

Raumbezogene Information im Internet: Die "Open Source-Option"

Stefan LEHMKÜHLER & Martin MAY

Dr.-Ing. Stefan Lehmkuhler, Fachgebiet Stadt- und Regionalplanung, Fakultät Raumplanung, Universität Dortmund, Tel: +49 231 755 2379, Fax: +49 231 755 2539, Email: PinkPanther@CommUnity.de
Cand.-Ing. Martin May, CommUnity GmbH, Westenhellweg 77-79, 44137 Dortmund, Tel: +49 231 9143 491, Fax: +49 231 9143 157, Email: gandalf@CommUnity.de

1 EINLEITUNG

Die Verfügbarkeit raumbezogener Informationen im World Wide Web (WWW) ist mittlerweile zu einem akzeptierten und gewünschten Informationselement geworden, welches in seiner Verbreitung mit den vormals nur offline verfügbaren raumbezogenen Informationen – zumindest in der thematischen Breite – ebenbürtig ist. Diese thematische Vielfältigkeit bedeutet leider jedoch nicht, dass z.B. alle – oder zumindest die Mehrheit aller – raumbezogene(n) Informationsangebote entsprechend präsentiert werden. Da die vereinzelt Anwendungen in den unterschiedlichen thematischen Schwerpunkten wie z.B. Tourismus, Verkehr, Wirtschaftsförderung oder raumbezogene Planung belegen, dass technologisch keine (bzw. sehr geringe) Schwierigkeiten existieren, liegt die Vermutung nahe, dass auch im Bereich der nun webbasierenden Präsentation von raumbezogenen Informationen die üblichen Markteintrittsbarrieren – insbesondere finanzielle Hürden – eine angemessene Verbreitung behindern. Dieser Artikel wird daher den kommerziellen Applikationen in den Bereichen „Web-Mapping“ und „3D im WWW“ kostenlose und frei verfügbare Lösungen gegenüber stellen.

MAPSERVER

Obwohl die Sinnhaftigkeit einer abstrahierten zweidimensionalen Präsentation raumbezogener Informationen in Form von Karten seit geraumer Zeit kritisch betrachtet wird, stellt die Integration von Web-Mapping-Lösungen in WWW-Präsenzen zumeist den ersten Schritt zur Integration raumbezogener Informationen dar. Während kommerzielle Software bereits Kosten von zum Teil bis zu 25.000 Euro verursacht (ohne Applikationsentwicklung), stellt die im weiteren vorgestellte Open Source-Lösung „mapserver“, die unter der URL „<http://mapserver.gis.umn.edu/index.html>“ kostenlos erhältlich ist, eine beachtenswerte Alternative dar.

Auch der Funktionsumfang des „mapserver“ ist nach und nach weiter ergänzt worden. Inzwischen werden folgende Modelle der Anbindung zwischen Client und Server unterstützt: Das klassische Modell, also die Grafik und die Webseite werden auf dem Server generiert und als reine html-Seite an den Client gesendet (s. Abb. 1). Die um JAVA / JavaScript ergänzte Variante, bei der zusätzliche Abfragen erleichtert und Hintergrund-Informationen vom Server nachgeladen werden. Als Datengrundlage werden das weit verbreitete Shapefile-Format der Firma ESRI sowie diverse Rastergrafik-Formate unterstützt. Die Anwendung läuft als reguläres CGI-Script und liegt – neben einer Quelltextversion – in Varianten für Linux/Apache sowie für diverse Microsoft-Betriebssystemversuche vor. Kernpunkt der Applikationsentwicklung ist nach erfolgter Installation die Strukturierung und Ergänzung eines sog. Map-Files, welches die wesentlichen Applikations-Informationen enthält. Der Mapserver erzeugt GIF-Graphiken, die zum Client transferiert werden und in allen Browsern dargestellt werden können. Dieses ausgezeichnete Verhältnis zu allen vorhandenen Webbrowsern, die GIF als Standard unterstützen wird allerdings mit erhöhter Bandbreite erkauft. So löst jede Änderung z.B. des Bildausschnitts eine erneute Grafikberechnung auf dem Server und einen erneuten Transfer der erzeugten Graphik zum Client hervor. Als Vorteil dieser Variante bleibt jedoch hervorzuheben, dass die Installation eines Plug-In's oder einer sonstigen proprietären Erweiterung entfällt.

