

## Interaktion im virtuellen Raum durch Echtzeitkopplung mit 2D-GIS

Johannes BRAUNER, Martin MAY & Ilja ABRAMOVIC

Institut für Geoinformatik der Westf. Wilhelms-Universität Münster,  
Robert-Koch-Str. 26 - 28, D-48149 Münster, {joejoe,mmay,abramov}@ifgi.uni-muenster.de

### 1 EINLEITUNG

In den letzten Jahren gibt es im Umfeld der Geovisualisierung einen starken Trend hin zur 3D-Echtzeitvisualisierung und damit einhergehend auch zur Interaktion im virtuellen Raum. Über aufwändige Modelle muss die Verbindung zwischen Eingabegeräten, Fachdomäne und virtueller Welt erstellt werden. Erschwerend kommt hinzu, dass Eingabegeräte im Normalfall nur über zwei Freiheitsgrade verfügen. Betrachtet man die Anforderungen, die sich speziell im Umfeld der Raumplanung ergeben, so ist eine solche 3D-Interaktion nicht in jedem Falle erforderlich. Der Abstraktionsgrad einer 2D-Visualisierung erleichtert nicht nur die Einbindung von bestimmten Interaktionen sondern auch den Umgang mit solchen Systemen. Dennoch wird ein Echtzeit-Erleben von getätigten Eingaben ermöglicht. Dieser Beitrag fokussiert, wie eine dreidimensionale Umgebung für bestimmte Anwendungsfälle und -schritte durch eine entsprechende 2D-Komponente ergänzt werden kann. Änderungen in der einen Umgebung werden bei fachlich sinnvollen Bearbeitungen in Echtzeit auch in der anderen angezeigt.

Hierbei wird soweit möglich auf Datenstrukturen aus dem 2-Dimensionalen zurückgegriffen, die beispielsweise in Form von Texturen in der 3D-Umgebung zum Einsatz kommen. 3D-Daten kommen nur in Form von Visualisierungsobjekten, jedoch nicht als 3D-Geodaten vor. Dies ist jedoch keine prinzipielle Beschränkung der Applikation. Die benötigten Daten für die 3D-Echtzeitumgebung werden soweit möglich standardisiert lokal oder netzbasiert vorgehalten. Die editierten zweidimensionalen Geodaten können in ihrem Ausgangsformat (beispielsweise Shapefiles) abgespeichert werden und sind somit auch für andere Anwendungen bzw. innerhalb einer Geodateninfrastruktur verfügbar.

Über die reine Visualisierung hinaus bieten beide Komponenten ihrer Dimensionalität entsprechende GIS-Funktionalitäten. In einem Anwendungsfall aus der Landschaftsplanung wird eine derartige Interaktion praktisch umgesetzt und gezeigt, wie diese in eine heterogene System-Architektur eingebunden werden kann.

### 2 ANWENDUNGSFALL IN FLUMAGIS

FLUMAGIS ist der Prototyp eines Planning Support Systems (PSS) zur Unterstützung von Planungsentscheidungen bei der Umsetzung der Wasserrahmenrichtlinie (WRRL) der Europäischen Union (<http://europa.eu.int/scadplus/leg/de/lvb/l28002b.htm>, zuletzt abgerufen am 30.11.04). Ziel der EG-WRRL ist das Erreichen eines „guten ökologischen Zustands“ aller Fließgewässer im Gebiet der Europäischen Union. Im System werden für Gewässereinzugsgebiete Methoden und Visualisierungen zur Bearbeitung und Simulation/Prognose von Planungsentscheidungen bereitgestellt. Hierzu werden verschiedene hydrologische, vegetationskundliche, sozioökonomische und limnologische Modelle in das System eingebunden, die aufgrund vorhandener (Geo-)Daten Ergebnisse produzieren, die entsprechend visualisiert werden müssen. Visualisierungen sind sowohl im zwei- als auch im dreidimensionalen Raum möglich. Das System basiert auf einer Client-Server Architektur. (vgl. Möltgen & Streit 2004)

In FLUMAGIS ist neben der in Geoinformationssystemen (GIS) üblichen zweidimensionalen Darstellung für bestimmte Anwendungsfälle auch eine dreidimensionale Visualisierungsumgebung notwendig. Um beispielsweise ein Geländemodell mit Höheninformationen zu visualisieren, ist eine 3D-Umgebung unumgänglich. Darüberhinaus bietet die 3D-Umgebung eine für die Planungsbeteiligten realitätsnähere und „anfasserbarere“ Visualisierung. So wird der Akzeptanzgrad von Planungsentscheidungen erheblich verbessert und somit die Kommunikation im Planungsprozess gravierend vereinfacht. Eine gute Kommunikation kann Planungsprozesse verbessern (vgl. Lehmkuhler 1999, S. 121).

