

# ViComp – Architektur und Städteplanung mit virtueller Modellierung und Komposition

Andreas Schlempp und Ulrich Köthe \*  
FhG IGD/Rostock

## Zusammenfassung

Das System **ViComp** (Virtual Composer) [Sch96] dient zur interaktiven Komposition von virtuellen Szenen. **ViComp** wird innerhalb des vom Wirtschaftsministerium Mecklenburg-Vorpommerns geförderten Verbundprojekts „Erweitertes Architektur- und Planungsmodell, EAPM“ entwickelt. Anlaß hierzu waren die bei der Erstellung komplexer Architektur- und Stadtmodelle auftretenden Inkompabilitäten durch die Zusammenführung von aus verschiedenen Systemen stammenden Einzelteilen. Neben der sehr zeit- und damit auch kostenintensiven Modellierung der verschiedenen Objekte werden hier unter anderem Bild-basierte Rekonstruktionsverfahren, Objektdatenbanken oder auch 3D-CAD Daten eingesetzt. Die Inkompabilitäten beziehen sich vor allem auf den Datenbestand und finden sich typischerweise in unterschiedlichen Dateiformaten, Maßstäben oder Orientierungen der Koordinatensysteme. Neben der Überwindung dieser Inkompabilitäten wird mit **ViComp** eine Möglichkeit geschaffen, die verschiedenen Komponenten in einer Gesamtszene anzuordnen. Durch direktmanipulative und Constraint-basierte Interaktion mit den Standardeingabegeräten (2D-Maus, Tastatur) innerhalb einer einzigen 3D-Ansicht wird eine intuitive, schnelle und flexible Lösung dieser Aufgabe vorgestellt.

**Keywords:** Architectural Design, Scene Composition, Virtual Worlds.

## 1 Einleitung

Der Anstieg der allgemein verfügbaren Rechenleistung im Bereich von 3D-Visualisierungen führt auch in den Bereichen der Architektur- und der Städteplanung zu vermehrtem Einsatz von virtuellen Modellen. Die Vorteile und Möglichkeiten dieser Modelle (erhöhte Flexibilität in Materialauswahl und Beleuchtungsvarianten, realitätsnaher Eindruck durch virtuelles Begehen, ...) gegenüber den herkömmlichen Methoden wie Skizzen oder „harte“ Modelle aus Holz, Gips, etc. liegen klar auf der Hand und stellen wertvolle Hilfen sowohl im Planungsprozeß als auch in der anschließenden Öffentlichkeitsarbeit für die jeweiligen Projekte dar.

Ein Problem bei dieser Art der Projektvisualisierung besteht darin, daß die für die Baudurchführung erstellten CAD-Daten meist für eine 3D-Visualisierung nicht ausreichend sind (nur 2D oder  $2\frac{1}{2}$ D, keine Materialien, keine Beleuchtungsinformationen,...). Es ist also

---

\*Fraunhofer Gesellschaft - Institut Graphische Datenverarbeitung, Joachim-Jungius-Straße 9, 18059 Rostock, Germany. e-mail: schlempp@egd.fhg.de, ulli@egd.fhg.de

eine Nachmodellierung oder Aufbereitung der Grunddaten erforderlich, um eine virtuelle Ansicht eines Architekturentwurfs berechnen zu können. Dieser Aufwand erhöht sich entsprechend, falls das Umfeld eines Objekts mit in die Darstellung einbezogen werden soll (Straßenflucht, Stadtteil).

Sind die Genauigkeitsanforderungen nicht zu hoch (VR-Applikationen setzen auf visuelle Korrektheit, genaue Maße sind nicht unbedingt erforderlich), werden auch Bild-basierte Rekonstruktionsverfahren (z.B. IGD-Projekte REKO ([Luth95]), SMART oder ERSO) eingesetzt, um ohne zeitaufwendige Interaktion die erforderlichen Einzelmodelle zu gewinnen. Bei der Zusammenführung der aus eventuell unterschiedlichen Systemen stammenden Einzelobjekte (Rekonstruktionen, Nachmodellierungen, geeignete CAD-Strukturen, archivierte Daten) entstehen allerdings Probleme durch Inkompabilitäten. Das betrifft z.B. verschiedene Dateiformate und Maßstäbe oder unterschiedliche Orientierungen der Koordinatensysteme. Neben den rein technischen Schwierigkeiten wird dadurch auch die Platzierung der verschiedenen Objekte mit CAD-typischen Methoden erschwert, da unter Umständen keine spezifischen Einzelpositionen eines Modells (z. B. Hausecke) identifiziert werden können.

Im Rahmen des vom Wirtschaftsministerium Mecklenburg-Vorpommerns geförderten Verbundprojekts „Erweitertes Architektur- und Planungsmodell, EAPM“ wird das System **ViComp** (Virtual Composer) zur Überbrückung der oben genannten Kompatibilitätsprobleme entwickelt. Neben der Integration verschiedener Modelle unterschiedlicher Herkunft wird mit **ViComp** eine Möglichkeit geschaffen, diese schnell und einfach zu einer virtuellen Gesamtszene zusammenzustellen.