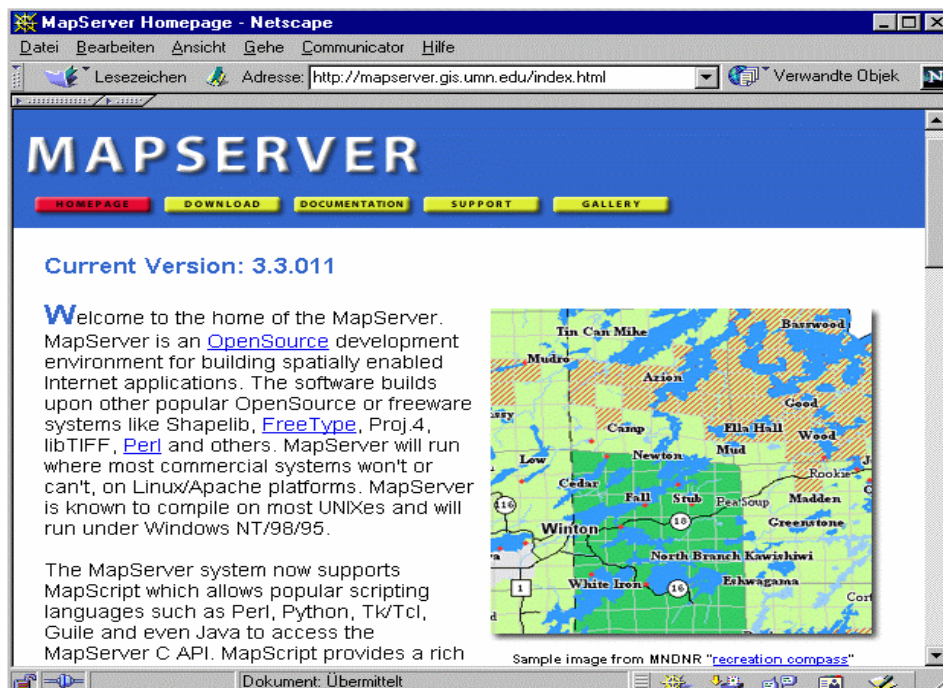


Abb. 1: Mapserver - Homepage der frei verfügbaren Web GIS-Lösung (Screenshot)

Aus dem Kreise der Unterstützer und Nutzer dieser Software wird laufend zur Weiterentwicklung der Software beigetragen, so zeigt Abb. 2 die Integration einer JAVA-basierenden Zoomfunktion. Weitere Neuentwicklungen im Bereich Utilities erleichtern die Erstellung von mapfiles (z.B. Mapserver Workbench, <http://www.nyx.net/~tpoindex/gis/index.html>).