Für eine Prognose des Zustands eines Fließgewässers bei Änderung der entsprechenden Landnutzungsart in der Flussaue müssen mit Hilfe entsprechender GIS-Funktionalitäten verschiedene Ausgangsszenarien modelliert und vorbereitet werden können. Deshalb ist es erforderlich, dass das System über die reine Visualisierung hinaus diese GIS-Funktionalitäten zur Editierung von Geometrien und Attributen zur Verfügung stellt. In unserem Anwendungsfall geht es um die Änderung der Landnutzung von intensivem Ackerbau zu extensiv genutztem Wald, damit der Eintrag von düngungsbedingtem Nitrit verringert wird. Zur Erzeugung der Szenarien ist es notwendig, dass polygonale Flächen editiert und in der Größe verändert werden können.

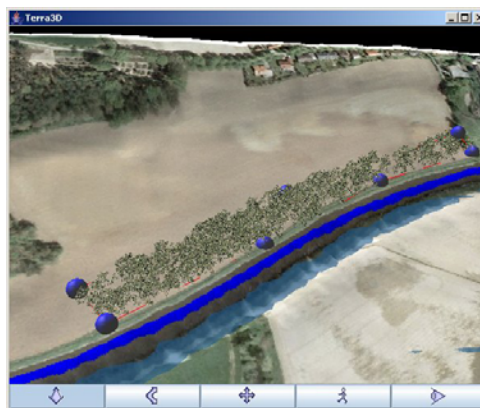


Abb. 1: Erste Implementierung eines 3D-Clients (Quelle: Eigene Darstellung)

Für die Bearbeitung der Fragestellung wird in der ersten Umsetzung aus verschiedenen Gründen eine großmaßstäbige dreidimensionale Umgebung geschaffen. Um fachlich sinnvoll arbeiten zu können, benötigt man dreidimensionale Geländemodelle, die verschiedene besondere Geländestrukturen (beispielsweise Terrassenkanten der Aue) identifizierbar zu machen. Weiterhin ist der Wald neben dem Gewässer das wichtigste Objekt, weil es den Baum als größte Entität des Lebensraums enthält. Zur kooperativen Planung sollte der Wald realitätsgetreu modelliert werden, damit die erzeugte Visualisierung qualitativ überzeugen kann.

Erste Tests mit der 3D-Umgebung (s. Abb. 1) zeigten auf, dass der Schwierigkeitsgrad der Interaktion im dreidimensionalen Raum deutlich höher liegt, als im zweidimensionalen. Mit den Standardeingabegeräten (in diesem Fall die Computermaus) ist es für den Benutzer schwierig zu unterscheiden, in welche Richtung er die Knoten des Polygons bei der Editierung bewegt. Ist es die Ebene, in der das Polygon liegt, oder verändert er die Knoten in der Höhe? Grundsätzlich gibt es in der 3D-Interaktion verschiedene Voraussetzungen zu beachten (siehe Kapitel 4). Die Anforderungen für eine erfolgreiche Interaktion sind für den Benutzer höher.

In einer 2D-Umgebung (s. Abb. 2) sind Polygoneditierungen unproblematischer. Bei der Editierung (beispielsweise Verschiebung) von Knoten ist eine Interaktion ohne Betrachtung von Voraussetzungen und komplizierten technischen Feinheiten möglich. Sie findet in einer für den Anwender „gewohnten“ Umgebung statt. Zusätzlich ist es für die Bearbeitung des Anwendungsfalls ausreichend, Polygone im zweidimensionalen Raum zu erstellen und zu bearbeiten. Allerdings fallen die o. a. Vorteile der höheren Realitätstreue durch die 3D-Visualisierung weg.

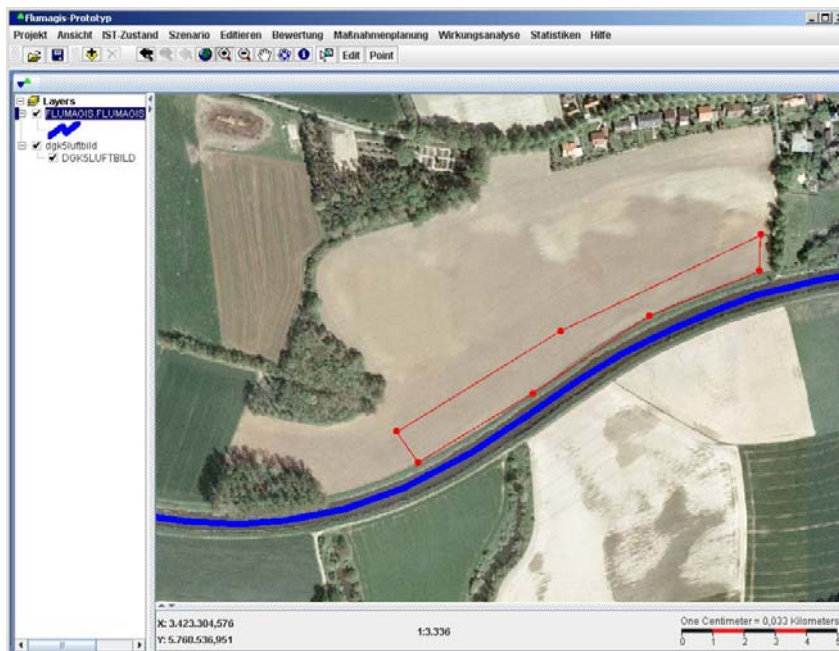


Abb. 2: Gleichzeitige Darstellung im 2D-Client (Quelle: Eigene Darstellung)

Um die hohen Anforderungen an den Benutzer und die zusätzlichen technischen Voraussetzungen einer reinen 3D-Visualisierung abzumildern, wird für die Interaktion zusätzlich eine „normale“ 2D-GIS-Umgebung inklusive Visualisierung hinzu gekoppelt. Somit werden die Vorteile der vereinfachten Interaktion in der 2D- und die Vorteile der besseren Visualisierung im 3D-Bereich kombiniert. Beide Visualisierungen werden gleichzeitig angezeigt und Veränderungen in einen Raum synchron und ohne Zeitverzug in der jeweils anderen Umgebung aktualisiert.

