungsschwächere Endgeräte eine serverseitige Transformation vorzuziehen. Über die Verwendung unterschiedlicher CSS-Stylesheets kann eine client- oder benutzerspezifische Formatierung der Daten erreicht werden.

Im Vergleich zu Binärformaten blähen XML-Dokumente das Datenvolumen sehr stark auf. Allerdings wird dieses Problem durch eine gute Komprimierbarkeit größtenteils wieder wettgemacht. Vorläufige Ergebnisse haben gezeigt, dass bei gleichen Daten eine ZIP-Komprimierung GML-Dateien etwa um einen Faktor 20, SVG-Daten etwa um den Faktor 7,5 und Binärdateien ungefähr um den Faktor 2,5 komprimiert.

### 3.3 Der Map-Server

Hauptaufgabe eines *Map-Servers* ist die Koordinierung des Zusammenspiels mit dem oder den Datenbanksystemen. Der Map-Server leitet die Anfragen an das jeweilige System weiter und bereitet die (Anfrage-) Resultate auf. Da bei spatiotemporalen Datenbanksystemen die Verwaltung der Daten im wesentlichen nach räumlichen und zeitlichen Kriterien erfolgt, kann nicht davon ausgegangen werden, dass die Daten in der Datenbank in XML vorliegen oder über XML angefragt werden können. Damit sind seitens des Map-Servers einerseits die Anfragen in datenbankspezifische Anfragesprachen und Datenmodelle umzusetzen und andererseits die Anfrageresultate in

Temporal GML als einheitliches Datenformat zu überführen. Aus Effizienzgründen kann dies aber durchaus im Rahmen eines Datenbankprozesses erfolgen.

### **3.4 Das spatiotemporale Datenbanksystem**

Zum Themenkomplex spatiotemporale Datenbanksysteme sind in den letzten Jahren eine Reihe von Arbeiten entstanden. Insbesondere die Themen Datenmodellierung und Definition von Anfragen (z.B. [19], [7]) sowie Indizierung (z.B. [20], [9], [14]) wurden betrachtet. Eine Reihe von Aspekten, die die Speicherung von spatiotemporalen Daten betreffen, wurde hingegen bislang noch nicht vertieft untersucht. Zu nennen sind z.B. eine geeignete Clusterbildung und spezifische Pufferungsstrategien.

#### **3.4.1 Kontinuierliche Anfrageweiterbearbeitung**

Die kontinuierliche Aktualisierung der Clients erfordert, dass die Bearbeitung einer Anfrage nicht nach Zusammenstellung des Resultats abgeschlossen ist, sondern fortgesetzt werden muss, bis ein gesonderter Befehl zur Beendigung der Anfrage beim spatiotemporalen Datenbanksystem eintrifft.

Ein erster Ansatz besteht darin, dass das Datenbanksystem, sobald Änderungen vorliegen, diese auf ihre Auswirkungen auf die vorliegenden, kontinuierlich weiterzubearbeitenden Anfragen prüft und ggf. Änderungen an den Map-Server weiterleitet. Eingeleitet werden kann dieser Prozess durch entsprechende Trigger. Dieser Ansatz berücksichtigt allerdings nicht, dass die Kommunikationsverbindungen zwischen dem Server und den Clients hinsichtlich der Geschwindigkeit und des Durchsatzes stark voneinander differieren können. Dies gilt bereits für das herkömmliche Internet und noch stärker für mobilfunkbasierte Netztechnologien. Zudem können sich die Clients in ihren Rechenleistungen stark voneinander unterscheiden. Damit besteht die Gefahr, dass die Weiterleitung aller geänderter Daten die Clients bzw. die vorhandenen Übertragungskapazitäten überfordert.

Ein weiterer Aspekt, der die Bearbeitung von kontinuierlichen Aktualisierungen beeinflusst, ist die vom Client geforderte Auflösung der Änderungen bezüglich Zeit und insbesondere Raum. So kann z.B. aufgrund einer beschränkten visuellen Auflösung es für einen bestimmten Client zweckmäßig sein, nur Veränderungen zu übertragen, die einen gegebenen Mindestabstand übersteigen.

