

# **Eine deklarative Komponentenarchitektur und Interaktionsbausteine für dreidimensionale multimediale Anwendungen**

## **Kurzfassung der Dissertation**

zur Erlangung des akademischen Grades  
Doktoringenieur (Dr.-Ing.)

vorgelegt an der  
Technischen Universität Dresden  
Fakultät Informatik

eingereicht von  
**Dipl.-Inform. Raimund Dachsel**  
geboren am 8. Februar 1971 in Elbingerode

Betreuender Hochschullehrer: Prof. Dr.-Ing. Klaus Meißner

Dresden im März 2004

# Inhaltsübersicht der Dissertation

|  |   |            |
|--|---|------------|
| <b>1</b>   | <b>Einleitung .....</b>   | <b>1</b>   |
| 1.1  | Hintergrund und Motivation .....                                  | 1          |
| 1.2  | Problemdefinition und Zielsetzung .....                           | 3          |
| 1.3  | Aufbau der Arbeit .....   | 5          |
| <b>2</b>   | <b>3D-Echtzeitgrafik im World Wide Web .....</b>                  | <b>7</b>   |
| 2.1  | Einführung und Abgrenzung .....                                   | 8          |
| 2.2  | Anwendungsdomänen.....  | 14         |
| 2.3  | Internetbasierte 3D-Formate und -Technologien.....                | 20         |
| 2.4  | Entwicklungsstand und Trends.....                                 | 26         |
| <b>3</b>   | <b>Metaphern und Widgets für interaktive 3D-Anwendungen .....</b> | <b>28</b>  |
| 3.1  | 3D-Benutzungsoberflächen .....                                    | 29         |
| 3.2  | Gestaltungsrichtlinien für 3D-Anwendungen.....                    | 42         |
| 3.3  | <i>Action Spaces</i> : Konzept und Metaphern .....                | 50         |
| 3.4  | 3D-Widgets: Klassifikation und Systematisierung .....             | 55         |
| 3.5  | Spezifikation von 3D-Widgets: auf dem Weg zu Standards .....      | 84         |
| <b>4</b>   | <b>CONTIGRA: Konzeption einer 3D-Komponentenarchitektur.....</b>  | <b>87</b>  |
| 4.1  | Motivation und Anforderungsanalyse .....                          | 88         |
| 4.2  | Software-Komponentenmodelle und ihre Eignung für 3D-Grafik.....   | 90         |
| 4.3  | Verwandte Arbeiten – eine Klassifikation.....                     | 94         |
| 4.4  | Der dokumentenzentrierte Lösungsansatz .....                      | 99         |
| <b>5</b>   | <b>CONTIGRA: Das deklarative Dokumentenmodell .....</b>           | <b>105</b> |
| 5.1  | Einsatz der Extensible Markup Language (XML) .....                | 106        |
| 5.2  | CoApplication – Beschreibung von 3D-Applikationen .....           | 108        |
| 5.3  | CoComponent – Beschreibung der Schnittstelle .....                | 110        |
| 5.4  | CoComponentImplementation – Beschreibung der Implementierung..... | 115        |
| <b>6</b>   | <b>CONTIGRA: Der Autorenprozeß und seine Werkzeuge .....</b>      | <b>129</b> |
| 6.1  | Existierende 3D-Autorenwerkzeuge.....                             | 130        |
| 6.2  | CONTIGRA-Prozeßphasen und beteiligte Autoren .....                | 136        |
| 6.3  | Anforderungen an ein 3D-Autorenwerkzeug.....                      | 140        |
| 6.4  | Der CONTIGRABUILDER im Überblick .....                            | 142        |
| 6.5  | Die prototypische Realisierung .....                              | 148        |
| 6.6  | Beispielanwendungen .....   | 152        |
| <b>7</b>   | <b>Zusammenfassung und Ausblick.....</b>                          | <b>153</b> |
| 7.1  | Zusammenfassung der Kapitel und ihrer Beiträge .....              | 153        |
| 7.2  | Diskussion .....  | 156        |
| 7.3  | Zukünftige Arbeiten .....   | 157        |
| <b>Anhang, Abkürzungsverzeichnis, Webreferenzen, Literaturverzeichnis,<br/>Verzeichnis eigener Publikationen</b> |   |            |

## Einleitung und Zielstellung

Obwohl gegenwärtige, an der Desktop-Metapher orientierte *WIMP-Interfaces* (*Windows, Icons, Menus, Pointing Device*) sehr erfolgreich sind, werden sie nicht von allen Nutzern als natürlich empfunden. Sie verschenken durch uniform aussehende Menüsysteme oder stereotype Mauspositionierungs- und Klick-Handlungen menschliche Handlungseffizienz, die im Umgang mit realen Werkzeugen erworben wurde. Interaktive Echtzeitgrafik und 3D-Benutzungsoberflächen als Bestandteil von *Post-WIMP-Interfaces* erlauben in bestimmten Anwendungsbereichen ein intuitiveres Arbeiten, indem sie u.a. die menschliche Fähigkeit zur Raumorientierung besser nutzen oder virtuelle Werkzeuge anbieten. 3D-Benutzungsoberflächen (*3D Graphical User Interfaces*, 3D-GUIs) wurden bisher vor allem im Bereich Virtuelle Realität (VR) entwickelt, wo trotz langjähriger Forschungsarbeiten jedoch kaum Richtlinien und keine Standards für die Entwicklung dreidimensionaler Anwendungen existieren.