May et al. 2004 geben eine genauere Übersicht über die Systemspezifikation in FLUMAGIS.

### 3 THEORETISCHE GRUNDLAGEN DER 3D-GEOVISUALISIERUNG BZW. 3D-VISUALISIERUNG

Aus den Entwicklungen des Forschungsfelds Computergraphik in den letzten Jahrzehnten ist das Arbeitsfeld der interaktiven 3D-Visualisierung entstanden.

Geovisualisierung stellt besondere Anforderungen an die Visualisierung. Charakteristika von Geodaten und darauf operierenden Methoden müssen berücksichtigt werden. Hierfür ist es beispielsweise erforderlich, dass alle visualisierten Objekte einen exakten Raumbezug vorweisen und das Visualisierungssystem zusätzlich einfache GIS- und kartographische Funktionalitäten bereitstellt. „Ziel der Forschungs- und Entwicklungsaktivitäten im Umfeld der Geovisualisierung ist die Bereitstellung theoretischer Grundlagen, Methoden und Werkzeuge für den Aufbau entsprechender Visualisierungsumgebungen zum Zweck der Exploration, Analyse, Synthese oder Präsentation raumbezogener Daten.“ (Schmidt 2002, S. 13)

Für die theoretische Betrachtung von 3D-Geovisualisierung sind zwei Aspekte von besonderer Bedeutung:

Wie werden meine Geobjekte dargestellt?

Wie kann ich mit den visualisierten Geobjekten interagieren?

Für die Interaktion in der 3D-Visualisierungsumgebung gibt es verschiedene Möglichkeiten. Sogenannte Metaphern umschreiben die Interaktionsmöglichkeiten. Kapitel 4 beschreibt u. a. einige dieser Metaphern und die vorhandenen Hardware-Lösungen.

In der klassischen 3D-Visualisierung nutzt man das Prinzip der Visualisierungspipeline (Haber & McNabb 1990) zur Generierung der 3D-Darstellung. Eine Visualisierungspipeline besteht aus drei streng aufeinander abfolgenden Arbeitsschritten: Selektion/Filterung, Mapping und Rendering. Für 3D-Geovisualisierung müssen spezielle Aspekte beachtet werden.

Für die Darstellung der Geoobjekte im virtuellen Raum muss als erstes überlegt werden, welche Geoobjekte für die fachliche Fragestellung von Bedeutung sind. Die Darstellung von allen vorhandenen aber nicht im Sinne der Problemstellung unbedingt bedeutsamen Geoobjekten ist nicht sinnvoll. Werden zu viele Geoobjekte zeitgleich angezeigt, wird der Anwender schnell überfordert und von den für ihn interessanten Geoobjekten abgelenkt. Zusätzlich gilt es, die Anzahl der Geoobjekte aus reinen Performanzgründen möglichst gering zu halten. Dieser Schritt wird als „filtern“ beschrieben. Im zweiten Schritt muss überlegt werden wie die nun „gefilterten“ Geoobjekte sinngemäß visualisiert, also auf entsprechende Visualisierungsobjekte „gemappt“ werden. Dieser Schritt ist erforderlich, damit unterschiedlich dimensionierte Geoobjekte realitätsgetreu visualisiert werden können. (vgl. Schmidt 2002, S. 64f)

Beim Mapping wird noch keine Sichtbarkeitsanalyse im Sinne der Computergraphik durchgeführt, sondern ausschließlich thematisch entschieden, welche Geoobjekte überhaupt von Bedeutung sind. Durch sogenanntes Rendering sämtlicher vorhandener Visualisierungsobjekte entsteht ein wahrnehmbares Bild für den Anwender. Erst beim Rendering wird eine Sichtbarkeitsanalyse durchgeführt und Bäume, die von der Kameraposition nicht mehr sichtbar oder nur zum Teil sichtbar sind, nicht oder nur im sichtbaren Teil visualisiert. Hier geht es um rein optische Aspekte und die Betrachterposition.

