

Eine deklarative Komponentenarchitektur und Interaktionsbausteine für dreidimensionale multimediale Anwendungen

Raimund Dachzelt

Technische Universität Dresden
Fakultät Informatik
Lehrstuhl für Multimediatechnik
01062 Dresden
dachzelt@acm.org

Abstract: Multimediale, interaktive 3D-Grafik besitzt Erfolgspotential für zahlreiche Anwendungen, z.B. Lehr-/Lernsysteme oder künftige Betriebssystemoberflächen. Bisher existieren jedoch kaum Richtlinien und Standards für deren Entwicklung. Der erste Teil der Dissertation¹ führt in die Thematik dreidimensionaler Benutzungsschnittstellen ein. Mit der Beschreibung von Gestaltungsrichtlinien und Metaphern sowie einer Systematisierung existierender 3D-Interaktionselemente soll dem Entwickler die Konzeption von 3D-Anwendungen erleichtert werden. Für deren Erstellung wird im zweiten Teil eine dokumentenzentrierte Komponentenarchitektur auf XML-Basis vorgestellt. Mit Hilfe eines visuellen 3D-Autorenwerkzeuges ist es nun möglich, interaktive 3D-Anwendungen – auch mit Raumklang – aus wiederverwendbaren, standardisierten Bausteinen zusammenzusetzen. Die resultierenden Anwendungen können schließlich in verschiedene 3D-Zielformate, vor allem für das World Wide Web, transformiert werden.

1 Einleitung und Motivation

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 Bereichen ein intuitiveres Arbeiten, indem sie u.a. die menschliche Fähigkeit zur Raumorientierung besser nutzen. Aktuelle Prototypen dreidimensionaler Betriebssystemoberflächen, beispielsweise von Sun und Microsoft, unterstreichen das künftige Potential. 3D-Benutzungsoberflächen (*3D Graphical User Interfaces, 3D-GUIs*) wurden bisher jedoch vor allem im Bereich Virtuelle Realität (VR) entwickelt, wo

¹ Dieser Beitrag faßt die Dissertation gleichen Titels zusammen, die auch als Buch [Da04] erschienen ist.

trotz langjähriger Forschungsarbeiten keine Standards für die Entwicklung dreidimensionaler Anwendungen existieren. Die wenigen allgemeinen *3D-Widgets* – Interaktionselemente eines 3D-GUIs – 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 VR-Spezialhardware unter Nutzung von Standard-PCs. Gerade durch die Leistungsentwicklung von Prozessoren und Grafiksyste-men sowie breitbandige Internet-Anbindungen erhält erstmals ein großer Nutzerkreis Zugang zu Web-basierten, multimedialen 3D-Anwendungen. Damit steht als wichtiger Teilbereich der Desktop-VR die 3D-Echtzeitgrafik im World Wide Web (kurz *Web3D*) im Fokus der Dissertation und wird mit Anwendungsgebieten und einem bewertenden Technologieüberblick in einem einführenden Kapitel näher beleuchtet.

Defizite existieren jedoch nach wie vor 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. 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.

Vor diesem Hintergrund war es das Ziel dieser Arbeit, interaktive 3D-Anwendungen in einfacher Weise aus wiederverwendbaren, standardisierten Bausteinen zusammensetzen zu können, wobei intuitive, interdisziplinär nutzbare Werkzeuge zum Einsatz kommen. Neben der technischen Basis, die sich auf offene Standards gründen muß, sollten 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 Dissertation [Da04] in zwei wesentliche Hauptbeiträge. Im ersten Teil werden zur Konzeption von Desktop-VR-Anwendungen Gestaltungsrichtlinien und Metaphern zur räumlichen Gliederung von 3D-Anwendungen vorgestellt sowie 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 beschrieben. Der dokumentbasierte High-Level-Komponentenansatz basiert auf einem deklarativen XML-Dokumentenmodell, wird durch einen visuellen Autorenprozeß mit dem 3D-Autorenwerkzeug CONTIGRA-BUILDER unterstützt und durch Beispielanwendungen überprüft.