*3D-Widgets* – die Interaktionselemente eines 3D-GUIs – sind bisher unzureichend erforscht worden und stehen nicht in Form von Bibliotheken oder APIs zur freien Verfügung. Dabei erleichtern sie – als Mittler zwischen niedrigdimensionalen Eingabegeräten und höherdimensionalen Aufgaben im 3D-Raum – die Arbeit besonders im Bereich *Desktop-VR*. Das ist interaktive 3D-Grafik ohne komplizierte und ermüdende VR-Spezialhardware unter Nutzung von Standard-PCs. Gerade durch die dynamische Leistungsentwicklung von Prozessoren und Grafiksубsystemen sowie breitbandige Internet-Anbindungen erhält erstmals ein großer Nutzerkreis Zugang zu Web-basierten 3D-Anwendungen. Diese besitzen auch für Alltagsnutzer Erfolgspotential in Bereichen wie Produktvisualisierungen, E-Commerce-Anwendungen, Lehr-/Lernsystemen oder Architekturvisualisierungen.

Damit steht als wichtiger Teilbereich der Desktop-VR die dreidimensionale Echtzeitgrafik im World Wide Web (kurz *Web3D*) im Fokus der Arbeit. In einem einführenden Kapitel der Dissertation wird der Stand der Forschung und Technik auf diesem Gebiet näher beleuchtet. Bei der Betrachtung aktueller Anwendungsdomänen wird deutlich, welche Erfolgchancen Web3D-Grafik als wichtiger Bestandteil eines multimedialen und interaktiven Webs besitzt. Dies zeigt sich auch in der Vielfalt von Web-basierten 3D-Technologien, die sich in die beiden Lager der proprietären Formate kommerzieller Anbieter und der durch institutionelle Gremien entwickelten Standardformate aufteilen lassen. Einen Schwerpunkt bildet dabei der vom Web3D-Consortium konzipierte künftige Standard Extensible 3D (X3D) [X3D@]. Betrachtet man die Standardisierungsbemühungen für multimediale Web-Technologien (XHTML, SMIL, SVG, MPEG-4), zeichnet sich die Bedeutung von XML für die Interoperabilität der einzelnen Formate deutlich ab. Neben diesen Trends sind auch die gesteigerte visuelle Qualität von Web3D-Grafik und die Erweiterung der Zielplattformen auf mobile Endgeräte zu erwähnen.

Defizite existieren jedoch noch bei der Unterstützung eines multidisziplinären Autorenprozesses mit geeigneten 3D-Autorenwerkzeugen sowie in Form hoher Erstellungskosten aufgrund mangelnder Ansätze zur Wiederverwendbarkeit und Skalierbarkeit. Die Vielfalt proprietärer und standardisierter Web3D-Formate erzeugt Entwicklungsunsicherheit und Formatabhängigkeit, auch bieten 3D-Formate selten High-Level-Ansätze oberhalb des Szenengraphniveaus. Schließlich läßt sich konstatieren, daß die Definition von 3D-Sound und die Integration weiterer Medien bisher unzureichend adressiert wurden. Auch die Realisierung von Interaktivität und Objektverhalten in Web3D-Formaten wurde bisher ungenügend unterstützt oder ist nur eingeschränkt und mit externer Programmierung möglich.

**Zielstellung:** Vor diesem Hintergrund soll mit dieser Arbeit ein Beitrag dazu geleistet werden, daß interaktive 3D-Anwendungen in einfacher Weise aus wiederverwendbaren, standardisierten Bausteinen zusammengesetzt sind, wobei intuitive, interdisziplinär nutzbare Werkzeuge zum Einsatz kommen. Weniger imperative und primär visuelle Programmierung würde auch einem breiten Personenkreis die Erstellung derartiger Anwendungen ermöglichen. Neben der technischen Basis, die sich auf offene Standards gründen muß, sollen künftigen Entwicklern auch Richtlinien zur Erstellung interaktiver 3D-Applikationen, Designempfehlungen und ein Repertoire von Interaktionstechniken und 3D-Widgets zur Verfügung stehen.

Damit gliedert sich die Arbeit in zwei wesentliche Hauptbeiträge. Im ersten Teil werden zur Lösung der konzeptionell-gestalterischen Probleme von Desktop-VR-Anwendungen Gestaltungsrichtlinien, Metaphern zur räumlichen Gliederung von 3D-Anwendungen vorgestellt und schließlich 3D-Widgets in Form einer hierarchischen Systematik klassifiziert. Im zweiten Teil wird die CONTIGRA-Architektur zur Spezifikation von 3D-Widgets und zur Erstellung interaktiver 3D-Anwendungen vorgestellt. Der dokumentbasierte High-Level-Komponentenansatz basiert auf einem deklarativen XML-Dokumentenmodell und wird durch einen visuellen Autorenprozeß mit dem 3D-Autorenwerkzeug CONTIGRABUILDER unterstützt.

## **Teil 1: Metaphern und Widgets für interaktive 3D-Anwendungen**

Der erste Teil der Dissertation beschäftigt sich mit Metaphern und Widgets für interaktive 3D-Anwendungen im Bereich Desktop-VR und stellt damit das theoretische Fundament der Arbeit dar, wobei konzeptionelle Aspekte von 3D-GUIs im Mittelpunkt stehen. Zugrunde liegt die Analyse von etwa 240 wissenschaftlichen Arbeiten zu 3D-GUIs, VR-Anwendungen und Interaktionstechniken, bei denen die Möglichkeiten interaktiver 3D-Grafik für Post-WIMP-Schnittstellen untersucht wurden. Neben 3D-Betriebssystemoberflächen als Alternativentwicklungen zur Desktop-Metapher existieren auch andere Anwendungen, bei denen die gesamte Schnittstelle als 3D-Raum ausgeführt ist, z.B. 3D-Modellierungs- und Animationsprogramme, 3D-Dateibrowser und -Dokumentenmanager oder kollaborative Arbeitsumgebungen.