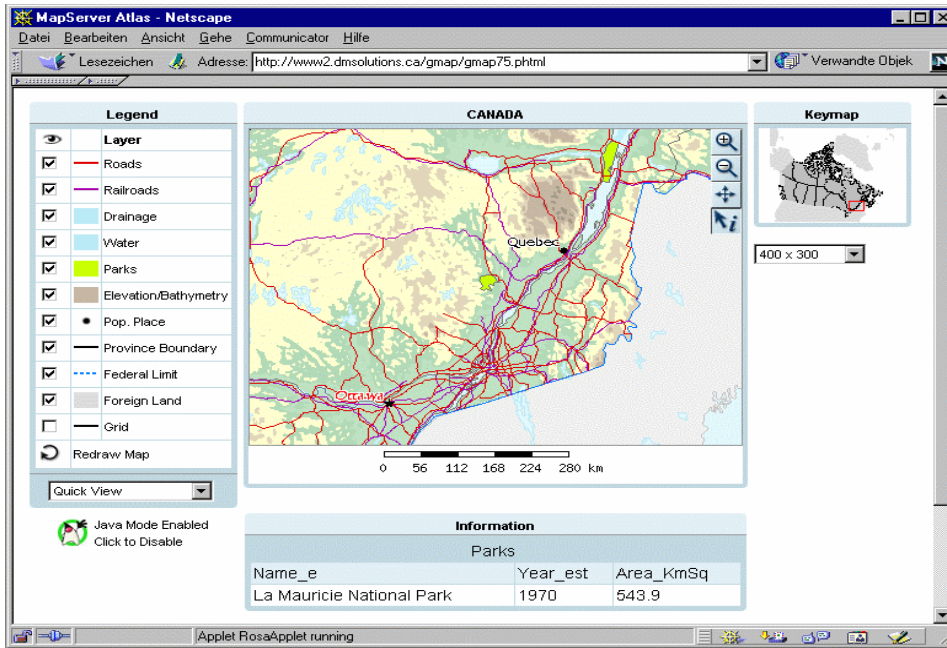


Abb. 2: Mapserver – Darstellung attributiver Daten und Integration weiterer Funktionen mittels JAVA (Screenshot)

Insgesamt stellt der Mapserver eine bemerkenswerte Entwicklung der Open-Source-Community dar, welche die wesentlichen Aspekte der traditionellen Präsentation raumbezogener Informationen sehr gut abdeckt. Die Funktionalität dieser Lösung und ihre universelle Einsetzbarkeit, verbunden mit der Möglichkeit ein verfügbares Budget fast ausschließlich für die eigentliche Applikationsentwicklung einzusetzen, machen dieses Projekt zu einem sehr wertvollen Element der Vermittlung raumbezogener Informationen über das WWW.

3D IM WEB (VRML / X3D / J3D)

1.1 Geschichte und Grundlagen des VRML Standards

Während die computergestützten Analyse- und Visualisierungstechniken GIS und CAD bereits seit geraumer Zeit in der Fachöffentlichkeit bekannt sind, hat Virtual Reality (VR) in Form der Modelliersprache VRML erst eine kurze Vergangenheit. Dieses originär als Dateiformat für dreidimensionale Geometrien entwickelte Konstrukt entstand im Mai 1994 auf Grundlage einer Diskussion auf der ersten internationalen WWW-Konferenz. VRML97 seit nunmehr vier Jahren als ISO-Standard verfügbar, stellt den aktuellen Endpunkt dieser Entwicklung dar.

Analog zur Entwicklung von HTML im World Wide Web – dem Multimediateil des Internets – bildeten auch bei der Entwicklung von VRML die Unabhängigkeit von genutzter Rechnerplattform und Betriebssystem sowie die Integration einer verteilten Informationsbasis die Grundlage für diesen weltweit getragenen Ansatz und gleichzeitig die Chance, ohne Rücksicht auf eine bestehende Klientel, aktuelle Anforderungen nach verteilter Informationsverarbeitung zu realisieren.

Einen ersten Schritt auf diesem Weg stellt die Entwicklung von Anwendungen dar, die auf Grundlage bestehender Vernetzungsinfrastruktur und unter Einsatz dieser Technik die kooperative Bearbeitung raumbezogener Daten ermöglichen. Die bereits 1998 an diesem Ort vorgestellte Applikation VRMLView zeigte, durch die Möglichkeit attributive Daten zu ändern bzw. zu ergänzen, einen wesentlichen Ansatz auf, der nur hier daher kurz angesprochen werden soll¹.