2 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. Zugrunde liegt die Analyse von etwa 240 wissenschaftlichen Arbeiten zu 3D-GUIs, VR-Anwendungen und Interaktionstechniken. Neben 3D-Betriebssystemoberflächen als Alternativen zur Desktop-Metapher kann interaktive 3D-Grafik

auch erfolgversprechend für andere Anwendungen genutzt werden, bei denen die gesamte Schnittstelle als 3D-Raum ausgeführt ist, darunter 3D-Modellierungs- und Animationsprogramme, 3D-Dateibrowser und -Dokumentenmanager oder kollaborative Arbeitsumgebungen. Der Nutzen von 3D-GUIs im Vergleich zu traditionellen WIMP-Interfaces wurde bisher kontrovers diskutiert. Die in der Dissertation vorgenommene Analyse und Gegenüberstellung von 2D- und 3D-GUIs kann die Frage „3D or not 3D?“ [Sh02] 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. Den Vorteilen von 3D-GUIs, wie räumlichem Orientierungs- und Handlungswissen, verbesserter Aufmerksamkeitssteuerung und höherer Attraktivität stehen konzeptionelle bzw. Navigations- und Interaktionsprobleme gegenüber. Die dafür in der Arbeit entwickelten Lösungsansätze werden in den folgenden Kapiteln zusammengefaßt.

2.1 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 [LPR97]. So werden in dieser Arbeit generelle Designrichtlinien für die Entwicklung dreidimensionaler Anwendungen und spezielle Richtlinien für 3D-Widgets entwickelt und detailliert beschrieben. Dazu zählen u.a. die Nutzung von 3D-Metaphern, die Vermeidung einer 1:1-Abbildung der Realität, ein produktsprachlicher Ansatz, visuelle Klarheit, reduziertes Szenenlayout und Bildschirmeffektivität, vielfältige Medienintegration und die Schaffung von Interface-Identität.

2.2 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 sind dabei zusammen mit den Interaktionselementen zu ihrer Manipulation (3D-Widgets [Co92]) in räumliche Strukturen (Action Spaces [Da00]) 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 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



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

für Desktop-VR-Anwendungen gelöst. Das Konzept wird durch die Vorstellung zahlreicher Raum- und Strukturmetaphern sowie geeigneter Navigationsmetaphern abgerundet.

2.3 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 Mangel an verfügbaren Widgets als Grundlage möglicher Standards sowie ungenügende Ansätze zu ihrer Systematisierung und Klassifikation motivieren diese Arbeit. 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 [MHP00]. Neben zahlreichen in die Klassifikation eingeflossenen Einzelarbeiten wurden als verwandte Arbeiten Widget-Bibliotheken und andere Klassifikationsansätze berücksichtigt. In [LPR97] wird erstmalig ein Überblick über die bis dahin existierenden 3D-Widgets gegeben. Wenige weitere Subklassifikationen von 3D-Widgets – z.B. zur direkten Objektmanipulation – existieren, die teilweise in die Systematik eingeflossen sind. Reine VR-Interaktionstechniken, für die bereits mehrere Taxonomien (z.B. [Bo99]) vorgestellt wurden, fanden nur dann Berücksichtigung, wenn sie an eine geometrische Ausprägung gekoppelt sind.

Der Schwerpunkt der Klassifikation lag somit auf interaktiven 3D-Widgets mit geometrischer Repräsentation und insbesondere auf Widgets zur Anwendungskontrolle sowie 3D-Menüs. 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 Interaktionselemente einbezogen. Tabelle 1 zeigt alle Klassifikationskategorien für 3D-Widgets im Überblick. Eine ausführliche Darstellung der Kategorien und der jeweils zugehörigen Widgets findet sich in der Dissertation, online unter [3DW@] auch eine aktuelle Fassung der Hierarchie. Die über 70 identifizierten Widgets wurden nicht nur klassifiziert, sondern auch in einheitlicher Form mit ihren Parametern / ihrer Funktion spezifiziert. Erst dadurch ist ihre applikationsübergreifende (Wieder-)Verwendung und flexible Kombination im Sinne der künftigen Standardisierung von 3D-Benutzungsschnittstellen möglich. Eine Auswahl der spezifizierten und bereits implementierten 3D-Widgets aus dem Standardrepertoire ist in Abbildung 2 zu sehen.