Der Nutzen von 3D-GUIs im Vergleich zu traditionellen WIMP-Interfaces wurde bisher sehr kontrovers zwischen optimistischer Verfechtung und kritischer Ablehnung diskutiert. Je nach konkreter Aufgabe, Anwendungsdomäne und eingesetzter Displaytechnologie resultieren vergleichende Arbeiten in Präferenz aber auch Unterlegenheit von 3D-GUIs [Cockb02]. Die in der Arbeit vorgenommene Analyse und Gegenüberstellung von 2D- und 3D-GUIs kann die Frage „3D or not 3D?“ [Shnei02] nicht abschließend beantworten, macht aber deutlich, daß für 3D-Anwendungen Fortschritte auf konzeptioneller und Implementierungsseite nötig sind, bevor sich die Effektivität von 3D-GUIs für bestimmte Anwendungsdomänen besser überprüfen läßt.

### **Gestaltungsrichtlinien für 3D-Anwendungen**

Die Komplexität des Designs von 3D-GUIs ist aufgrund der erweiterten Freiheitsgrade (Räumlichkeit, Perspektive, Kamera etc.) sehr hoch. 3D-Grafikanwendungen sind deutlich schwieriger zu entwerfen und zu implementieren als konventionelle Anwendungen, was sich auch bei der Gestaltung und Realisierung von 3D-Widgets zeigt [Leine97]. Richtlinien zum Entwurf von WIMP-Interfaces sind dabei nur teilweise übertragbar. So werden in dieser Arbeit generelle Designrichtlinien für die Entwicklung dreidimensionaler Anwendungen und

spezielle Richtlinien für 3D-Widgets entwickelt und detailliert beschrieben. Dazu zählen die Nutzung von 3D-Metaphern, die Vermeidung einer 1:1-Abbildung der Realität, ein Produktsprachlicher Ansatz, die Reduktion im Szenenlayout, visuelle Klarheit und Bildschirmeffektivität, vielfältige Medienintegration und die Schaffung von Interface-Identität.

### Metaphern zur Raumgliederung: Action Spaces

Die Abbildung 1 zeigt die Hauptbestandteile einer dreidimensionalen Benutzungsschnittstelle am Beispiel einer Desktop-VR-Anwendung. Zu bearbeitende bzw. zu betrachtende *3D-Dokumente* – wie Produkte oder abstrakte Informationsvisualisierungen – sind dabei zusammen mit den 3D-Interaktionselementen zu ihrer Manipulation (*3D-Widgets* [Conne92]) in räumliche Strukturen (*Action Spaces* [Dachs00]) eingebettet. Action Spaces sind virtuelle dreidimensionale Räume bzw. aufgabenzentrierte Zonen, zwischen denen Nutzer einfach navigieren können. 3D-Widgets werden darin zusammen mit ihren Containern um einen vordefinierten Blickpunkt gruppiert, womit sie Arbeitsplätzen aus der realen Welt (Küche, Werkstatt etc.) ähneln. Mit der effektiven Aufteilung des Bildschirms durch vordefinierte Anordnungen und Blickpunkte, der Integration von Widgets in Raumstrukturen und der vereinfachten Navigation zwischen einzelnen Bereichen der 3D-Anwendung werden wichtige Probleme gerade für Desktop-VR-Anwendungen gelöst. Abgerundet wird das Konzept durch die Vorstellung zahlreicher Raum- und Strukturmetaphern sowie geeigneter Navigationsmetaphern zum Wechsel der Räume.



Abbildung 1: Beispiel für einen Action Space eines 3D-GUI

### Klassifikation und Spezifikation von 3D-Widgets

Der Hauptfokus des ersten Teils der Arbeit liegt auf der systematischen Klassifikation und Spezifikation von *3D-Widgets* als zentrale Bestandteile von Desktop-VR-Anwendungen. Der allgemein beklagte Mangel an verfügbaren Widgets als Grundlage möglicher Standards sowie ungenügende Ansätze zu ihrer Systematisierung und Klassifikation [Leine97] zeigen den Bedarf an weiteren Forschungsarbeiten. Es muß erst eine Reihe etablierter Interaktionstechniken und Widgets vorhanden sein, bevor High-Level-Werkzeuge entworfen und sinnvolle 3D-Applikationen auf breiterer Basis entwickelt werden können [Myers00].

Neben den zahlreichen in die Klassifikation eingeflossenen Einzelarbeiten wurden als verwandte Arbeiten Widget-Bibliotheken und andere Klassifikationsansätze berücksichtigt. In [Kett93] werden als theoretisches Fundament 42 3D-Metaphern zur geometrischen Manipulation klassifiziert. In [Leine97] wird erstmalig ein Überblick über bis dahin existierende 3D-Widgets gegeben und eine grobe Unterteilung in geometrische Manipulation und Anwendungskontrolle vorgenommen. Wenige weitere Subklassifikationen von 3D-Widgets existieren, die teilweise in die Systematik eingeflossen sind, darunter die in [Watt98] zur direkten Objektmanipulation und in [Shnei98] zur Einteilung von Menütechniken. VR-Interaktionstechniken, für die mehrere Taxonomien (z.B. [Bowma99b]) vorgestellt wurden, fanden nur Berücksichtigung, wenn sie an eine geometrische Ausprägung gekoppelt sind.