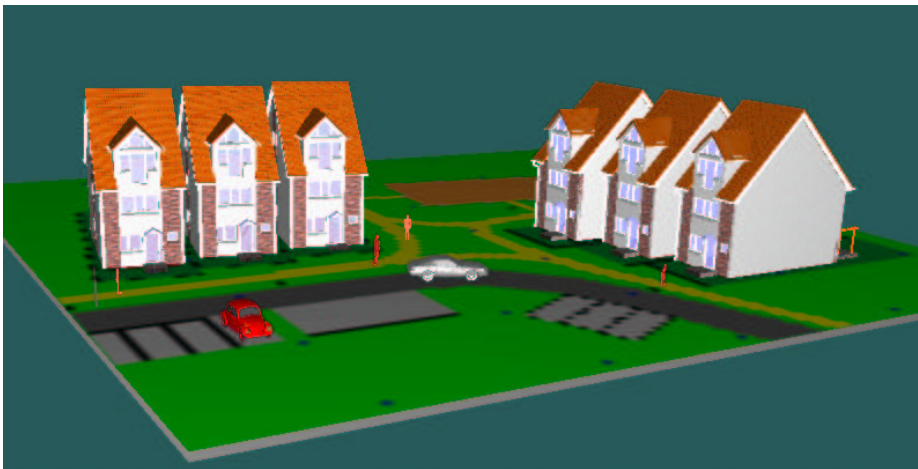


Abbildung 1: Mit **ViComp** erstellte virtuelle Szene

## 2 Objektintegration

**ViComp** setzt auf den im World-Wide-Web (WWW) eingesetzten Quasi-Standard zur Beschreibung virtueller Welten, VRML (Virtual Reality Modeling Language)[ANM96] auf. Dieses Format ist inzwischen sehr weit verbreitet und wird von vielen Modellierungs- und CAD-Systemen unterstützt. Auch die oben angeführten Rekonstruktionssysteme sind in der Lage, VRML zu exportieren. Ein weiterer Vorteil dieses Formats besteht darin, daß über das WWW eine Unzahl von kleinen Hilfsmodellen (Menschen, Pflanzen, ...) verfügbar ist, welche der Realitätssteigerung einer virtuellen Szene sehr dienlich sind.

Ein vorgeschalteter Fittingprozeß überbrückt bei der Aufnahme eines neuen Objekts in die aktuelle Szene die Inkompatibilitäten bezüglich der Orientierung des Objekt-Koordinatensystems und des Maßstabs.

In der ersten Stufe des Fittingprozesses redefiniert der Benutzer interaktiv die Orientierung und den Ursprung des lokalen oder auch Objekt-Koordinatensystems des neuen Objekts. Die dazu nötige Transformation wird direktmanipulativ durch Rotation oder Verschiebung des Objekts innerhalb eines eingeblendeten Koordinatensystems festgelegt. Alternativ dazu stehen einige Dialogelemente zur Verfügung, um die gängigen Veränderungen (Tauschen von Koordinatenachsen, Anpassung des Ursprungs an die Bounding-Box, Änderung von Links- auf Rechtssystem) durch einen einfachen Mausklick zu bewirken (Abb. 2). Selbst wenn das Objekt in einem konformen Koordinatensystem definiert ist, wird durch die individuelle Festlegung des Koordinatensystem-Ursprungs ein Bezugspunkt für die spätere Interaktion mit dem Objekt bestimmt und diese dadurch wesentlich vereinfacht.

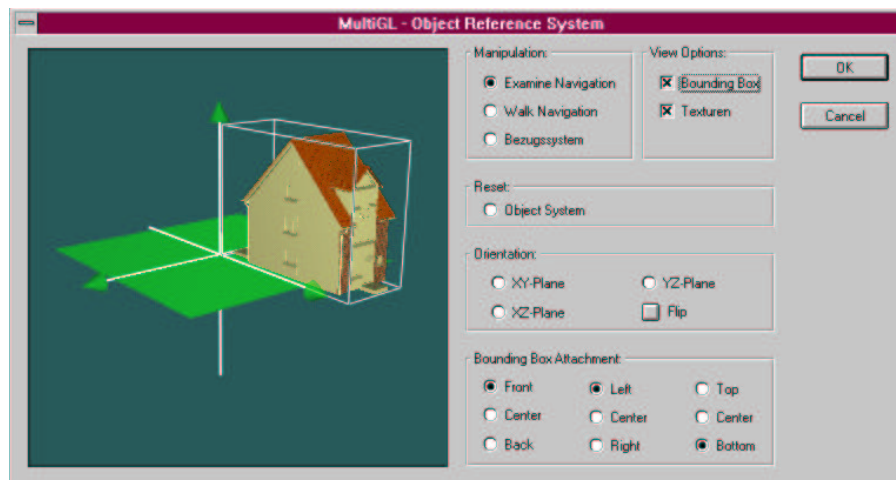


Abbildung 2: Direktmanipulative Festlegung des lokalen Koordinatensystems eines Objekts

Die zweite Stufe des Fittingprozesses fügt das neue Objekt automatisch in die aktuelle Szene ein. Das dazugehörige Welt-Koordinatensystem der Szene ist in **ViComp** auf für Visualisierungssysteme typische Art definiert und bildet ein Rechtssystem mit nach oben zeigender Y-Achse (Vertikale).