Abbildung 2: Auswahl spezifizierter und implementierter 3D-Widgets zur Anwendungskontrolle

Direkte 3D-Objektinteraktion <ul style="list-style-type: none"> Objektselektion <ul style="list-style-type: none"> Direkte Selektion Okklusionsselektion Distanzselektion Geometrische Manipulation <ul style="list-style-type: none"> Lineare Transformation Nichtlineare Transformation High-Level-Objektmanipulation 	Exploration und Visualisierung <ul style="list-style-type: none"> Geometrische Exploration Hierarchievisualisierungen Visualisierungen von Graphen Visualisierung von 2D-Daten und Dokumenten Wissenschaftliche Visualisierung
Manipulation der 3D-Szene <ul style="list-style-type: none"> Orientierung und Navigation <ul style="list-style-type: none"> Direkte Blickpunktselektion Geführter Transport Miniaturdarstellungen Steuerung der Szenenpräsentation <ul style="list-style-type: none"> Lichtmanipulation Kameramanipulation Soundsteuerung 	Anwendungskontrolle <ul style="list-style-type: none"> Zustandsänderung / Diskrete Wertgeber <ul style="list-style-type: none"> Aktivierung Zwei Zustände Multiple Zustände Kontinuierliche Wertgeber <ul style="list-style-type: none"> Skalare Werte Multiple Werte Spezielle Werteeingaben <ul style="list-style-type: none"> Farbwähler Menüselektion <ul style="list-style-type: none"> Temporäre Optionsmenüs Einzelmenüs Menühierarchien Container

Tabelle 1: Überblick der Klassifikationskategorien für 3D-Widgets

3 CONTIGRA – eine deklarative 3D-Komponentenarchitektur

Im zweiten Teil der Dissertation wird die 3D-Komponentenarchitektur CONTIGRA (*Component-Oriented Three-dimensional Interactive Graphical Applications*) [CON@] als praktischer Lösungsansatz zur Spezifikation von 3D-Widgets und zur Erstellung interaktiver, multimedialer 3D-Anwendungen vorgestellt. Nach der Anforderungsanalyse wurden zunächst etablierte Komponententechnologien (u.a. CORBA, DCOM, EJB) aus der Softwaretechnik als Realisierungsbasis betrachtet. Diese sind aufgrund des großen Kommunikationsoverheads und der Orientierung auf komplexe, verteilte Client-Server-Anwendungen für interaktive 3D-Applikationen nur wenig geeignet. In mehreren Forschungsansätzen wurde versucht, verschiedene Grafiktechnologien in Form spezieller 3D-Komponentenarchitekturen zu erweitern. Diese verwandten Arbeiten lassen sich in fünf Gruppen einteilen: frühe Komponentenansätze, Code-zentrierte Lösungen, Systeme auf Basis von Komponententechnologien, spezielle 3D-Komponentenansätze, dokumentenzentrierte Ansätze. Vor allem letztere mit ihrer deklarativen Syntax beeinflussen die CONTIGRA-Architektur, die ebenfalls zur letzten Gruppe gezählt werden kann. Trotz ihrer Stärken besitzen die in [Da04] detailliert betrachteten verwandten 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.

3.1 Konzeption der Komponentenarchitektur

Für die Erstellung interaktiver 3D-Grafikapplikationen wird ein dokumentenzentrierter Komponentenansatz vorgeschlagen [DHM02]. 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 alle Details konsequent deklarativ beschreiben lassen. Abbildung 3 zeigt die Komponentenentwicklungsebenen der CONTIGRA-Architektur, die damit verbundenen Aufgaben, zugehörigen Dokumente und verwendeten Werkzeuge.