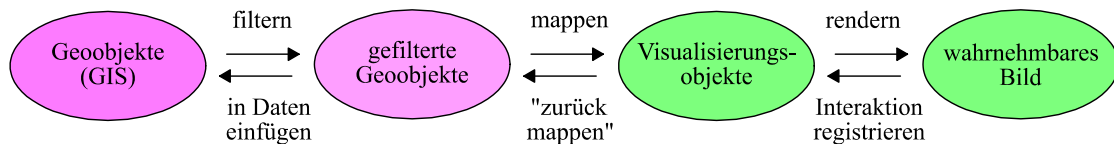


Abb. 3: Schritte der 3D-Visualisierung nach Schmidt 2002 (Quelle: Eigene Darstellung)

Für die Visualisierung beispielsweise der Landnutzung in einer Flussaue liegen die Daten in zweidimensionalen Polygonen vor. Für zweidimensionale Geoobjekte (in diesem Fall die Landnutzung Wald in einer bestimmten Fläche) benötigt man keine dreidimensionale Darstellung. Für die 3D-Darstellung wird vom Polygon „Wald“ auf einzelne dreidimensionale Baummodelle gemappt, die innerhalb des räumlich referenzierten „Walddpolygons wachsen“. Alle Bäume bilden den Wald, den sich der Anwender vorstellen soll. Auch die Rückrichtung vom Visualisierungsobjekt zum Geoobjekt muss im Falle einer Interaktion betrachtet werden. Die visualisierten Objekte müssen wieder auf Geoobjekte zurück gemappt und die Geoobjekte wieder in die vorhandene Datengrundlage eingefügt werden. Wird beispielsweise der Wald in der flächenhaften Ausdehnung geändert (die Verteilung der Visualisierungsobjekte „Bäume“ wird verändert) so müssen diese wieder in ein verändertes Landnutzungspolygon umgewandelt und in die Datengrundlage eingepflegt werden. Abb. 3 verdeutlicht dieses Schema. Die Trennung von Originaldaten und Visualisierungsdaten gibt sich durch die unterschiedliche Farbgebung.

#### 4 GEO- UND INTERAKTIONSMODELLE

Neben den erweiterten Datenmodellen - dreidimensionale Daten haben zusätzlich zu X- und Y- noch eine Z-Koordinate für den Höhenwert - müssen für eine erfolgreiche Interaktion sechs statt zwei Freiheitsgrade der Objekte kontrolliert werden können. Freiheitsgrade beschreiben die Möglichkeiten, wie Objekte im Raum bewegt bzw. gedreht werden können. Je mehr Dimensionen, desto mehr Richtungen in die verschoben werden kann und je mehr Achsen, um die man drehen kann. Für eine volle Kontrolle über alle sechs Freiheitsgrade im Dreidimensionalen muss teure Hardware vorhanden sein. Es gibt verschiedene Lösungen (genannt Metaphern) im immersiven (beispielsweise Cave) und semi-immersiven Bereich (z. B. 3D-Workbench). Zusätzlich braucht man Zeigergeräte, die alle sechs Freiheitsgrade unterstützen (beispielsweise 3D-Maus o. ä.). Für den Einsatz in der Planungspraxis sind derartige Systeme momentan aus verschiedenen Gründen nicht effektiv einsetzbar. Kosten (bis in den sechsstelligen Eurobereich) und hoher Administrations- und Programmieraufwand sprechen dagegen.

Um auf 3D-Visualisierung aber nicht verzichten zu müssen, gibt es andere Lösungen für Standard-Desktop-Arbeitsplätze. Mit der „normalen“ Computermaus können zwei Freiheitsgrade kontrolliert werden (im Sinne der X- bzw. Y-Achse jeweils in beide Richtungen). Folglich müssen vier fehlende Freiheitsgrade ausgeglichen werden. In der Geovisualisierung kann man sich zu Nutze machen, dass Geoobjekte nicht nur durch Geometrie (absolute Lage im Raum) und Topologie (die Lage der Geoobjekte zueinander), sondern auch durch Attribute und Semantik thematisch definiert sind. Durch diese thematischen Eigenschaften lassen sich sogenannte Constraints definieren, die beschreiben, was man mit den Geoobjekten „machen darf“ und wie sich im Raum verhalten. Constraints können geometrischer, topologischer oder struktureller Natur sein. Folgendes Beispiel zur Verdeutlichung: Ein Baum hat immer einen festen Punkt in einer Ebene (dort wo er wächst). Man kann also für die Interaktion die Höhe außer Betracht lassen („ein Baum kann nicht in der Luft schweben“), denn es interessiert nur der Punkt, wo er in der Ebene verankert ist. Für die Visualisierung ist die Z-Koordinate hingegen wichtig, da die Höhe des Baums sonst nicht modelliert werden kann. „Der Benutzer kann also nur so mit einem Geoobjekt interagieren, wie es dem Verhalten seines 'realen Gegenstücks' entspricht.“ (Merten 2002, S. 4)

Um im beschriebenen Anwendungsfall die Interaktion in der 3D-Umgebung zu ermöglichen, müssen die zugrunde liegenden Daten genauer analysiert werden. Für die interaktive geometrische Änderung von Landnutzung reicht es aus, sich auf zweidimensionale Landnutzungspolygone zu beschränken. Der einzelne Baum ist für die Interaktion nicht von Interesse, ebenso wenig der Höhenwert. Ein Polygon besteht aus Knoten (Eckpunkten), die Knoten verbindenden Kanten (Begrenzungslinien) und einer durch die Knoten und Kanten definierten Innenteil. Das Polygon liegt achsenparallel zur X- und Y-Achse und ist in einem Punkt mit dem darunterliegenden Geländemodell verankert. Dies ist der Constraint der auf dem betrachteten Anwendungsfall beruht. Wenn nun das Landnutzungspolygon in der Größe verändert wird (beispielsweise durch Verschieben der Knoten innerhalb der aufgespannten Ebene) wird für die neue Fläche ein Wald generiert, der aus einzelnen Visualisierungsobjekten „Baum“ besteht und durch das

Polygon an sich begrenzt wird. Diese Eigenschaft erlaubt eine gleichzeitige Darstellung und Interaktionsmöglichkeit in der 2D-Umgebung, da dort ebenfalls auf reinen zweidimensionalen Polygonen operiert wird.