Der Faktor  $S$  für die entsprechende Skalierung zur Maßstabsanpassung des aufzunehmenden Objekts berechnet sich durch den interaktiv änderbaren Bruchteil  $r$  des Größenverhältnisses zwischen dem Geländemodell  $G$  und dem neuen Objekt  $O$ :

$$S = \frac{1}{r} * \frac{MAX(G.bbox().width(), G.bbox().height(), G.bbox().depth())}{MAX(O.bbox().width(), O.bbox().height(), O.bbox().depth())}$$

Auf diese Weise wird das Objekt dem Benutzer in einer handhabbaren Anfangsgröße präsentiert, welche dann interaktiv an die Verhältnisse in der Szene angepaßt werden kann. Die bei der Aufnahme des Objekts in die Szene implizit gegebene Übereinstimmung der Orientierung des eventuell neu festgelegten Objektsystems mit dem des Welt-Koordinatensystems gewährleistet eine gemeinsame Vertikale aller Objekte der Szene.

Eine abschließende Translation des Objektsprungs auf eine Anfangsposition innerhalb der Szene schließt den Fittingprozeß ab. Die Berechnung des dazu nötigen Translationsvektors wird im nachfolgenden Kapitel erläutert.

### 3 Objektplatzierung

Um virtuelle Modelle in einer 3-dimensionalen Szene anzuordnen, müssen deren Positionen im 3D bestimmt werden. Die Vorgaben bei der Entwicklung von **ViComp** sind das Arbeiten mit den Standardeingabegeräten (2D-Maus, Tastatur) innerhalb einer einzigen 3D-Ansicht. Es müssen also geeignete Constraints eingeführt werden, um eine gezielte 3D-Positionseingabe unter diesen Voraussetzungen zu ermöglichen.

Das von **ViComp** anvisierte Anwendungsfeld besteht in der Komposition von Architektur- und Städtmodellen. Es kann also davon ausgegangen werden, daß für jede Szene ein entsprechendes Geländemodell in Form einer geometrischen Beschreibung (VRML-Modell) vorliegt. Sollte dies nicht der Fall sein, so wird von **ViComp** eine quadratische Fläche als Standardgelände generiert.

Dieses Geländemodell wird als erstes Modell geladen und kann im Verlauf der Komposition durch eine neue oder selektierte Geometrie ersetzt werden. Neben dem virtuellen Gelände an sich wird durch dieses Modell das Constraint-System für die weitere Interaktion initialisiert. Ziel des Constraint-Systems ist die Reduzierung der Problematik der 3D-Positionsbestimmung auf eine einfache 2D-Eingabe mit automatischer Bestimmung der fehlenden Dimension.

In **ViComp** erfolgt die Positionierung der Objekte entweder absolut durch Eingabe der gewünschten Zielposition oder durch eine relative Verschiebung.

Die Zielposition für die absolute Positionierung kann entweder, falls die entsprechenden Weltkoordinaten bekannt sind, über die Tastatur eingegeben oder mit der Maus direkt auf dem Geländemodell markiert werden.

Bei der Eingabe mit der Maus werden die Koordinaten für die Zielposition wie bei Ray-Casting Verfahren durch den Schnitt eines durch die Mauszeigerposition gehenden Strahls mit dem Geländemodell berechnet. Dazu wird die Position des Mauszeigers innerhalb des Arbeitsfensters auf die Bildebene des Kamera-Koordinatensystems projiziert und mit dem Ursprung des Kamera-Koordinatensystems in das Welt-Koordinatensystem transformiert.

Diese beiden Positionen bilden den Durchgangs- und Startpunkt des nun 3-dimensionalen geometrischen Schnittstrahls.

Der Schnitt dieses Strahls mit dem Geländemodell liefert die Zielposition, die wiederum aus dem Objekt-Koordinatensystem des getroffenen Geländeprimitivs in das Welt-Koordinatensystem transformiert wird. Eine abschließende Transformation dieser Zielposition in das lokale Koordinatensystem des zu positionierenden Objekts liefert schließlich den Translationsvektor für die entsprechende Verschiebung.

In gleicher Weise wird bei der Aufnahme eines neuen Objekts in die Szene dessen Anfangsposition vom Benutzer auf dem Gelände markiert und die den Fittingprozeß abschließende Translation bestimmt.