Der Schwerpunkt der Klassifikation liegt auf interaktiven 3D-Widgets mit geometrischer Repräsentation und insbesondere auf Widgets zur Anwendungskontrolle. Die Klassifikation der

Widgets erfolgte nach dem Kriterium des Einsatzbereiches bzw. Interaktionszieles. Es wurden sowohl aus 2D-GUIs übernommene als auch speziell für den 3D-Raum entwickelte Widgets einbezogen. Tabelle 1 zeigt alle Klassifikationskategorien für 3D-Widgets im Überblick. Eine ausführliche Darstellung der Kategorien und vor allem der zugehörigen Widgets findet sich in der Dissertation, zusätzlich online unter [CONTIGRA@] eine aktuelle Fassung der Hierarchie. Die über 70 identifizierten Widgets werden nicht nur klassifiziert, sondern auch in einheitlicher Form spezifiziert. Erst dadurch ist ihre applikationsübergreifende Verwendung und flexible Kombination möglich, womit ein Beitrag zu Wiederverwendbarkeit und künftiger Standardisierung im Bereich der 3D-Benutzungsschnittstellen geleistet wird.

|                                       |  |
|---------------------------------------|--|
| <b>Direkte 3D-Objektinteraktion</b>   | <b>Objektselektion</b>                       |
|                                       | Direkte Selektion                            |
|                                       | Okklusionsselektion                          |
|                                       | Distanzselektion                             |
|                                       | <b>Geometrische Manipulation</b>             |
|                                       | Lineare Transformation                       |
|                                       | Nichtlineare Transformation                  |
|                                       | High-Level-Objektmanipulation                |
| <b>Manipulation der 3D-Szene</b>      | <b>Orientierung und Navigation</b>           |
|                                       | Direkte Blickpunktselektion                  |
|                                       | Geführter Transport                          |
|                                       | Miniaturdarstellungen                        |
|                                       | <b>Steuerung der Szenenpräsentation</b>      |
|                                       | Lichtmanipulation                            |
|                                       | Kameramanipulation                           |
|                                       | Soundsteuerung                               |
| <b>Exploration und Visualisierung</b> | Geometrische Exploration                     |
|                                       | Hierarchievisualisierungen                   |
|                                       | Visualisierungen von Graphen                 |
|                                       | Visualisierung von 2D-Daten und Dokumenten   |
|                                       | Wissenschaftliche Visualisierung             |
| <b>Anwendungskontrolle</b>            | <b>Zustandsänderung / Diskrete Wertgeber</b> |
|                                       | Aktivierung                                  |
|                                       | Zwei Zustände                                |
|                                       | Multiple Zustände                            |
|                                       | <b>Kontinuierliche Wertgeber</b>             |
|                                       | Skalare Werte                                |
|                                       | Multiple Werte                               |
|                                       | <b>Spezielle Werteeingaben</b>               |
|                                       | Farbwähler                                   |
|                                       | <b>Menüselektion</b>                         |
|                                       | Temporäre Optionsmenüs                       |
|                                       | Einzelmenüs                                  |
|                                       | Menühierarchien                              |
|                                       | <b>Container</b>                             |

Tabelle 1: Überblick der Klassifikationskategorien für 3D-Widgets

## Teil 2: CONTIGRA – eine deklarative 3D-Komponentenarchitektur

Im zweiten Teil der Dissertation wird die 3D-Komponentenarchitektur CONTIGRA (*Component-Oriented Three-dimensional Interactive Graphical Applications*) als praktischer Lösungsansatz zur Spezifikation von 3D-Widgets und zur Erstellung interaktiver 3D-Anwendungen vorgestellt.

### Verwandte Arbeiten

Nach der Anforderungsanalyse wurden etablierte Komponententechnologien aus der Softwaretechnik als Realisierungsbasis betrachtet. Die *Common Object Request Broker Architecture (CORBA)* und CORBA-Komponenten sind aufgrund des großen Kommunikationsoverheads und der Orientierung auf komplexe, verteilte Client-Server-Anwendungen für interaktive 3D-Applikationen kaum geeignet. Ähnliches gilt auch für die verteilten Kompo-

nenntenmodelle *Distributed Component Object Model (DCOM)* und *Enterprise JavaBeans (EJB)*. *ActiveX-Controls* und alle COM-basierten Technologien bieten weder Portabilität noch Plattformunabhängigkeit, *JavaBeans* und *EJB* sind sprachabhängig. Alle untersuchten Technologien sind zudem programmzentriert. Nur wenige unterstützen die direkte Einbindung von 3D-Grafik, weshalb mehrere Forschungsansätze versuchen, verschiedene Grafiktechnologien in Form spezieller 3D-Komponentenarchitekturen zu erweitern. Diese verwandten Arbeiten lassen sich in fünf Gruppen einteilen:

- Frühe Komponentenansätze (Open-Inventor-NodeKits, VRML97-Prototypen)
- Code-zentrierte Lösungen (NPSNET-V [Capps00], Bamboo [Watse98] u.a.)
- Systeme auf Basis von Komponententechnologien (Bamboo, 3D-Beans [Dörne00])
- Spezielle 3D-Komponentenansätze (i4D [Geige00], Smart Virtual Prototypes [Salme00])
- Dokumentzentrierte Ansätze (Jamal [Rudol99a])

Vor allem der Jamal-Ansatz mit seiner deklarativen Syntax beeinflusste die CONTIGRA-Architektur, die ebenfalls zur letzten Gruppe gezählt werden kann. Trotz ihrer Stärken besitzen die genannten Arbeiten als Hauptnachteil bestimmte Abhängigkeiten von konkreten 3D-Formaten, Komponententechnologien, Programmiersprachen oder Realisierungsplattformen. Wenige setzen auf Standards, nur einige bieten eine High-Level-Abstraktion oberhalb des Szenengraphs. Auch das im Projekt CHAMELEON ([Meißn01], [Wehne01]) entwickelte XML-Dokumentformat hat den CONTIGRA-Ansatz beeinflusst.