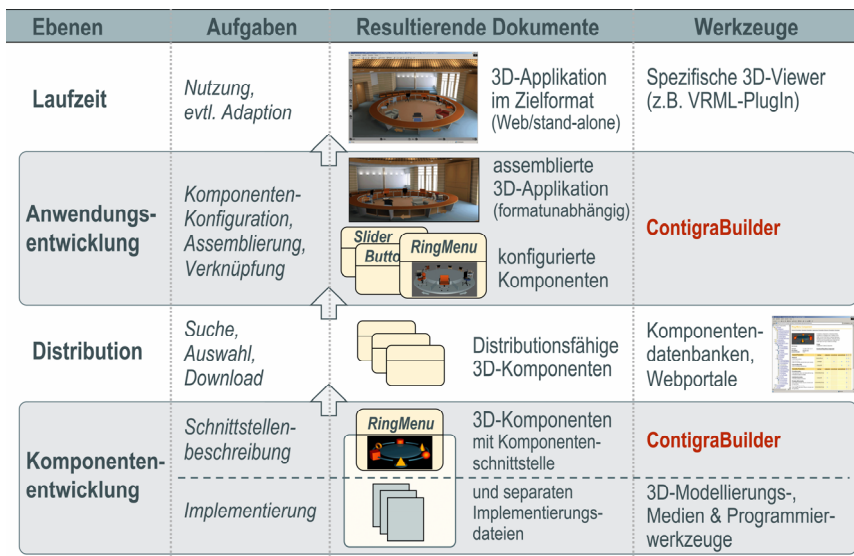


Abbildung 3: Komponentenentwicklungsebenen der Contigra-Architektur

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.

3.2 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 4 zeigt im oberen Teil die eigens entwickelten XML-Schemata (Ausnahme: X3D) und darunter ihre korrespondierenden Instanzdokumente, die wiederum verschiedene Szenengraphdateien sowie externe Dateien referenzieren.

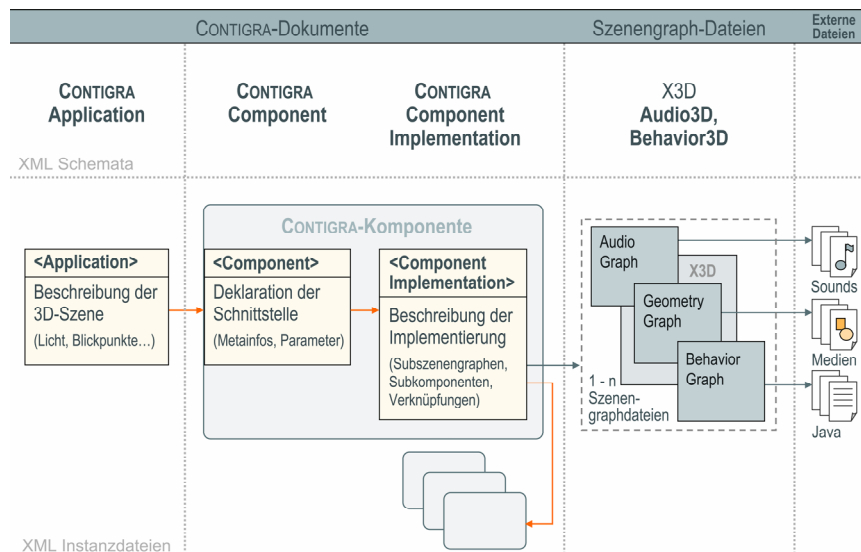


Abbildung 4: Die Auszeichnungssprachen und korrespondierenden Dokumente für die Ebenen der CONTIGRA-Architektur

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 Schnittstellendokument mit konfigurierbaren High-Level-Parametern und Metainformationen (*CoComponent*) 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 Web3D-Standard X3D zum Einsatz kommt, werden für die anderen Teile als multimediale Erweiterung die eigens entwickelten Sprachen *Audio3D* [HDM03] und *Behavior3D* [DR03] verwendet. Mit *Audio3D* können komplexe akustische Szenen (Soundquellen, Hindernisobjekte, Räume, Hörer, Hallparameter etc.) 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 und der Interaktion mit der Szene vorgestellt, wobei Erstellung und Verwendung neuer Verhaltensknoten automatisiert sind und zahlreiche neuartige Knoten in Anlehnung an SMIL und X3D implementiert wurden.

3.3 Der Autorenprozeß und das Werkzeug CONTIGRABUILDER

Um 3D-Komponenten bzw. 3D-Anwendungen erstellen und damit Instanzdokumente der CONTIGRA-Schemata visuell bearbeiten zu können, wurde ein fünfphasiger Autorenprozeß mit den beiden Hauptaufgaben Komponenten- und Anwendungsentwicklung konzipiert. 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 5 zu sehen.

