

# **Verteilte und kooperative VR-Anwendungen**

von Michael Sedlmair

## **Abstract**

Mit verteilten und kooperativen VR-Anwendungen ermöglicht man geographisch voneinander entfernten Menschen sich in einer virtuellen Umgebung zu treffen, dort miteinander zu interagieren und kollaborativ zusammenzuarbeiten.

In dieser Arbeit soll aufgezeigt werden, wie sich solche Anwendungen realisieren lassen und inwiefern sie heute bereits sinnvoll eingesetzt werden können. Hierzu wird im Folgenden kurz auf allgemeine technische Anforderungen an die Netzwerke eingegangen. Außerdem wird die 3D-Beschreibungssprache VRML und mit DIVE ein Framework zur Erstellung von verteilten VR-Welten vorgestellt. Danach wird anhand einiger Anwendungsbeispiele aus verschiedenen Bereichen aufgezeigt, welche Chancen in dieser neuartigen Technologie stecken.

Bei der bereits heute unüberschaubaren Menge an verschiedenen Anwendungen und der Komplexität ihrer Strukturen, würde eine ausführliche Abhandlung dieses Themas den Rahmen der Arbeit sprengen. Es soll daher nur ein repräsentativer Überblick über die Materie gegeben werden.

## **1. Einleitung**

Die Zusammenarbeit von geographisch voneinander entfernten Menschen hat in der Vergangenheit - und wird es wohl auch in Zukunft - immer mehr an Bedeutung gewonnen, wertvolle Zeit kann eingespart werden und effektiveres Arbeiten wird ermöglicht.

Verteilte virtuelle Anwendungen, in denen es möglich ist dreidimensionale Objekte darzustellen und diese kollaborativ mit anderen Benutzern zusammen interaktiv zu verwenden, obgleich man womöglich viele tausende Kilometer voneinander entfernt ist, erscheinen hierbei als eine weiteres sehr lukratives Anwendungsgebiet.

Nicht zu vernachlässigen ist aber auch die Möglichkeit, sich ohne jeglichen kommerziellen Hintergrund im „Cyberspace“ zu treffen, man kann dort zusammen Spaß haben oder sich einfach in einer virtuellen Umgebung treffen und kommunizieren. Im folgenden sollen nun die Grundlagen für solch ein Unterfangen kurz erläutert und einige Anwendungen vorgestellt werden.

## **2. Netzwerke für verteilte VR-Anwendungen**

### **2.1. Datennetze**

Signifikant für die Netzwerkkommunikation in verteilten VR-Anwendungen ist die Menge der zu übertragenden Daten. Es ist daher naheliegend sich für sehr leistungsstarke Netze, wie z.B. ein ATM-Netzwerk, zu entscheiden, um den Datentransfer so effizient wie möglich zu gestalten. Andererseits wäre es aber auch wünschenswert, die Anforderungen für verteilte VR-Anwendungen weit herabzusetzen, sie z. B. schon über einen normalen ISDN-Anschluss

zugänglich zu machen. Somit könnten entsprechende Anwendungen von zahlreichen Menschen genutzt und sogar flächendeckend verbreitet werden.

## 2.2. Netzwerktopologie

Netzwerke können in Client/Server-Modelle (siehe Abb. 1) und Peer-to-Peer-Architekturen (siehe Abb. 2) eingeteilt werden.

Ersteres stellt eine einfach zu implementierende Kommunikation zwischen einem Client (Dienst-Nutzer) zu einem Server (Dienst-Anbieter) dar. Dabei stellt der Client eine Anfrage an den Server, der diese bearbeitet und das Ergebnis dem Client zurücksendet. Ein Server besitzt und synchronisiert dabei die gesamte Datenbank (VR-Welt), außerdem stellt er normalerweise seine Dienste mehreren Clients zur Verfügung. Für VR-Anwendungen kann daher eine hohe Konsistenz<sup>1</sup> gewährleistet werden. Da der aktuelle Datensatz jedoch nur auf dem Server liegt und viele Clients auf einmal auf diesen zugreifen möchten, wird der Datenbankzugriff zum Flaschenhals. Dies wirkt sich problematisch auf die Reaktionszeit aus, wodurch verteilte VR-Welten gegebenenfalls nicht mehr in Echtzeit genutzt werden könnten. Dies ist jedoch bei verteilten VR-Anwendungen meist unerlässlich, deswegen wird meist auf die Peer-to-Peer-Architektur zurückgegriffen.

Beim Peer-to-Peer-Modell verzichtet man auf einen zentralen Server, jeder Host fungiert als Client, der mit anderen Clients über ein Netz verbunden ist. Jeder Anwender erhält eine eigene Kopie der Datenbank und versendet bei Änderungen Updates an alle weiteren Anwender. Die Reaktionszeit wird dadurch enorm verringert, Flaschenhals ist nun die Netzwerkbandbreite, da ständig kommuniziert werden muss, um die Konsistenz der VR-Welt auf den verschiedene hosts sicherzustellen.