### **Konzeption der Komponentenarchitektur**

Für die Erstellung interaktiver 3D-Grafikapplikationen wird ein dokumentenzentrierter Komponentenansatz vorgeschlagen. 3D-Anwendungen und Komponenten werden dabei ausschließlich durch strukturierte Dokumente beschrieben, mit denen sich von der Schnittstelle über die Komponentenimplementierung bis hin zur Konfiguration, Komposition und Verknüpfung zu komplexeren Szenen alle Details konsequent deklarativ beschreiben lassen. Abbildung 2 zeigt für die CONTIGRA-Architektur die Komponentenentwicklungsebenen, die damit verbundenen Aufgaben, zugehörigen Dokumente und verwendeten Werkzeuge.

Die *Komponentenentwicklungsebene* umfaßt die Implementierung von 3D-Komponenten und die Spezifikation ihrer Schnittstelle für den Zugriff auf die Parameter der Komponente. Die Implementierung unterscheidet sich dabei insofern von der imperativen Softwareprogrammierung, daß Szenengraphen für die Beschreibung von Geometrie, Erscheinungsbild, Verhalten und auditiven Eigenschaften deklarativ erstellt werden. Die *Distributionsebene* umfaßt die Suche, Auswahl und Beschaffung von 3D-Komponenten über das Web. Auf der Ebene der *Anwendungsentwicklung* erfolgt zunächst die Anpassung und Konfiguration ausgewählter Komponenten. Diese werden dann zu formatunabhängigen 3D-Anwendungen oder komplexeren Komponenten zusammengefügt, wobei neben Subkomponenten auch weitere Szenengraphbestandteile und zusätzliche Funktionalität ergänzt werden können. Komponenten und Szenengraphbestandteile werden außerdem in Form verbindungsorientierter deklarativer Programmierung miteinander verknüpft. Aus den einzelnen Beschreibungsdokumenten kann im Autorenwerkzeug CONTIGRABUILDER oder über externe Transformationen schließlich eine auf der *Laufzeitebene* ausgeführte Applikation für ein konkretes 3D-Zielformat generiert werden. Durch die Trennung von Komponenten- und Anwendungsentwurf in Form verschiedener Dokumente wird Wiederverwendbarkeit explizit unterstützt.

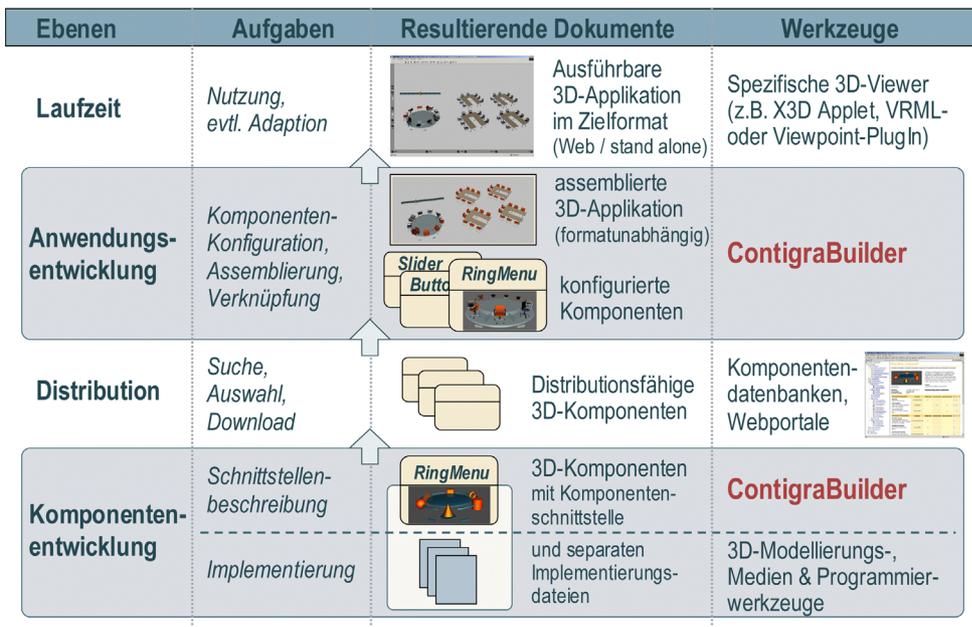


Abbildung 2: Komponentenentwicklungsebenen der CONTIGRA-Architektur

### Die deklarativen Beschreibungssprachen

Das Herz des CONTIGRA-Ansatzes stellen die auf der Basis der *Extensible Markup Language (XML)* entwickelten Auszeichnungssprachen für die einzelnen Komponentenebenen dar. Abbildung 3 zeigt noch einmal die Aufgaben der Ebenen, dazu die jeweils entwickelten XML-Schemata und ihre korrespondierenden Instanzdokumente, die wiederum verschiedene Szenengraphdateien und auch Dateien anderen Typs referenzieren.