Damit sich das Datenbanksystem auf die genannten Eigenschaften einstellen kann, ist es erforderlich, dass der Client seine *Verbindungs- und Verarbeitungseigenschaften* bekannt gibt und diese in der Verwaltungsdatenbank für den Zeitraum der Anfragebearbeitung gespeichert werden. Hierbei lassen sich zwei prinzipielle Fälle unterscheiden: 1. die Eigenschaften ändern sich über die Laufzeit der Anfrage nicht und 2. die Eigenschaften ändern sich beispielsweise wegen geänderter Auflösungsanforderungen (z.B. durch eine Zoom-Operation) oder aufgrund veränderter Übertragungseigenschaften; so hängt bei UMTS die Übertragungsgeschwindigkeit von der Entfernung des Endgerätes zur nächsten Sende- und Empfangsstation ab.

Die Verbindungs- und Verarbeitungsinformationen können nun zur Filterung von Daten im Rahmen der kontinuierlichen Aktualisierung eines Clients genutzt werden. Im wesentlichen beeinflussen 1. das Ausfiltern von nichtsignifikanten Änderungen, 2. die Einhaltung von Mindestzeiträumen zwischen zwei Aktualisierungen und 3. die

Beschränkung des Übertragungsvolumens die Filterung von Daten. Damit ist erforderlich, dass das Datenbanksystem Anfrageergebnisse clientspezifisch puffert und ggf. vor einer Übertragung verändert oder gar verwirft. Eine Konsequenz aus einer solchen Zwischenspeicherung ist, dass 1. eine geeignete Selektion von Daten, die übertragen werden sollen, erfolgt und dass 2. zur Begrenzung des Speicherbedarfs beim Datenbanksystem auch Aktualisierungen mit (vermeintlich) geringerer Relevanz für den Client aus dem Puffer entfernt werden. Für diese *Auswahlmechanismen* sind geeignete Heuristiken zu entwickeln und im Rahmen der vorgestellten Architektur näher zu untersuchen. Durch diesen Auswahl- und Verzögerungsprozess können in der Antwortmenge, die der Client erhält, entweder Informationen fehlen oder erst zeitverzögert übermittelt werden. Damit ist es auch erforderlich, ein geeignetes Maß zur Beschreibung der Qualität der Antwort zu entwickeln und zu überprüfen.

#### 4 Zusammenfassung und Ausblick

In diesem Beitrag wurde eine Architektur zur XML-basierten Repräsentation von bewegten Geo-Objekten vorgestellt. Wichtige Aspekte dieser Architektur sind die Berücksichtigung unterschiedlicher Endgeräte auf der Client-Seite, die Verwendung von XML-basierten Daten- und Anfrageformaten für spatiotemporale Daten, die Unterstützung von spatiotemporalen Anfragen und einer kontinuierlichen Aktualisierung der Clients.

Wesentliche Aufgabe ist es nun, für die vorgestellten Konzepte eine Feinspezifikation vorzunehmen. Dies betrifft insbesondere die Datenrepräsentation in „Temporal GML“, die Definition von Anfragen („Spatiotemporal XQL“) und die Untersuchung von Heuristiken zur Datenfilterung. Auf Grundlage der vorgestellten Architektur soll die weitere Fortentwicklung und Untersuchung ausgewählter Komponenten insbesondere zur kontinuierlichen Anfragebearbeitung erfolgen.

#### 5 Literatur

1. Apache Software Foundation: XML at Apache, <http://xml.apache.org/>
2. Brinkhoff T.: "Requirements of Traffic Telematics to Spatial Databases". *Proceedings 6th International Symposium on Large Spatial Databases*, Hong Kong, China, 1999, in: Lecture Notes in Computer Science, Vol.1651, Springer, 365-369.
3. Brinkhoff T.: "The Impacts of Map-Oriented Internet Applications on Internet Clients, Map Servers and Spatial Database Systems", *Proceeding 9th International Symposium on Spatial Data Handling*, Beijing, China, 2000.
4. ESRI Inc.: "ArcView Internet Map Server", 1999, <http://www.esri.com/software/arcview/extensions/imsext.html>
5. Forlizzi L., Güting R.H., Nardelli E., Schneider M.: "A Data Model and Data Structures for Moving Objects Databases", *Proceedings ACM SIGMOD International Conference on Management of Data*, Dallas, TX, 2000, 319-330.
6. Friebe J.: "Eine GeoServer-Architektur zur Nutzung von GIS-Funktionalität über Internet-Technologie", *Proceedings GI-Fachtagung Datenbanksysteme in Büro, Technik und Wissenschaft*, Freiburg, Germany, 1999, in: Informatik aktuell, Springer, 164-184.