Im Fall von FLUMAGIS wird die Interaktion zusätzlich in der 3D-Umgebung gewährleistet. Das Polygon wird in der oben beschriebenen Ebene aufgespannt und die Knoten durch Sphären dargestellt, die innerhalb der Ebene verschoben werden können. Die Visualisierung von 3D-Interaktion im 2D-Darstellungsraum ist ebenso synchronisiert.

## 5 ARCHITEKTUR

Die Implementierung des Anwendungsfalls basiert auf zwei Architekturmustern (auch „Pattern“) und dem objektorientierten Programmier-Paradigma (einen guten Überblick bietet Oestreich 2001). Pattern sind Architekturmuster zur optimierten Lösung von abgrenzbaren oft auch sich wiederholenden Teilproblemen innerhalb eines Softwaresystems (vgl. Gamma et al. 1995). Sie erleichtern die Implementierung und Wartbarkeit von Programmcode. Das System ist getreu dem Model-View-Controller Pattern (Eilebrecht & Starke 2004 S. 55 ff., Krutscher 1997), kurz MVC aufgebaut. Die Kommunikation zwischen der 2D- und 3D-Umgebung folgt dem Observer-Pattern (Cooper 1998 S. 177 ff., auch Beobachter-Muster bei Seemann & v. Gudenberg 2000 S. 190 ff.). Mit dem Observer-Pattern lässt sich das Problem lösen, dass mehrere Objekte vom Zustand eines anderen Objekts abhängig sind. Wenn sich der Zustand dieses dritten Objekts ändert, werden die abhängigen Objekte entsprechend benachrichtigt.

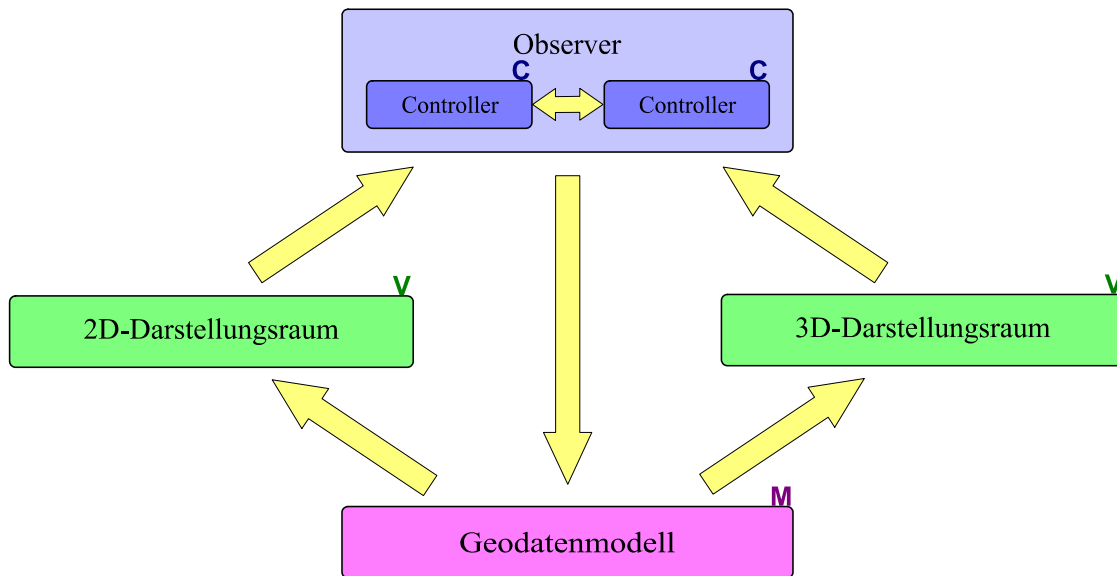


Abb. 4: Architekturmodell (Quelle: Eigene Darstellung)