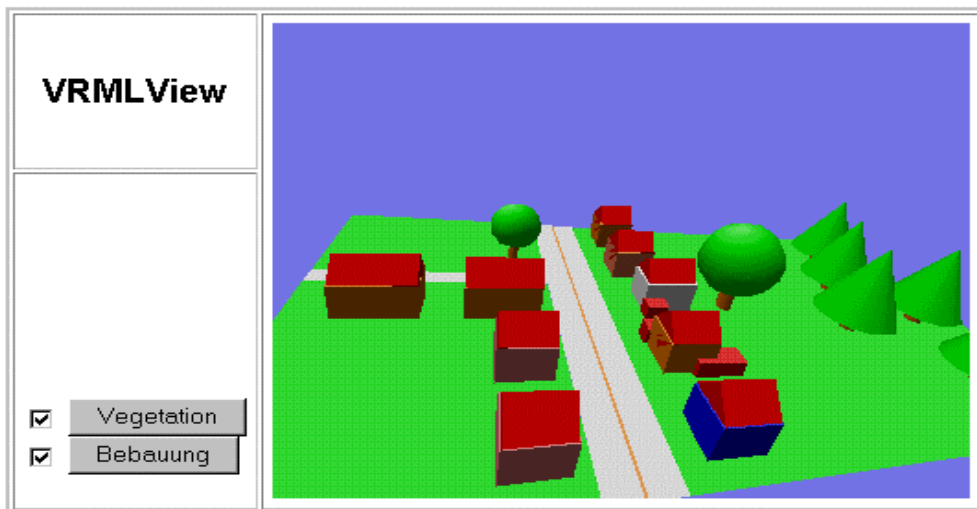


Abb. 3: VRMLView (Screenshot)

Unter Nutzung der beschriebenen Grundlagen demonstrierte die Anwendung VRMLView 1998 die technische Machbarkeit einer VRML-basierten Lösung zur Darstellung raumbezogener Daten (Gebäude, Vegetation, Straßenfläche), zur Bildung von „Themen“ oder „Layern“, deren Verwendung aus „Desktop GIS“-Systemen bekannt ist, und die Einbindung eines digitalen Diskussionsforums. Die Anwendung wurde durch den Aufruf einer Informationsseite im Internet gestartet, in der die Anwendung VRMLView eingebettet ist (vgl. Abb. 3).

1.2 X3D

X3D ist die Kurzform von „Extensible 3D“. Das Web3D-Konsortium hat sich zum Ziel gesetzt, das recht umfangreiche Datenformat VRML in der Version VRML97 auf seine zentralen Elemente zu reduzieren. Man möchte so den Umgang mit diesem Datenformat erleichtern und durch die Reduktion auf wesentliche Sprachelemente die Möglichkeit geben, neue performantere Software zu entwickeln.

Grundsätzlich hatten sich trotz der diversen Vorteile, die auch durch vorsichtige Weiterentwicklung der Dokumentsprache VRML zustande kamen, einige Kritikpunkte herauskristallisiert, von denen folgende die beiden wesentlichen sind:

- VRML ist trotz der Lesbarkeit im Quelltext eine „eigene“ Sprache, deren Sinninhalte sich dem Leser zwar erschließen, dennoch wird meist ein Export oder eine Konvertierung nach VRML zur Generierung von VRML genutzt.
- Die Stärken, die VRML bietet, werden so (in ihrer letzten Fassung VRML97) kaum genutzt und die erzeugten Dateien sind meist nur wenig optimiert, was sich wiederum in der Darstellung bemerkbar macht.

Aus dieser Situation und unter Berücksichtigung der allgemeinen Entwicklung hin zur Verwendung von XML als flexibler Dokumentsprache resultieren die wesentlichen Konsequenzen für die Entwicklung von X3D:

- Reduktion: Der Ansatz, zunächst eine Reduktion auf die wesentlichen Bestandteile umzusetzen, um so eine optimale Webfähigkeit sicherzustellen.
- XML: Um dennoch auch anderen Ansprüchen gerecht werden zu können, geht man davon aus, dass es spezielle erweiterte Formen von X3D geben wird. Diese können dann über andere Dokumenttyp-Definitionen in XML leicht realisiert werden und für sich genommen ebenfalls optimal eingesetzt werden.

Dokumenttyp-Definitionen (dtd) stellen von vorneherein das Kernstück von XML dar. Man verwendet für die verschiedensten Zwecke ein und dieselbe Syntax, jedoch werden die Inhalte von xml-Dateien (Extension *.xml) in zusätzlichen Dateien (Extension *.dtd) definiert. Die zentralen Bestandteile von X3D-Dateien sind somit in der folgenden Datei eindeutig definiert:

<http://www.web3d.org/TaskGroups/x3d/translation/x3d-compromise.dtd>

Beispielsweise erarbeitete das schon länger bestehende Projekt GEOvrml eine eigene Erweiterung dieses Formates, welches in der folgenden Datei festgehalten wird:

<http://www.web3d.org/TaskGroups/x3d/translation/GeoVrml.dtd>

Da sowohl XML als auch VRML eine baumartige Struktur haben, waren die Formate von vorneherein verwandt. Als zusätzlichen Vorteil kann man nun XML-Editoren für die Bearbeitung von X3D-Dateien benutzen, was zwar immer noch eine Arbeit direkt am Quelltext bleibt, dennoch wird die Struktur einer solchen Datei durch den XML-Editor ebenfalls baumartig dargestellt (vgl. Abb. 4).