7. Güting R.H., Böhlen M.H., Erwig M., Jensen C.S, Lorentzos N.A., Schneider M., Vazirgiannis M.: "A Foundation for Representing and Querying Moving Objects", *ACM Transactions on Database Systems*, Vol. 25, No.1, 2000, 1-42.
8. Intergraph Corp.: „Intergraph Announces GeoMedia 4.0 Product Suite”, Nov. 2000, [http://www.intergraph.com/gis/newsroom/press00/gm40\\_rlsf.asp](http://www.intergraph.com/gis/newsroom/press00/gm40_rlsf.asp)
9. G. Kollios, D. Gunopulos, V.J. Tsotras: "On Indexing Mobile Objects", *Proceedings ACM Symposium on Principles of Database Systems*, Philadelphia, PA, 1999, 261-272.
10. Macromedia Inc.: "Macromedia Flash", 2000, <http://www.macromedia.com/software/flash/>
11. Microsoft Inc.: "Microsoft .NET Framework", <http://www.microsoft.com/net>
12. Neumann A., Winter A.: „Kartographie im Internet auf Vektorbasis, mit Hilfe von SVG nun möglich“, *carto:net*, Version 2.0, 09/2000, <http://www.carto.net/papers/svg/index.html>
13. OpenGIS Consortium: "Geography Markup Language, v1.0", May 2000, <http://www.opengis.org/techno/specs.htm>
14. Pfoser D., Jensen C.S., Theodoridis Y.: „Novel Approaches to the Indexing of Moving Object Trajectories“, *Proceedings 26<sup>th</sup> International Conference on Very Large Databases*, Cairo, Egypt, 2000.
15. Sellis T.: "Research Issues in Spatio-temporal Database Systems". *Proceedings 6th International Symposium on Large Spatial Databases*, Hong Kong, China, 1999. In: *Lecture Notes in Computer Science*, Vol.1651, Springer, 5-11.
16. Smallworld: "Smallworld Internet Application Server: Product Overview", Nov. 2000, <http://www.gesmallworld.com/english/products/internet/SIAS.pdf>
17. Sun Microsystems Inc.: "Java™ 2 Platform, Micro Edition (J2ME™)", 2000, <http://java.sun.com/j2me/>
18. Sun Microsystems Inc.: "XML at Sun", <http://xml.sun.com/>
19. Sistla A.P., Wolfson O., Chamberlain S., Dao S.: "Modeling and Querying Moving Objects", *Proceedings 13th International Conference on Data Engineering*, Birmingham, UK, 1997, 422-432.
20. Theodoridis Y., Sellis T., Papadopoulos A.N., Manolopoulos Y.: "Specifications for Efficient Indexing in Spatiotemporal Databases", *Proceedings 10th International Conference on Scientific and Statistical Database Management*, Capri, Italy, 1998, 123-132.
21. World Wide Web Consortium: "EXtensible Markup Language, W3C Recommendation", Febr. 1998, <http://www.w3.org/TR/REC-xml>
22. World Wide Web Consortium: "Synchronized Multimedia Integration Language (SMIL) 1.0 Specification", <http://www.w3.org/TR/REC-smil>
23. World Wide Web Consortium: "XSL Transformations (XSLT), Version 1.0, W3C Recommendation", Nov. 1999, <http://www.w3.org/TR/xslt>
24. World Wide Web Consortium: "Scalable Vector Graphics (SVG) 1.0 Specification, W3C Candidate Recommendation", Aug. 2000, <http://www.w3.org/TR/2000/CR-SVG-20000802/>
25. Wolfson O., Sistla A.P., Chamberlain S., Yesha Y.: "Updating and Querying Databases that Track Mobile Units", *Distributed and Parallel Databases*, Vol. 7, No. 3, 1999, 257-287.