Anhand von Abb. 4 wird die verwendete Architektur ersichtlich. Die unterschiedliche Farbgebung und die Einzelbuchstaben repräsentieren die verschiedenen Systemeinheiten Model (Rosa), View (Grün) und Controller (blau). Sowohl die 2D- als auch die 3D-Applikation sind in sich eigenständige Komponenten, die getrennt voneinander jeweils nach dem MVC-Pattern entwickelt worden sind. Durch objektorientierte Programmierung, die auf bestmögliche Erweiterbarkeit ausgelegt ist, war es einfach beide Komponenten zu koppeln. Beide operieren allerdings im Falle der Kopplung auf dem selben Modell, welches das Geodatenmodell (im Anwendungsfall das Landnutzungspolygon) und entsprechende Funktionalitäten zur Bearbeitung der Daten bereitstellt. Der Anwender interagiert direkt mit einem der beiden Darstellungsräume. Jeder Darstellungsraum wird auf Benutzereingaben durch einen eigenen Controller überwacht, der Veränderungen registriert. Ist eine Interaktion aus einer der beiden Darstellungsräume gemeldet worden (Eingabe), wird vom Observer das Geodatenmodell entsprechend geändert und auf Basis des Datenmodells eine neue Ausgabe für den jeweils anderen Darstellungsraum angefordert. Durch den Observer wird also gewährleistet, dass die Ausgabe in beiden Darstellungsräumen synchronisiert wird und die gemeinsam verwendeten Geodaten entsprechend aktualisiert werden. Jeder Darstellungsraum ist für sich dafür verantwortlich, entsprechend der Änderungen im Datenmodell eine geeignete Visualisierung zu erstellen. Für die 3D-Darstellung ist dies in Kapitel 3 und 4 beschrieben, die Visualisierung im 2D erfolgt nach „bewährten“ Prinzipien, die nicht Gegenstand dieses Papiers sind.

## 6 UMSETZUNG

Abb. 5 zeigt die Eingliederung und Umsetzung des Anwendungsfalls in den FLUMAGIS-Prototyp. Für die Visualisierung sind die voneinander unabhängigen Clients für den 2D- und 3D-Darstellungsraum zuständig. Diese sind über eine Controller-Komponente mit der Serverseite verbunden. Der Controller überwacht die Interaktionen des Benutzers aus beiden Clients und löst entsprechende Befehle auf der Serverseite aus. Die Serverseite ist nicht als „ein“ Server zu verstehen, sondern aus verteilten Servern mit jeweils gekapselten Funktionalitäten. Die Geodatenbank hält die Geodaten vor. Über die Wissensbasis sind die verschiedenen Modelle zur Simulation, Analyse und Prognose angebunden (s. Kapitel 2). Das komplette System mit Ausnahme der Geodatenbank ist in Java implementiert worden. (vgl. May et al. 2004)



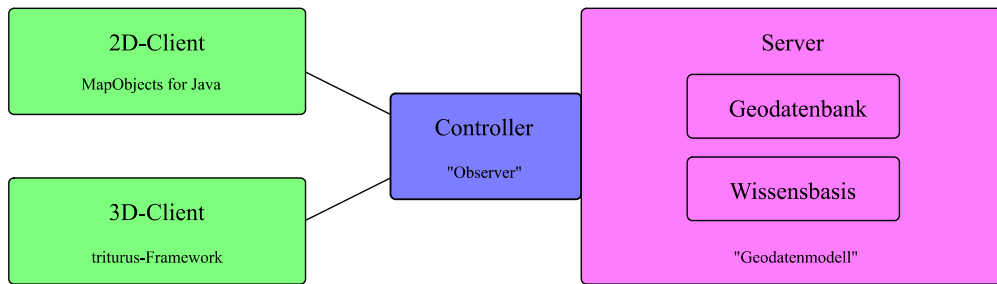


Abb. 5: Umsetzung des Anwendungsfalls in FLUMAGIS (Quelle: Eigene Darstellung)

Der 2D-Client wurde mit "Map Objects for Java 2" der Firma ESRI entwickelt. Map Objects stellt Bibliotheken, vorgefertigte Visualisierungskomponenten und Datenstrukturen vorprogrammiert zur Verfügung. Nach dem Baukasten-Prinzip kann einfach eine Applikation zusammengestellt werden. Map Objects ist unter einer proprietären Lizenz veröffentlicht. Die offene Systemarchitektur ermöglicht, dass künftig auch Alternativen zu Map Objects aus dem Bereich der Freien Software angebunden werden. Erste diesbezügliche Versuche werden zur Zeit durchgeführt.

Die 3D-Visualisierungsumgebung basiert auf dem triturus Framework (Schmidt et al. 2004) des Instituts für Geoinformatik (Universität Münster) und der conterra GmbH (Münster). triturus stellt 3D-Datenmodelle zur Verfügung und kapselt die Daten von der jeweiligen Visualisierungengine. Für die Visualisierung können beliebige auf Szenengraph-APIs basierende Renderengines, beispielsweise Java3D oder Xith3D, angebunden werden. In triturus wird Java3D der Firma Sun Microsystems verwendet. Für die Visualisierungsobjekte „Baum“ werden Modelle der Firma Xfrog verwendet, die insbesondere in Größe und Wuchsform mittels eines Simulationsmodells realitätsnah erzeugt wurden. Die Baumart wird durch Attribute im Geodatenbestand festgelegt.

Das zu bearbeitende Landnutzungspolygon wird dynamisch auf die Geländeoberfläche aufgelegt, so dass es in jedem Fall oberhalb der Selbigen liegt. Hierzu sind entsprechende geometrische Operationen notwendig, die bei jeder Veränderung durchgeführt werden müssen.