Die relative Verschiebung eines Objekts wird durch das Ziehen mit der Maus (drag and drop) vorgegeben, einer Interaktion, die eine Reihe diskreter Mauszeigerpositionen innerhalb des Arbeitsfensters erzeugt. Aus der Differenz zweier aufeinanderfolgender Positionen wird ein Bewegungsvektor innerhalb der Bildebene generiert (der Faktor  $F$  steuert die Geschwindigkeit der Translation):

$$Move_{Bildebene} = \begin{pmatrix} mouse_n.x() - mouse_{n-1}.x() \\ mouse_n.y() - mouse_{n-1}.y() \\ 0 \end{pmatrix} * F$$

Dieser wird in das Welt-Koordinatensystem transformiert und auf den ebenfalls in das Welt-Koordinatensystem transformierten Ursprung des zu verschiebenden Objekts<sup>1</sup> aufaddiert. Die so entstandene Position definiert den Ausgangspunkt des Strahls, der außerdem parallel zur Y-Achse des Welt-Koordinatensystems verläuft (senkrechtes Lot) und zum Schnitt mit dem Geländemodell gebracht wird. Analog zur absoluten Positionierung mit der Maus wird aus der entstandenen Schnittposition der Verschiebungsvektor für das Objekt berechnet.

Neben der Positionierung eines Objekts muß dieses auch orientiert und skaliert werden. Da aber die Vertikale und die Seitenverhältnisse eines Objekts innerhalb von **ViComp** erhalten bleiben, reduzieren sich diese Interaktionen auf die Eingabe je eines 1-dimensionalen Parameters (Drehwinkel um die Vertikale bzw. uniformer Skalierungsfaktor) und können problemlos durch Ziehen des Mauszeigers eingegeben werden.

Nach der Frage, wie die Objekte in der Szene positioniert werden, steht das Problem, wohin sie positioniert werden müssen, um gegebene Verhältnisse wiederzuspiegeln.

In **ViComp** werden zu diesem Zweck die Lage- und Grundrißinformationen in Form einer 2-dimensionalen Textur auf das Geländemodell appliziert und die Objektpositionen damit graphisch eingetragen. Durch die orthogonale Projektion der Textur auf das 3-dimensionale Geländemodell in **ViComp** werden die fehlenden Höhen der Objektpositionen automatisch festgelegt.

Wird diese Textur nicht mit dem Geländemodell bereitgestellt, so können die erforderlichen Daten leicht aus Unterlagen wie Stadt- und Bebauungsplänen oder Katasterkarten extrahiert werden. In einem Preprozeß werden nach interaktiver Definition zweier Referenzpunkte die restlichen Texturkoordinaten automatisch berechnet.

---

<sup>1</sup>Die Bestimmung der Transformation von Objekt- in Weltkoordinaten ist für beide Positionierungsverfahren notwendig und wird im Abschnitt *Datenmodell* erläutert.

Die Textur dient dem Benutzer bei der Verschiebung der Objekte auf dem Geländemodell als Vorlage und ermöglicht die für Visualisierungszwecke ausreichend genaue Positionierung und Skalierung der Objekte.

Soll beispielsweise ein Gebäudemodell auf einem Bebauungsplan positioniert werden, so wird im Fittingprozeß eine der unteren Hausecken als Bezugspunkt definiert und in der Szene durch einen Mausklick an seine Stelle auf der Grundrißtextur transliert. Eine anschließende Rotation bringt eine der Hausseiten mit der entsprechenden Grundrißlinie zur Deckung und die Skalierung mit feststehendem und korrekt positioniertem Bezugspunkt endet bei völliger Überdeckung des virtuellen Modells mit seinem Grundriß.

## 4 Datenmodell

Der Entwurf des internen Datenmodells ist von entscheidender Bedeutung für die erreichbare Funktionalität und Flexibilität eines Programmsystems. In diesem Kapitel wird in groben Zügen der Entwurf und das endgültige Design des Datenmodells von **ViComp** beschrieben, welches mit Hilfe der Object-Modeling-Technique (OMT) nach Rumbaugh, et. al. ([RBP<sup>+</sup>91]) entwickelt wurde.

Zu Beginn eines jeden Entwurfs steht eine Reihe von zu erfüllenden Anforderungen, welche im Verlauf der weiteren Entwicklung präzisiert, korrigiert und auch ergänzt werden. Die informellen Anforderungen an das Datenmodell von **ViComp** sind:

- Beschreibung einer virtuellen graphischen Szene
- Unterstützung semantischer Strukturen
- Bestimmung eines Schnittpunktes mit einem geometrischen Strahl
- Wiederverwendbarkeit von Strukturen (Prototyping)
- Variantenmodellierung (Abb. 3)
- Neudefinition des lokalen Koordinatensystems von graphischen Objekten
- Selektion und Verschiebung von graphischen Objekten
- Verlust- und redundanzfreies Konvertieren von und nach VRML
- Schnelles Traversieren für das Rendering (OpenGL) und Picking

Die ersten beiden Anforderungen führen direkt zu einer Menge von erforderlichen Objekten, um einen sogenannten Geometrie- oder Szenengraphen aufzubauen. Diese Menge kann in folgende drei Klassen aufgeteilt werden:

- Graphische Primitive (Punkt, Linie, Fläche, ...)
- Eigenschaften der graphischen Primitive (Material, Textur, Transformation)
- Strukturierungsklassen (Gruppierung)