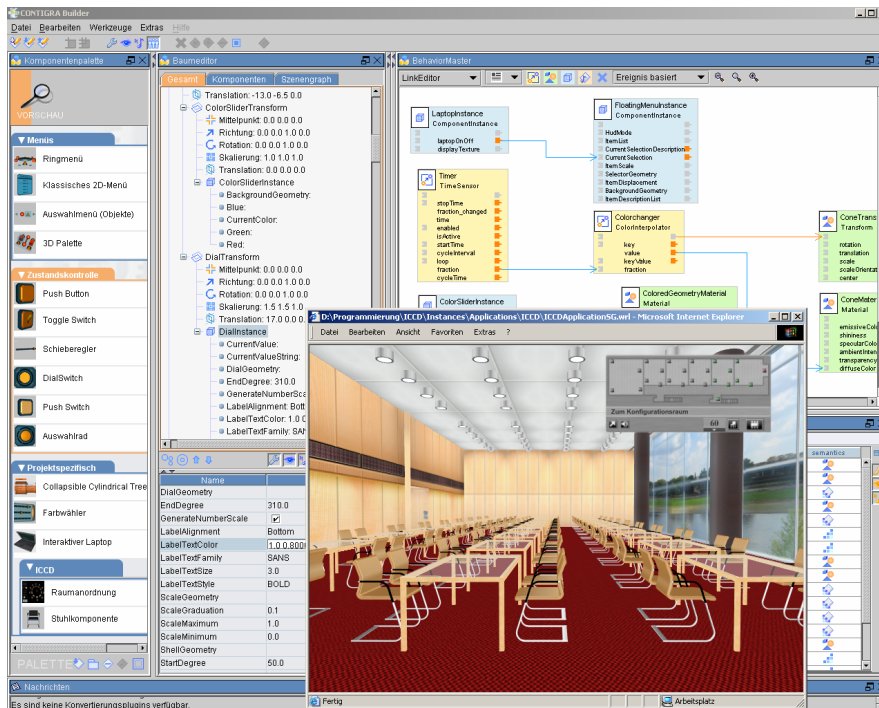


Abbildung 5: 3D-Autorenwerkzeug CONTIGRABUILDER und erzeugte VRML97-Beispielanwendung

Das in Java realisierte Werkzeug bietet visuelle Editoren, darunter eine *Komponentenpalette* (ganz links), den *Zentralen Baum- und Parametereditor* (daneben) sowie den *Verknüpfungs- und Verhaltenseditor* (rechts oben). Zudem unterstützt es mehrere Autorenrollen sowie Erweiterungen über eine Plug-In-Architektur. Die mit dem CONTIGRABUILDER erstellten XML-Dokumente können schließlich durch Objektmodell-gestützte oder auf XSLT-Stylesheets basierende Transformationen flexibel in verschiedene Zielformate überführt werden. Realisiert sind bereits Transformationen nach HTML (für das prototypische CONTIGRA-Komponentenportal [3DW@]), VRML97, X3D und OpenSG, in Arbeit sind Java3D und MPEG-4. Die praktische Umsetzbarkeit des Ansatzes wurde an mehreren Beispielanwendungen demonstriert. Dazu zählt eine komplexe 3D-Anwendung zur Planung von Raumlayouts für ein internationales Kongreßzentrum (Beispielansicht im überlagerten Fenster der Abbildung 5). Deren Realisierung durch 15

Komponenten mit mehreren hundert Instanzen sowie zahlreichen Interaktionsmöglichkeiten zeigt die Skalierbarkeit des CONTIGRA-Ansatzes, der vor allem für die Anwendungsklassen Produktpräsentationen/E-Commerce, Lehr-/Lernanwendungen sowie Informationsvisualisierung und Navigation geeignet ist.

4 Zusammenfassung

Das Gebiet Desktop-VR mit dem Fokus Web3D-Anwendungen stand im Mittelpunkt dieser Arbeit. Für 3D-Anwendungen wurden Gestaltungsrichtlinien vorgestellt, ein Konzept zur metaphorbasierten Gliederung virtueller Räume entwickelt und zahlreiche 3D-Widgets klassifiziert und spezifiziert. Damit steht nun ein systematisiertes Repertoire an wiederverwendbaren Bausteinen zur Verfügung, die sich anpassen, zusammenfügen und zu interaktiven 3D-Anwendungen verknüpfen lassen. Dafür wurden eine dokumentenzentrierte, mehrschichtige Komponentenarchitektur mit einem deklarativen, auf XML-Grammatiken basierenden Dokumentenmodell sowie ein visuelles 3D-Autorenwerkzeug entwickelt. Zusammengefaßt sind folgende wissenschaftlichen Hauptbeiträge zu nennen:

- Klassifikation und realisierungsunabhängige Spezifikation von 3D-Widgets als Beitrag zur Weiterentwicklung und Standardisierung von 3D-GUIs.
- Dokumentbasierter High-Level-Komponentenansatz oberhalb des Szenengraphniveaus zur vereinfachten Erstellung von interaktiven 3D-Anwendungen.
- Durchgängig deklaratives Dokumentenmodell auf XML-Basis zur homogenen Beschreibung von 3D-Anwendungen, Komponentenschnittstellen, -konfigurationen, -kompositionen und -verknüpfungen.
- Wiederverwendbarkeit und Anpaßbarkeit auf verschiedenen Ebenen, auch durch Trennung des internen Szenengraphen in die Teile Geometrie, Audio u. Verhalten.
- Integration neuartiger Ansätze zur deklarativen Beschreibung von Raumklang und zur deklarativen Verhaltens- und Interaktionsmodellierung.

Zukünftige Arbeiten ergeben sich im Bereich der weiteren Entwicklung, Standardisierung und Evaluation von 3D-Widgets/Benutzungsoberflächen, beim Ausbau der multimedialen Möglichkeiten (z.B. Audio3D, MPEG-4), der Weiterentwicklung des Autorenwerkzeuges CONTIGRABUILDER und der Unterstützung von Adaptionsmechanismen für verschiedene Nutzergruppen, Zielplattformen und Endgeräte.

Literaturverzeichnis

- [3DW@] Online-Klassifikation u. Spezifikation von 3D-Widgets: <http://www.3d-components.org>
- [CON@] Webseiten des Forschungsprojektes CONTIGRA: <http://www.contigra.de/>
- [Bo99] Bowman, Doug A.: Interaction Techniques for Common Tasks in Immersive Virtual Environments: Design, Evaluation, and Application. Dissertation, Juni 1999; Georgia Institute of Technology.
- [Co92] 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.

- [Da00] 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, April 2000.
- [Da04] Dachzelt, Raimund: Eine deklarative Komponentenarchitektur und Interaktionsbausteine für dreidimensionale multimediale Anwendungen. Der Andere Verlag, Tönning; ISBN 3-89959-271-9, Dezember 2004.
- [DHM02] 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.
- [DR03] 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.
- [HDM03] 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.
- [LPR97] Leiner, Ulrich; Preim, Bernhard; Ressel, Stephan: Entwicklung von 3D-Widgets - Überblicksvortrag. In: Proceedings of Simulation und Animation: SCS Europe, Erlangen, S. 170-188, 6.-7. März 1997.
- [MHP00] 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.
- [Sh02] Shneiderman, Ben: "3D or Not 3D: When and Why Does it Work?" Keynote. In: Proceedings of the Seventh International Conference on 3D Web Technology, Febr. 2002.



Raimund Dachzelt, Jahrgang 1971, studierte Informatik an der Technischen Universität Dresden und der University of Glasgow. Während des Studiums waren interaktive Computergrafik und Informationsvisualisierung Interessenschwerpunkte. Im postgradualen Ergänzungsstudiengang Designinformatik an der Burg Giebichenstein, Hochschule für Kunst und Design Halle, entstanden in interdisziplinärer Zusammenarbeit VR-Anwendungen mit 3D-Benutzungsschnittstellen, u.a. für die Mercedes Benz AG. Nach einer Stipendiatenzeit innerhalb eines DFG-Graduiertenkollegs ist der Autor seit 2001 als wissenschaftlicher Mitarbeiter am Lehrstuhl für Multimediatechnik der TU Dresden beschäftigt. Streaming Medien, Web3D-Grafik, Dokumentformate und Informationsvisualisierung stehen dabei im Mittelpunkt von Forschung und Lehre. Dr. Dachzelt ist Mitglied der ACM SIGGRAPH, GI und des Web3D-Consortiums. Er ist als Mitorganisator internationaler Workshops sowie als Gutachter für verschiedene Konferenzen tätig.