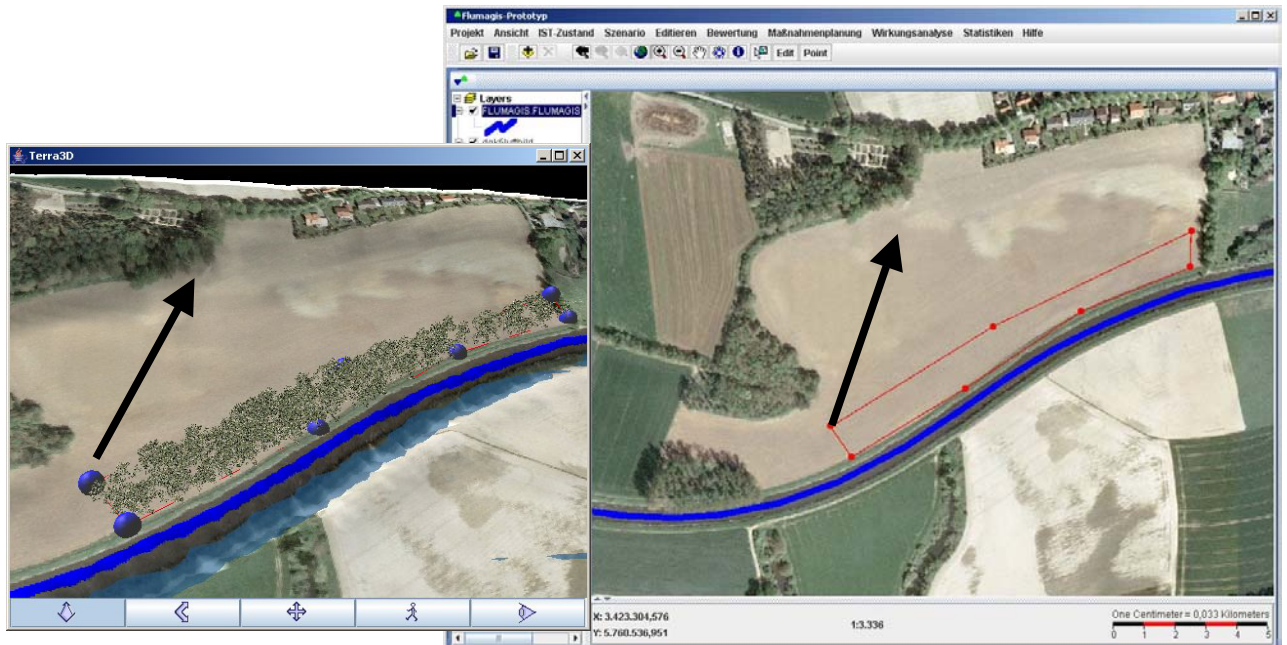


Abb. 6: Synchrone Visualisierung in beiden Darstellungsräumen (Quelle: Eigene Darstellung)

Während einer durch eine Interaktion (sei es zur Veränderung der Betrachterposition oder Editierung) verursachten Neuberechnung des Bildes werden sämtliche Visualisierungsobjekte durchlaufen (entsprechend dem Iterator-Pattern in Eilebrecht & Starke 2004, S. 36 ff.) und entsprechend der neuen Situation gerendert. Im Falle einer Polygoneditierung wird zusätzlich das Geodatenmodell entsprechend aktualisiert und für das Visualisierungspolygon die Anzahl der Bäume angepasst.

Da die Clients für die beiden Darstellungsräume und der Observer auf dem selbem Clientrechner ausgeführt werden, ist die Verwendung eines speziellen Kommunikationsprotokolls nicht notwendig. Die Clients und er Observer tauschen ihre Informationen java-intern aus.

In Abb. 6 wurde beispielhaft ein Uferrandstreifen mit Wald angelegt, der landschaftsökologisch gesehen einen Puffer zwischen der argrarökonomisch genutzten Ackerfläche und dem Gewässer (in diesem Fall die Ems bei Eiden) bilden soll. Die beiden Pfeile deuten eine Änderung des am Weitesten links liegenden Knoten und seine synchrone Visualisierung in beiden Darstellungsräumen an.

## 7 FAZIT

Durch Verringerung der Dimensionalität des Darstellungsraums wird die Interaktion vereinfacht. Die 2D-Darstellung ist hierbei als weiter abstrahiert zu betrachten. Die 3D-Interaktion stellt höhere Anforderungen an den Benutzer. Das Ziel der Visualisierungs-/Interaktionskopplung beider Darstellungsräume: Dem Nutzer in seiner gewohnten Domäne entgegen kommen und somit die Usability von computergestützten Planungswerkzeugen erhöhen.