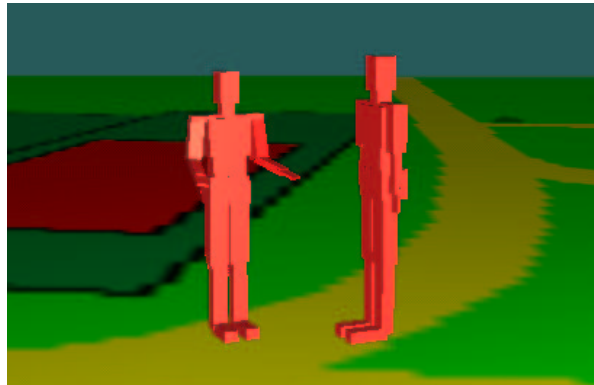


Abbildung 3: Variantenmodellierung zur Erhöhung des Realitätseindrucks

Für die Primitive und Gruppierungselemente wird die gemeinsame Basisklasse Graphikelement (part) eingeführt, um eine gemeinsame Schnittstelle für allgemeingültige Operationen zu ermöglichen („Composite“ Design Pattern [GHJ+95]).

Bei der Definition der verschiedenen Primitive muß gewährleistet sein, daß die Objekte für das oben beschriebene Positionierungsverfahren die mathematische Schnittpunktberechnung mit einem 3-dimensionalen geometrischen Strahl unterstützen. Die Gruppierungselemente (assembly parts) delegieren diese Aufgabe nach erfolgreichem Bounding Box Test an deren Nachfahren im Szenengraphen weiter. Transformationen innerhalb des Szenengraphen transformieren den Schnittstrahl in das lokale Koordinatensystem der jeweils von ihnen betroffenen Graphikelemente.

Die aufgelisteten Eigenschaften sind nur bedingt Attribute der sie betreffenden Graphikelemente. So ist die Lage eines Graphikelements (Transformation) vielmehr Merkmal der Beziehung zwischen dem Graphikelement und seiner Umgebung bzw. seiner Vorgänger im Szenengraphen. Im Datenmodell wird diese Beziehung ausgedrückt, indem eine Transformation als eigenständiges Objekt des Szenengraphen definiert wird und das Bindeglied zwischen einem Graphikelement und dessen Vorgänger darstellt.

Dieses Konzept hat zur Folge, daß jedem Graphikelement eine Transformation übergeordnet ist, die neben der initialen Positionierung durch den Fittingprozeß auch die spätere interaktive Verschiebung akkumulieren kann.

Im Gegensatz zu der Lagebeschreibung sind die Oberflächeneigenschaften (Material, Textur) direkte Attribute der darstellbaren Primitive und haben für Gruppierungselemente keine Bedeutung. Um aber das Rendering mit OpenGL optimal zu unterstützen werden auch diese Attribute aus den Primitive herausgelöst und ebenfalls als eigenständige Objekte definiert. Ein solches Material- oder Texturobjekt wird in der sequentiellen Anordnung der Kinder einer Gruppierung vor die von ihm betroffenen Primitive eingefügt und behält Gültigkeit bis zum nächsten traversierten Material- bzw. Texturobjekt (Abb. 4). Eine aufeinanderfolgende und mit OpenGL „teure“ Definition eines identischen Materials kann so umgangen und damit das Rendering beschleunigt werden. Die damit festgelegte Traversierungsreihenfolge

der Kinder einer Gruppierung wird als kleinerer Nachteil in Kauf genommen.

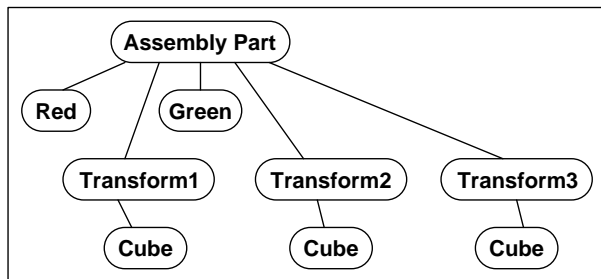


Abbildung 4: Ein roter und zwei grüne Würfel

Da bislang sowohl die Transformations- als auch die Material- und Texturobjekte Kinder einer Gruppierung sein können, ist die Definition der gemeinsamen Basisklasse Modifizierer erforderlich, um aus Sicht der Gruppierung eine einheitliche Schnittstelle zu dessen Kindern präsentieren zu können.

Die Beschreibung von Lichtquellen im **ViComp**-Datenmodell ist ebenfalls durch deren Behandlung in VRML und OpenGL beeinflusst. So wird eine Lichtquelle wie ein Material- oder Texturobjekt als Modifizierer des Zustands des Renderers definiert und beleuchtet somit lediglich die im Szenengraphen nachfolgenden Primitive.

Die beiden Forderungen nach Wiederverwendbarkeit und Variantenmodellierung umschreiben die zwei Möglichkeiten, graphische Objekte oder ganze Untergraphen zu kopieren (Abb. 5).