Abb. 4: Verfügbare Tools – Hier: X3D-Edit (Screenshot)

¹ Die ausführliche Beschreibung dieser Anwendung ist dem Tagungsband der CORP 1998 zu entnehmen (Lehmkuhler S. 399ff)

Um den geringen Unterschied zwischen VRML und X3D-XML zu verdeutlichen, sei hier ein Auszug aus einer Datei, die eine Box beschreibt angeführt. Das Beispiel stammt von der Web-Site des Web3D-Konsortiums <http://www.web3d.org> und ist eines von Hunderten, die an dieser Stelle eingesehen werden können.

Beispiel 1: VRML

```
Shape {
  appearance Appearance {
    material Material {
    }
  }
  geometry Box {
  }
}
```

Beispiel 2: XML (x3d-compromise.dtd)

```
<Shape>
  <appearance>
    <Appearance>
      <material>
        <Material/>
      </material>
    </Appearance>
  </appearance>
  <geometry>
    <Box/>
  </geometry>
</Shape>
```

Für die Verwendung von X3D gibt es jedoch noch einen entscheidenden anderen Grund: Von vornherein wurde darauf geachtet, dass diverse Software parallel zur Definition von X3D mitentwickelt wurde. So konnten verschiedene Hersteller am Prozess teilhaben und als Resultat ging nicht nur ein neues Dokumentenformat aus dem Prozess hervor, sondern auch nahezu alles, was man an sonstiger Software benötigt:

- Allein lauffähige Viewer (ohne Web-Browser)
- Als Applet lauffähige Viewer für Java 1.1 und Java 2
- Editoren
- Konverter

Insgesamt stellt X3D **das** leistungsfähige Dokumentenformat zur Beschreibung raumbezogener Informationen dar, welches von einer breiten Basis aus kommerziellen Entwicklern und Privatanwendern getragen wird und sich vorzüglich mit anderen XML-Anwendungen kombinieren lässt.

1.3 Java 3D

Dem interessierten Entwickler wird in letzter Zeit die Weiterentwicklung von Java allgemein nicht entgangen sein. In Zusammenhang mit dieser Fortentwicklung gibt es auch immer mehr Erweiterungen, die Java mit zusätzlichen Fähigkeiten ergänzen.

Eine dieser Erweiterungen ist Java 3D, ein sogenanntes Java API (Application-Programming-Interface), welches grundlegende Sprachbestandteile zur Entwicklung von 3D-Applikationen enthält. Offiziell definiert die Firma SUN, die sowohl Java als auch Java3D entwickelt hat bzw. weiterentwickelt, Java3D folgendermaßen: „Java 3D is a network-centric, scene-graph-based API for developing 3D applets an applications.“

Somit soll die Philosophie, die Java zu dem plattformübergreifenden Software-Standard schlechthin gemacht hat, auch für Java 3D Gültigkeit besitzen: Man schreibt eine Java-Software einmal und kann sie dann auf allen Plattformen laufen lassen, die Java bzw. Java 3D unterstützen.

Eine besondere Stärke geht aus der Deklaration „scene-graph-based“ hervor, die in folgender Weise zu verstehen ist: Bei den Entwicklungen von 3D-Software sind in den letzten Jahren verschiedene „evolutionäre“ Schritte vollzogen worden. Java 3D wird von Software-Experten als mittel- bis hochwertige API der vierten Generation klassifiziert. Der Unterschied zu Software der dritten Generation ist nun der, dass dort die Objekte nach „display-lists“ organisiert wurden. Durch die Organisation der Objekte in einem Szenen-Graphen wird das Rendern (also die Darstellung der Objekte auf dem Bildschirm) weitgehend automatisch möglich (vgl. Abb. 5) und erheblich effizienter. Der Entwicklungsprozess wird somit erheblich flexibler und die Inhalte können erheblich leichter verarbeitet werden.²