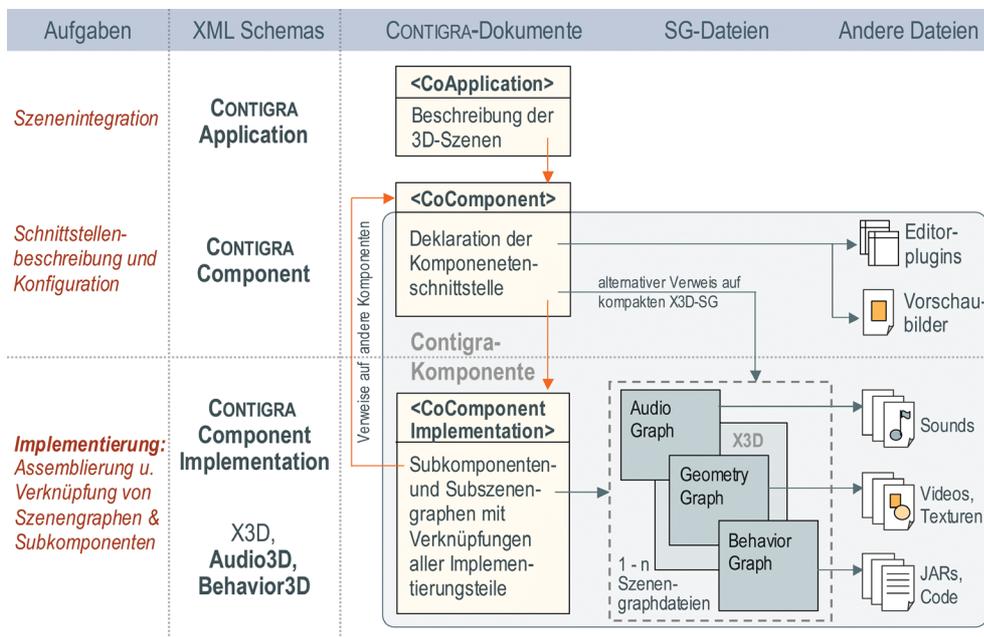


Abbildung 3: Die Auszeichnungssprachen und korrespondierenden Dokumente für die Ebenen der CONTIGRA-Architektur

Auf der obersten Ebene ist *CoApplication* als Beschreibungssprache für interaktive 3D-Applikationen angesiedelt, mit der sich typische Szenenparameter festlegen lassen. Für 3D-Komponenten existieren zwei Schemata, da jede CONTIGRA-Komponente aus einem Schnitt-

stellendokument mit konfigurierbaren High-Level-Parametern und Metainformationen (*Co-Component*) sowie einem Implementierungsdokument (*CoComponentImplementation*) besteht. Die Implementierung umfaßt eine Transformationshierarchie von Subkomponenten, die drei Szenengraphen *Geometry*, *Audio* und *Behavior* sowie Verknüpfungsinformationen für alle Bestandteile. Während für den Geometrieteil der künftige Web3D-Standard X3D [X3D@] zum Einsatz kommt, werden für die anderen Teile die eigens entwickelten Sprachen *Audio3D* [Hoffm03] und *Behavior3D* [Dachs03a] verwendet. Mit *Audio3D* können komplexe akustische Szenen unabhängig von 3D-Sound-APIs beschrieben werden, wobei die Deklaration spezialisierter Soundquellen und verbundener Räume wichtige Neuerungen darstellen. Mit *Behavior3D* wird ein flexibles, objektorientiertes Konzept zur primär deklarativen Modellierung des Verhaltens von 3D-Objekten vorgestellt, wobei Erstellung und Verwendung neuer Verhaltensknoten automatisiert sind und zahlreiche neuartige Knoten implementiert wurden.

### Der Autorenprozeß und seine Werkzeuge

Um 3D-Komponenten bzw. 3D-Anwendungen erstellen und damit Instanzdokumente der CONTIGRA-Schemata visuell bearbeiten zu können, wurde ein geeigneter Autorenprozeß mit Werkzeugunterstützung konzipiert. Der CONTIGRA-Autorenprozeß umfaßt in fünf Phasen die beiden Hauptaufgaben Komponenten- und Anwendungsentwicklung, an denen verschiedene Autorengruppen beteiligt sind. Als Ergebnis der Analyse existierender 3D-Autorenwerkzeuge ergab sich die Notwendigkeit einer eigenen Werkzeugentwicklung für den deklarativen CONTIGRA-Ansatz. Der Prototyp des CONTIGRABUILDER ist in Abbildung 4 zu sehen.

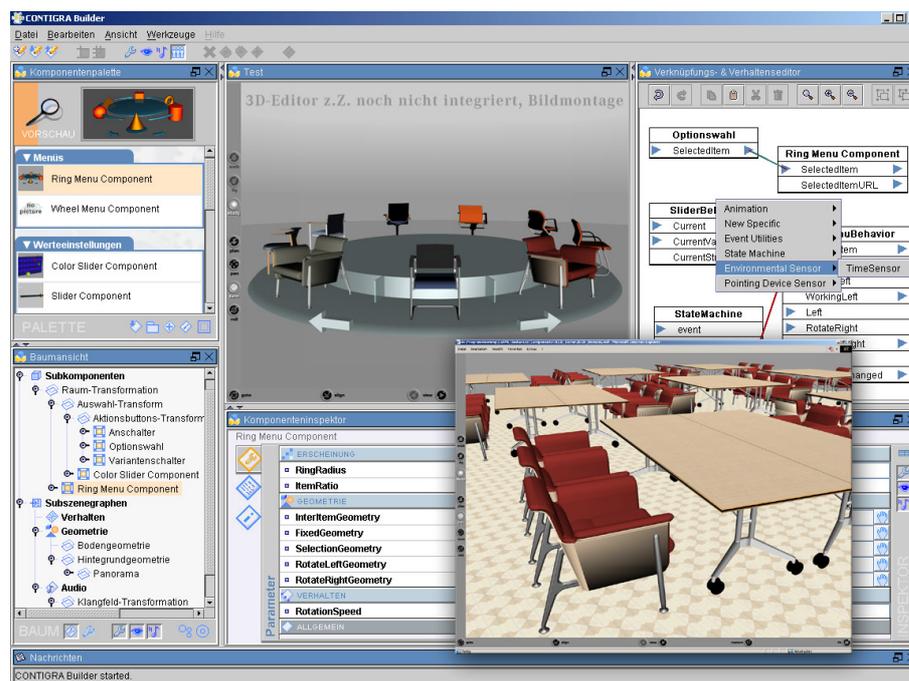


Abbildung 4: 3D-Autorenwerkzeug CONTIGRABUILDER und erzeugte VRML97-Beispielanwendung