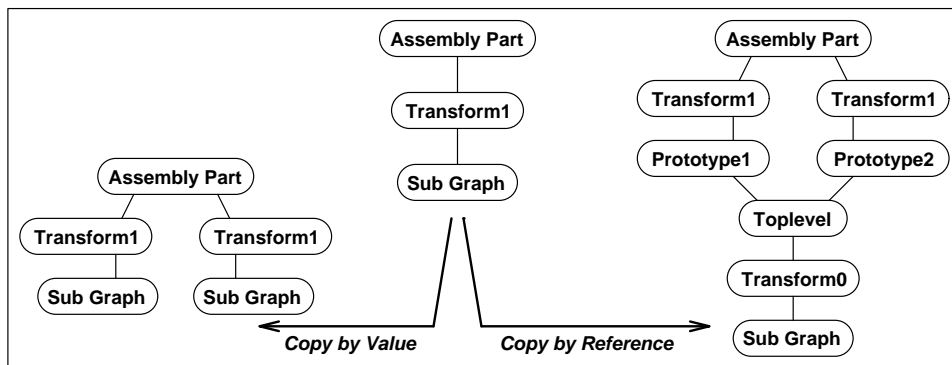


Abbildung 5: Die beiden Möglichkeiten, Subgraphen zu kopieren

Soll ein Objekt mehrmals in identischer Form (i. allg. durch eine weitere Transformation räumlich versetzt) in einer Szene vorkommen, so genügt es, zusätzliche Referenzen auf die ursprünglichen Daten zu erzeugen (Copy by Reference). Hierzu ist der neue Objekttyp Prototyp notwendig, um anzuzeigen, daß der gespeicherte Subgraph mehrfach benutzt wird. Ein Prototyp-Element vertritt aus der Sicht seiner Vorfahren im Szenengraphen eine



homogene Geometrie und wird daher innerhalb der Klassenhierarchie des Datenmodells auf eine Ebene mit den Primitiven gestellt. Da umgekehrt gewährleistet sein soll, daß jedes Graphikelement einen eindeutigen Vorfahren im Szenengraphen hat, muß ein zusätzliches Wurzelobjekt (toplevel) zur Aufnahme der Mehrfachreferenzierung definiert werden. Die Wurzelobjekte haben semantisch keine Vorfahren, verwalten aber die auf sie verweisenden Prototypen (deren Anzahl im Falle eines globalen Wurzelobjekts auch Null sein kann). Um Redundanzen im Subgraphen zu vermeiden, wird nach dem Löschen des vorletzten Prototypen eines Wurzelobjekts dieses aufgelöst und der entsprechende Subgraph direkt in den Szenengraphen aufgenommen (Abb. 6).

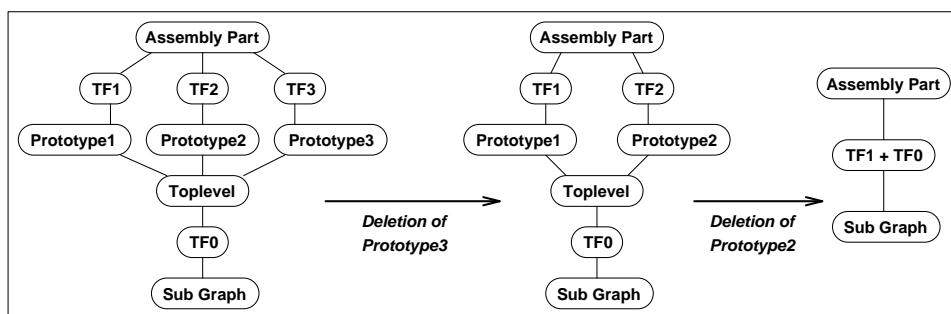


Abbildung 6: Die Auswirkung von sukzessiver Löschung von Prototypen auf ein und denselben Subgraphen

Um eine Kopie zu erzeugen, die unabhängig vom Original bearbeitet werden kann, müssen die Daten komplett geklont werden (Copy by value). Das „Verwandschafts“-Verhältnis zwischen Original und Kopie wird in **ViComp** nicht gespeichert. Somit werden für diese Operation keine zusätzlichen Klassen benötigt.

Die Erweiterung der Gruppe der Graphikelemente um ein zusätzliches Koordinatensystem isoliert dieses Koordinatensystem definierende Transformation und stellt sicher, daß diese immer als erste Transformation auf die dazugehörige Geometrie angewandt wird.

Die Gesamtszene enthält schließlich die globalen Wurzelobjekte des Geländemodells und aller geladenen Einzelteile. Weitere Informationen zur vollständigen Beschreibung einer virtuellen Szenerie, wie etwa das Kameramodell, sind ebenfalls hier angesiedelt.

Abbildung 7 stellt die erhaltene Klassenhierarchie in der OMT-Notation dar.

Die weiteren Anforderungen nach Datenaustausch, effizientem Rendering und Selektion von Objekten sind im bisherigen Entwurf des **ViComp**-Datenmodells ebenfalls gelöst.