	<i>Abstraktion</i>	<i>Interaktion</i>
<i>2D-Darstellung</i>	Hoch	Einfach
<i>3D-Darstellung</i>	Gering	Komplex

Tab. 1: Eigenschaften der einzelnen Darstellungsräume bzgl. Abstraktion und Interaktionskomplexität (Quelle: Eigene Darstellung)

Durch die synchrone Koppelung von 2D- und 3D-Darstellung bei gleichzeitiger Operation auf den selben Daten werden die jeweiligen Vorteile (dargestellt in Tab. 1) der einzelnen Darstellungen kombiniert. „Die dreidimensionale Visualisierung eines Planungsentwurfs ist ein geeignetes Mittel, um die visuelle Kommunikation im Planungsprozess zu erleichtern, da diese Form der Darstellung eher der alltäglichen Wahrnehmung des Menschen entspricht.“ (Merten 2002, S.1) Aufgrund dessen ist in der 3D-Darstellung eine geringere Abstraktionsanforderung an den Benutzer gestellt. Die hohe Interaktionskomplexität ist als gravierender Nachteil zu werten. Dieser Nachteil wird in unserem Beispiel durch die verringerte Komplexität der Interaktion in der 2D-Darstellung ausgeglichen. Der Nachteil der hohen Abstraktion vom reellen Raum in die 2D-Darstellung wird durch die reellere Visualisierung im Dreidimensionalen ausgeglichen.

Es wurde ein System entwickelt, das ohne teure Hardware (3D-Metaphern) und komplizierte Constraints-Implementierung an jedem Desktop-Arbeitsplatz verfügbar ist.

Dabei ist zu beachten, dass dies nicht für alle fachlichen Fragestellungen und Szenarien geeignet ist. Sobald für Geodaten der Höhenwert als zusätzliche Variable betrachtet werden muss, ist der beschriebene Ansatz nicht anwendbar.

## 8 LITERATUR

- Cooper, J. W. (1998): The Design Pattern Java Companion, Addison-Wesley Publishing Company.
- Eilebrecht, K. & G. Starke (2004): Patterns kompakt - Entwurfsmuster für effektive Software-Entwicklung, Spektrum Akademischer Verlag, Heidelberg und Berlin.
- Gamma, E., R. Helm, R. Johnson & J. Vlissides (1995): Design Patterns (elements of reusable object-oriented Software), Addison-Wesley Publishing Company.
- Haber, R. B. & D. A. McNabb (1990): Visualization Idioms: A Conceptual Model for Scientific Visualization Systems. In B. Schriver, G. M. Nielson & L. J. Rosenblum: Visualization in Scientific Computing, IEEE Computer Society Press, S. 74 - 93.
- Krutscher, A. (1997): Modell-Ansicht-Bediener (Model-View-Controller), Fachhochschule Wedel.  
(zuletzt abgerufen am 30.11.04: <http://www.fh-wedel.de/~si/seminare/ws97/Ausarbeitung/3.Krutscher/archmu3.htm>)
- Lehmkuhler, S. (1999): Computergestützte Visualisierungstechniken in der Stadtplanung. Bedingungen und Ersatze des Potentials traditioneller durch computergestützte Visualisierungstechniken in der Stadtplanungspraxis. In: Dortmunder Beiträge zur Raumplanung, Bd. 91, Dortmund.
- May, M., K. Rinke, I. Abramovic, J. Brauner & T. Heinen (2004): Systemspezifikation - Funktionen, Visualisierung, Schnittstellen und Editierung in FLUMAGIS. In: J. Möltgen & D. Petry, Hrsg.: Interdisziplinäre Methoden des Flusseinzugsgebietsmanagements, Workshopbeiträge, 15./16. März 2004, IfGIprints, Bd. 21, Münster: Institut für Geoinformatik / Solingen: Verlag Natur & Wissenschaft, S. 249 - 260.
- Merten, S. (2002): Untersuchung der Nutzbarkeit von constraintbasierten 3D-Interaktionen zur Gestaltung von Planungsentwürfen, Diplomarbeit am Institut für Geoinformatik, Universität Münster.
- Möltgen, J. & U. Streit (2004): FLUMAGIS - Interdisziplinäre Methoden- und Werkzeugentwicklung zur Planung und Kontrolle von Maßnahmen für das Flusseinzugsgebietsmanagement mit GIS. In: J. Möltgen & D. Petry, Hrsg.: Interdisziplinäre Methoden des Flusseinzugsgebietsmanagements, Workshopbeiträge, 15./16. März 2004, IfGIprints, Bd. 21, Münster: Institut für Geoinformatik / Solingen: Verlag Natur & Wissenschaft, S. 11 - 20.
- Oestreich, B. (2001): Objektorientierte Softwareentwicklung: Analyse und Design mit der Unified modeling language, 5. Auflage, Oldenbourg Verlag, München.
- Schmidt, B. (2002): Verknüpfung der Datenmodelle für GIS und interaktive 3D-Visualisierung, IfGIprints, Bd. 17, Münster: Institut für Geoinformatik / Solingen: Verlag Natur & Wissenschaft.
- Schmidt, B., M. May & T. Heinen (2004): Triturus - Ein Java-Framework für Applikationen zur 3D-Geovisualisierung. In: May, M., Hrsg.: 3D Geovisualisierung - Ein Überblick, "geoinformatik online", Ausgabe 01/2004, Institut für Geoinformatik, Universität Münster.  
(zuletzt abgerufen am 30.11.04: <http://gio.uni-muenster.de>)
- Seeman, J. & J. W. v. Gudenberg (2000): Software-Entwurf mit UML, Springer Verlag, Berlin und Heidelberg.