Er unterstützt mehrere Autorenrollen und Erweiterungen über eine Plug-In-Architektur. Zu den prototypisch realisierten Editoren zählen die *Komponentenpalette* (links oben) zur Auswahl und Anordnung von 3D-Komponenten, die *Baumansicht* (darunter) mit Komponenten- und Subszenengraphen, der *Komponenteninspektor* (unten) zum Editieren von Parametern und ein *Verknüpfungs- und Verhaltenseditor* (rechts oben). Der zentral angeordnete *3D-Editor* befindet sich noch in Entwicklung.

Die mit dem CONTIGRABUILDER erstellten XML-Dokumente können schließlich mit Hilfe von Objektmodell-gestützten oder auf XSLT-Stylesheets basierenden Transformationen in verschiedene Zielformate für das WWW überführt werden, womit die praktische Umsetzbarkeit und Flexibilität des Ansatzes demonstriert wird. Abbildung 4 zeigt im überlagerten Fenster auch eine der Beispielanwendungen im Format VRML97. Ein prototypisches Komponentenportal zur Distribution von Komponenten im Web rundet die praktische Realisierung ab.

## Zusammenfassung

Das Gebiet Desktop-VR mit dem Fokus Web3D-Anwendungen wurde in dieser Arbeit näher betrachtet. Es ist deutlich geworden, daß sowohl technologische als auch konzeptionell-gestalterische Probleme den erfolgreichen Einsatz dreidimensionaler Anwendungen behindern. Deshalb wurden für 3D-Anwendungen Gestaltungsrichtlinien vorgestellt, ein Konzept zur metaphorbasierten Gliederung virtueller Räume entwickelt und zahlreiche 3D-Widgets klassifiziert und spezifiziert. Damit steht ein systematisiertes Repertoire an wiederverwendbaren Bausteinen zur Verfügung, die sich anpassen, zusammenfügen und zu interaktiven Anwendungen verknüpfen lassen. Dazu wurde eine dokumentzentrierte, mehrschichtige Komponentenarchitektur mit einem deklarativen, auf XML-Grammatiken basierenden Dokumentenmodell entwickelt. Ergänzend wurde schließlich ein geeigneter Autorenprozeß konzipiert und ein visuelles 3D-Autorenwerkzeug prototypisch implementiert. Die wichtigsten Resultate der Arbeit konnten in mehreren internationalen Veröffentlichungen vorgestellt werden ([Dachs01b], [Dachs02], [Dachs03a], [Hoffm03]). Zusammengefaßt sind folgende wissenschaftlichen Hauptbeiträge zu nennen:

- Klassifikation und realisierungsunabhängige Spezifikation von 3D-Widgets als Beitrag zur Weiterentwicklung und Standardisierung von 3D-Benutzungsoberflächen.
- Dokumentbasierter High-Level-Komponentenansatz oberhalb des Szenengraphniveaus zur vereinfachten Erstellung von interaktiven 3D-Anwendungen und flexiblen 3D-GUIs.
- Durchgängig deklaratives Dokumentenmodell auf XML-Basis zur homogenen Beschreibung von 3D-Anwendungen, Komponentenschnittstellen, -konfigurationen, -kompositionen und -verknüpfungen sowie zur Verhaltensrealisierung und Raumklangdefinition.
- Wiederverwendbarkeit und Anpaßbarkeit auf verschiedenen Ebenen (Komponenten, Sub-szenengraphen, Verhaltensbausteine), auch durch Trennung des internen Szenengraphen in die drei Subgraphen Geometrie, Audio und Verhalten.
- Integration innovativer Ansätze zur deklarativen Beschreibung von Raumklang und zur deklarativen Verhaltensmodellierung.

Zukünftige Arbeiten ergeben sich in den Bereichen Widget-Entwicklung und Standardisierung, beim Ausbau von Behavior3D und Audio3D, der Weiterentwicklung und Evaluation des Autorenwerkzeuges und der Unterstützung verschiedener Zielplattformen bzw. von Adaptionmöglichkeiten.

## Literatur

- [CONTIGRA@] Webseiten des Forschungsprojektes CONTIGRA: <http://www.contigra.de/>  
 [X3D@] Extensible 3D (X3D) - International Standard ISO/IEC FCD 19775:200x Final Committee Draft: [http://www.web3d.org/fs\\_specifications.htm](http://www.web3d.org/fs_specifications.htm)