Die Konvertierung von VRML in einen internen Szenengraphen und umgekehrt wird durch das bisherige Klassenkonzept ermöglicht, da mit VRML ebenfalls ein hierarchischer Szenengraph beschrieben wird, dessen Elemente (Nodes) sich in die drei Klassen Shape, Property und Group aufteilen. Der Internetlink des Gruppierungselements Anchor wird allerdings ignoriert und ein Inline-Element nur geladen, wenn sich das referenzierte Modell auf dem lokalen Dateisystem befindet.

Mit dieser Ähnlichkeit des **ViComp**-Datenmodells zu VRML ist auch die Unterstützung des

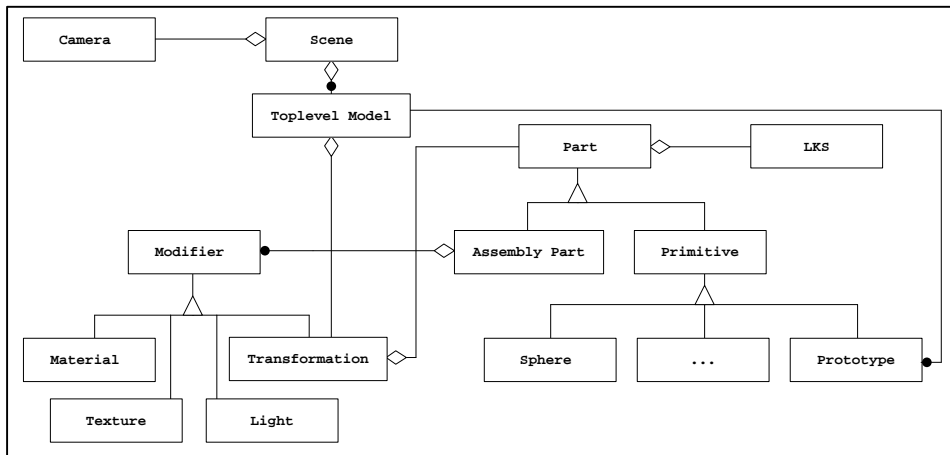


Abbildung 7: Klassendiagramm des **ViComp** Datenmodells in der OMT-Notation

Renderings mit OpenGL gewährleistet<sup>2</sup>. Beim Traversieren der Datenstruktur lassen sich ohne weitere Konvertierung direkt OpenGL-Aufrufe ableiten. Ein kleiner Effizienzgewinn wurde durch die Sonderstellung der Transformationen erreicht. Durch die Akkumulation mehrerer Transformationen beim Aufbau der Datenstruktur und späterer Interaktion werden diese Matrixoperationen beim Rendering nicht mehr durchgeführt.

Die Selektion (Picking) eines Objekts kann technisch auf zwei unterschiedliche Weisen realisiert werden. Zum einen wird abhängig von der Mauszeigerposition und der aktuellen Viewporttransformation ein 3-dimensionaler Strahl erzeugt. Ausgehend von der Wurzel des Geometriebaumes testen alle Objekte rekursiv, ob sie von dem Strahl geschnitten werden, wobei traversierte Transformationen diesen Strahl eventuell verändern. Die zweite Variante erzeugt einen sogenannten Hit, wenn ein graphisches Primitiv an der Mauszeigerposition in den Bildspeicher gezeichnet wird (OpenGL, [NDW93]). Neben dem „getroffenen“ Objekt ist der aktuelle Pfad zu diesem Primitiv Bestandteil des Hits.

Das **ViComp**-Datenmodell unterstützt beide Verfahren und liefert als Ergebnis einer Selektion einen Geometrieknoten vom Typ Part. Sollte das getroffene Objekt einen Vorfahren der Ausprägung Prototyp haben, so wird dieser Prototyp anstelle des Primitivs selektiert, um eine eindeutige Erweiterung der Selektion innerhalb der Szenenhierarchie (Aufstieg im Graphen) zu ermöglichen (linke Hand → linker Arm → beide Arme ...)[FDF<sup>+</sup>90]. Um die Selektion von einem Prototypen ausgehend weiter zu verfeinern (linker Daumen), muß der Prototyp zuerst durch eine Kopie (Klon) der von ihm referenzierten Geometrie ersetzt werden (andernfalls würden beide Daumen gleichzeitig selektiert sein), um eine eindeutige Selektion zu ermöglichen.

<sup>2</sup>Die Vorlagen für VRML und OpenGL, das Open Inventor File Format (iv) und die Graphics Library (GL), sind beide von Silicon Graphics entwickelt worden und direkt art-verwandt.

## 5 Bewertung und Ausblick

Erste Demonstrationen haben gezeigt, daß nach kurzer Einarbeitungszeit in die nötigen Interaktionsabläufe ein intuitives und zügiges Arbeiten mit dem **ViComp**-System möglich ist. Es ist allerdings offensichtlich, daß die Benutzbarkeit des Systems bei der direktmanipulativen Arbeitsweise wesentlich von der erreichbaren Bildwiederholrate des Renderings abhängt. Da außerdem die Einsatzfähigkeit auch auf Rechnern der unteren Leistungsskala (z. B. Windows NT-PC) gewährleistet sein soll, zielen die zukünftigen Arbeiten an **ViComp** auf die Optimierung und Beschleunigung des Arbeitsablaufs.