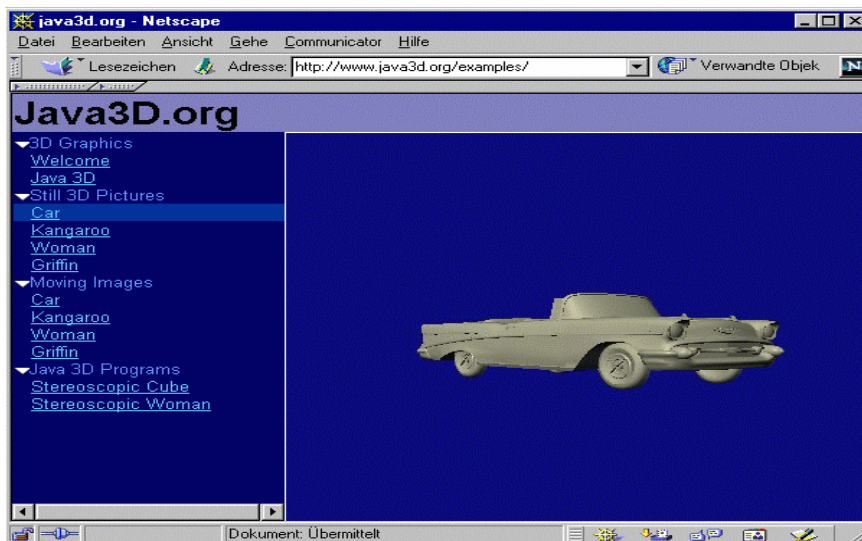


Abb. 5: Beispielanwendung (Screenshot)

² Barrilleaux, S. 203f.

Entscheidende Frage bei der Verwendung von 3D-Software war bisher immer das verwendete Datenformat. Java 3D ist nicht mit einem ‚eigenen‘ Datenformat entwickelt worden. Vielmehr gibt es verschiedene sogenannte „Loader“, die das Laden und Speichern auch verschiedener Datenformate ermöglichen. Diese Loader können auch für neue Datenformate entwickelt werden und so sind auch für X3D-Dateien schon mehrere verschiedene Loader verfügbar.

Inzwischen sind zusätzlich zu dem eigentlichen Java 3D (der sogenannten Core-API) weitere Pakete entwickelt worden, die grundlegende Elemente der 3D-Programmierung erleichtern: Es gibt Classes, die einfache Objekte erzeugen können, die zusätzliche Aufgaben, wie Texturen laden, erleichtern und nicht zuletzt Entwicklungen, um in 3D-Welten besser navigieren zu können.

Für raumplanerische Belange gibt es folgende Aspekte, die für die Verwendung von Java3D sprechen:

- Java3D verwendet als Ausgangsbasis sogenannte „Universen“, in denen Meter als Grundeinheit für Längenmaße verwendet wird.
- Genau wie Java ist auch Java3D objektorientiert, was bedeutet, dass im Grunde jedes Objekt auch als aktive Komponente zu sehen ist. Somit können hier spezielle Eigenschaften von bestimmten Objekten direkt im Objekt selber verankert werden, beispielsweise wechselnde Zustände, sich bewegende Objekte oder die Reaktion auf einen Eingriff. Denkbar wären auch Zusammenstöße oder sonstige Wechselwirkungen und Beeinflussungen.

FAZIT

Es bleibt festzuhalten, dass sowohl im traditionellen Bereich der Vermittlung raumbezogener Informationen mittels zweidimensionaler Darstellung (Karten) über das WWW, als auch auf dem Feld der fortgeschrittenen Visualisierung z.B. dreidimensionaler Objekte, leistungsfähige kostenlose und zum Teil im Quelltext verfügbare Lösungen existieren. Implementationen dieser Techniken bieten mindestens die wesentlichen Leistungsmerkmale ihrer kommerziellen Gegenstücke und befinden sich – getragen durch eine breite Unterstützer- und Anwender-Community – in stetiger Weiterentwicklung.

QUELLEN

Barrilleaux J.: 3D User Interfaces with JAVA 3D, Greenwich 2001.

Lehmkuhler St.: Virtual reality Modeling Language, in: Schrenk (Hg.) Computergestützte Raumplanung, S. 399ff, 1998 Wien.

Diverse Web-Sites wie im Text angegeben.