- [Bowma99b] Bowman, Doug A.: *Interaction Techniques for Common Tasks in Immersive Virtual Environments: Design, Evaluation, and Application*. Dissertation, Juni 1999; Georgia Institute of Technology.
- [Capps00] Capps, Michael; McGregor, Don; Brutzman, Don; Zyda, Michael: NPSNET-V: A New Beginning for Dynamically Extensible Virtual Environments. In: *IEEE Computer Graphics and Applications*, Vol. 20 / 5 (2000), S. 12-15.
- [Cockb02] Cockburn, Andy; McKenzie, Bruce: Evaluating the Effectiveness of Spatial Memory in 2D and 3D Physical and Virtual Environments. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*: ACM Press, New York, S. 203-210, 20.-25.4. 2002.
- [Conne92] Conner, D. Brookshire; Snibbe, Scott S.; Herndon, Kenneth P.; Robbins, Daniel C.; Zeleznik, Robert C.; van Dam, Andries: Three-Dimensional Widgets. In: *Proceedings of the ACM Symposium on Interactive 3D Graphics*: ACM Press, New York, S. 183-188, März 1992.
- [Dachs00] Dachzelt, Raimund: Action Spaces - A metaphorical concept to support navigation and interaction in 3D interfaces. In: *Proceedings of the Workshop "Usability Centred Design and Evaluation of Virtual 3D Environments"*: Shaker Verlag Aachen, 13./14. April 2000.
- [Dachs01b] Dachzelt, Raimund: *CONTIGRA: A High-Level XML-Based Approach to Interactive 3D Components*. In: *SIGGRAPH 2001 Conference Abstracts and Applications*, Computer Graphics Annual Conference Series. Los Angeles, California (USA): ACM Press, New York, 2001, S. 163.
- [Dachs02] Dachzelt, Raimund; Hinz, Michael; Meißner, Klaus: Contigra: An XML-based Architecture for Component-oriented 3D Applications. In: *Proceedings of the Seventh International Conference on 3D Web Technology*: ACM Press, New York, S. 155-163, 24.-28. Februar 2002.
- [Dachs03a] Dachzelt, Raimund; Rukzio, Enrico: Behavior3D: An XML-based Framework for 3D Graphics Behavior. In: *Proceedings of the Eighth International Conference on 3D Web Technology*: ACM Press, New York, S. 101-112, 9.-12. März 2003.
- [Dörne00] Dörner, Ralf; Grimm, Paul: Three-dimensional Beans - Creating Web Content Using 3D Components in a 3D Authoring Environment. In: *Proceedings of the Fifth Symposium on Virtual Reality Modeling Language (Web3D-VRML)*: ACM Press, New York, S. 69-74, 2000.
- [Geige00] Geiger, Christian; Paelke, Volker; Reimann, Christian; Rosenbach, Waldemar: A Framework for the Structured Design of VR/AR Content. In: *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*: ACM Press, New York, S. 75-82, Oktober 2000.
- [Hoffm03] Hoffmann, Heiko; Dachzelt, Raimund; Meißner, Klaus: An Independent Declarative 3D Audio Format on the Basis of XML. In: *Proceedings of the 9th International Conference on Auditory Display*: Boston University Publications Production Department, S. 99-102, Juli 2003.
- [Kettn93] Kettner, Lutz: *Mathematisch-Informationstheoretische Untersuchung von 3D-Metaphern*. Diplomarbeit, Dezember 1993; Universität Karlsruhe, Institut für Betriebs- und Dialogsysteme.
- [Leine97] Leiner, Ulrich; Preim, Bernhard; Ressel, Stephan: Entwicklung von 3D-Widgets - Überblicksvortrag. In: *Proceedings of Simulation and Animation*: SCS Europe, Erlangen, S. 170-188, 6.-7. März 1997.
- [Meißn01] Meißner, Klaus; Röttger, Simone; Wehner, Frank: Dynamische Visualisierung modularer, XML-basierter Kursdokumente. In: *Proceedings of the Workshop "11. Arbeitstreffen der GI-Fachgruppe 1.1.5/7.0.1 'Intelligente Lehr-/Lernsysteme'"*, 2001.
- [Myers00] Myers, Brad; Hudson, Scott E.; Pausch, Randy: Past, Present, and Future of User Interface Software Tools. In: *ACM Transactions on Computer-Human Interaction (TOCHI)*, Vol. 7 / 1 (2000), S. 3-28.
- [Rudol99a] Rudolph, Mark; Jamal: Components Frameworks and Extensibility: <http://www.web3d.org/TaskGroups/x3d/lucidActual/jamal/Jamal.html>, 1999
- [Salme00] Salmela, Marko; Kyllönen, Harri: Smart Virtual Prototypes: Distributed 3D Product Simulations for Web based Environments. In: *Proceedings of the Fifth Symposium on Virtual Reality Modeling Language (Web3D-VRML)*: ACM Press, New York, S. 87-93, 2000.
- [Shnei02] Shneiderman, Ben: "3D or Not 3D: When and Why Does it Work?" Keynote. In: *Proceedings of the Seventh International Conference on 3D Web Technology*, 24.-28. Februar 2002.
- [Watse98] Watsen, Kent; Zyda, Mike: Bamboo - A Portable System for Dynamically Extensible, Real-Time, Networked, Virtual Environments. In: *Proceedings of the Virtual Reality Annual International Symposium*, S. 252-259, 1998.
- [Watt98] Watt, Alan; Policarpo, Fabio: *The Computer Image*. ACM Press, New York & Addison-Wesley. 1998. ISBN 0-201-42298-0.
- [Wehne01] Wehner, Frank; Lorz, Alexander: Developing Modular and Adaptable Courseware Using TeachML. In: *Proceedings of the ED-MEDIA, World Conference on Educational Multimedia, Hypermedia & Telecommunications*: Association for the Advancement of Computing in Education (AACE), Juni 2001.
- [Zelez00] Zeleznik, Robert C.; Holden, Loring; Capps, Michael; Abrams, Howard; Miller, Tim: Scene-Graph-As-Bus: Collaboration between Heterogeneous Stand-alone 3-D Graphical Applications. In: *Proceedings of the Eurographics 2000 Conference*: The Eurographics Association and Blackwell Publishers, 2000.