Dabei werden zwei unterschiedliche Strategien verfolgt. Auf der einen Seite steht die Reduzierung des Aufwands an direktmanipulativer Interaktion (drag and drop) und der damit verbundenen wiederholten Darstellung der Zwischenergebnisse (Visual Feedback) im Vordergrund. Leistungsfähige Funktionen (point and click) können dabei den Arbeitsaufwand erheblich verringern.

Ein Ansatzpunkt hierfür wäre der Einsatz von Methoden der digitalen Bildanalyse. Bildanalysealgorithmen könnten beispielsweise die Textur der Grundebene auswerten, um den Grundriß eines Gebäudes automatisch zu finden. Durch ein optimales Matching zwischen diesem Grundriß und der Grundfläche eines Modells wird das Modell automatisch in der Szene platziert. Die für diese Arbeitsweise notwendige Interaktion beschränkt sich darauf, Objektmodelle den richtigen Grundrissen zuzuordnen.

Alternativ zur Verringerung des Arbeitsaufwands ist die Optimierung des Renderingvorgangs wesentlich. Ein entscheidender Faktor für die benötigte Renderingzeit ist die texturierte Darstellung der Szene. Zur Zeit wird in **ViComp** im Interaktionsmodus die Texturierung aller Objekte bis auf das Gelände ausgeschaltet. Aber selbst die texturierte Darstellung des Geländemodells ist unter Umständen (großes Arbeitsfenster, hochauflösende Textur) zu langsam, um ein Echo der Interaktion nach akzeptabel kurzer Zeit zu erhalten.

Zwei Ansätze werden verfolgt, um die Darstellung des Geländes zu beschleunigen:

**Vektortextur** In einem 2D-Graphikeditor werden die interessierenden Positionen und Grundrisse der zu erstellenden Szene in Form einer Vektorgraphik definiert. Als Vorlage dient dabei die in den Hintergrund des Zeichenfensters gelegte Raster-Geländetextur.

Die so erzeugten Grundrißdaten sind in Texturkoordinaten definiert. Unter Ausnutzung der Texturkoordinaten und der in das Welt-Koordinatensystem transformierten Positionen zweier Punkte im Geländemodell kann die Transformation dieser Grundrißinformationen in die XZ-Ebene des Welt-Koordinatensystems berechnet werden. Durch orthogonale Projektion entlang der Y-Achse des Welt-Koordinatensystems wird die Vektorgraphik dann auf das Geländemodell appliziert.

Die Einsparung an Renderingzeit durch den Verzicht auf die Texturierung des Geländes dürfte wesentlich größer sein, als der zusätzliche Aufwand der Darstellung dieser neuen Grundrißgeometrien (Vektortextur).

**Texturausschnitt** Dieses Konzept versucht den Aufwand der Geländetexturierung direkt zu vermindern, indem der zu texturierende Bereich des Geländes verkleinert wird. Der Benutzer markiert dazu mit der Maus in einem separaten 2D-Ansichtsfenster der

Textur eine sogenannte ROI (region of interest). Das Geländemodell wird dann dementsprechend automatisch in zwei disjunkte Teile zerlegt und nur der im allgemeinen wesentlich kleinere, der ROI entsprechende Teil, wird bei der Darstellung texturiert. Um die ROI möglichst genau in der 3D-Szene darzustellen, muß das Gelände eventuell neu trianguliert werden.

## Literatur

- [ANM96] A. L. Ames, D. R. Nadeau, J. L. Moreland. *The VRML Sourcebook*. John Wiley & Sons, Inc., New York, 1996.
- [FDF<sup>+</sup>90] J. D. Foley, A. v. Dam, S. K. Feiner and J. F. Hughes. *Computer Graphics - Principles and Practice*. Addison-Wesley, Reading, Massachusetts, 1990.
- [GHJ<sup>+</sup>95] E. Gamma, R. Helm, R. Johnson and J. Vlissides. *Design Patterns - Elements of Reusable Object-Oriented Software*. Addison-Wesley, Reading, Massachusetts, 1995.
- [Luth95] Dr. Wolfgang Luth, Katja Otto. *REKO: Interaktive bildgestützte 3-D Objektkonstruktion*. In: Computer Graphics Topics 6/95, S. 31-32, HdGDV, Wilhelminenstraße 7, D-64283 Darmstadt
- [NDW93] J. Neider, T. Davis and M. Woo. *OpenGL Programming Guide*. Addison-Wesley, Reading, Massachusetts, 1993.
- [RBP<sup>+</sup>91] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy and W. Lorensen. *Object-Oriented Modeling and Design*. Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1991.
- [Sch96] A. Schlempp. *ViComp: Urban Design using Virtual Modelling and Composition*. In: Computer Graphics Topics 6/96, S. 18-19, HdGDV, Wilhelminenstraße 7, D-64283 Darmstadt