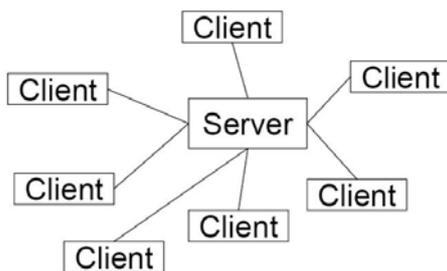


Abb. 1: Zentrale Client-Server-Architektur [1]

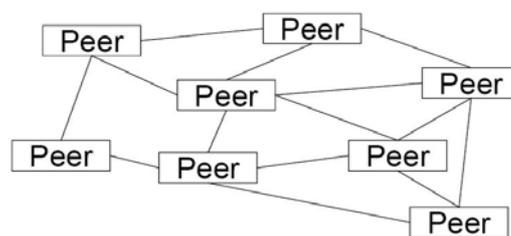


Abb. 2: Dezentrale Peer-to-Peer-Architektur [1]

Um die Vorteile der Client/Server- sowie der Peer-to-Peer-Architektur zu verbinden und die jeweiligen Nachteile zu minimieren, greift man auf Hybrid-Lösungen zurück. Hierbei wird eine virtuelle Welt in mehrere voneinander unabhängige, räumliche Teilbereiche unterteilt die jeweils einen eigenen Server erhalten.

Abschließend kann man sagen, dass für genügend kleine verteilte VR-Anwendungen meist eine peer-to-peer Verbindung gewählt wird und jeder Rechner eine komplette Kopie der virtuellen Welt erhält. Somit wird eine überlastete Kommunikation zu einem zentralen Server vermieden und die VR-Welten werden auf jedem Host explizit dem aktuellen Zustand angepasst. Bei größeren Anwendungen überschreitet die Größe der virtuellen Welt jedoch meist das Fassungsvermögen der einzelnen Host. Man greift dann auf eine Client-Server Architektur oder eine Hybrid-Lösung zurück. [1][2][3]

<sup>1</sup> Alle Teilnehmer einer verteilten VR-Anwendung sollten die virtuelle Welt zu jedem Zeitpunkt in gleichem Zustand vorfinden.

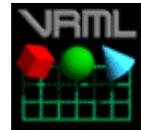
### 2.3. Dead Reckoning

Um die Nachrichtenzahl (bei peer-to-peer-Architekturen) zu reduzieren, wird bei VR-Environments oft ein Vorhersagemechanismus verwendet: Das dead reckoning. Objektpositionen werden dabei nicht konstant aktualisiert, sondern auf Basis ihres vorherigen Zustands berechnet. Dies findet lokal auf jedem Rechner statt, ohne dass andere über das Netzwerk kontaktieren werden müssen. Man kann dadurch die Netzlast verringern, es kann jedoch nicht garantiert werden, dass alle Anwender zur gleichen Zeit den gleichen Objektzustand sehen (Konsistenz). Für Sachobjekte, wie z. B. Fahrzeuge, eignet sich dieser Algorithmus hervorragend, zur Beschreibung von Avataren<sup>2</sup>, die in ihrem Verhalten nicht berechenbar sind, ist er aber eher ungeeignet. Diese müssen wie bisher in periodischen Abständen allen Clients Ihren aktuellen Zustand senden. [3]

## 3. VRML – Virtual Reality Modelling Language

### 3.1. Was ist VRML?

VRML ist eine plattformunabhängige 3D/4D<sup>3</sup>-Beschreibungssprache für Internet-Anwendungen. VRML ist, ähnlich wie HTML, dateiorientiert, d.h. VRML-Dateien werden aus dem Internet oder lokal geladen und dann ausgeführt. Die Struktur von VRML basiert auf der Verwendung von Szenengraphen mit beliebig ineinander verschachtelten Knoten<sup>4</sup>. VRML-Dateien benötigen einen speziellen Browser, der ihre Inhalte abspielen kann. Diese werden dann entweder als Plugin in den herkömmlichen Browser geladen (z. B. Cosmo Player) oder arbeiten als Standalone-Anwendungen (z. B. Liquid Reality). [4][5]



### 3.2. Geschichte

1995 wurde der erste VRML-Standard VRML 1.0 vorgestellt. In diesem Standard lassen sich jedoch lediglich statische 3D-Objekte und –Welten darstellen. Einzige Interaktionsmöglichkeit ist die Verlinkung von einzelnen Objekten eine Realisierung von verteilten Systemen ist dabei noch nicht möglich.

Mit VRML 2.0 wurde dann ein Jahr später die Möglichkeit zur Erschaffung von dynamischen Welten mit einbezogen, so dass der User mit seiner Umgebung interagieren kann. Durch die Anbindung an die Programmiersprache JAVA wird dem Anwender außerdem die Chance gegeben, verteilte VR-Systeme mit Hilfe von VRML (und JAVA) zu kreieren.

Ein Standard für Multiuser-Anwendungen aufbauend auf VRML existiert dagegen noch nicht. Mit der Verwirklichung dieses dritten großen Ziels befasste sich die Gruppe „Living Worlds“ und ein entsprechender Entwurf wurde auch bereits im April 1997 vorgelegt. Aufgrund der steigenden Komplexität weiterer Entwicklungen kam die Arbeit des Living Worlds Projekts jedoch zum ruhen.

Letztendlich hat VRML als Standard für verteilte 3D-Anwendungen im Internet in den letzten Jahren erheblich an Bedeutung verloren und ist bereits weitgehend durch das neue offene Echtzeit-3D-Datenformat für das Internet - X3D - abgelöst worden. In den jüngsten Tagen (August 2004) wurde nun das vom World Wide Web Consortium (W3C) entwickelte X3D als ISO Standard abgesegnet. [6][7]

### 3.3. Living Worlds

Die Arbeit des „Living Worlds“ Konsortiums zur Multiuser-Erweiterung des VRML-Standards forderte insbesondere zwei wichtige Aspekte: Interpersonalität und

<sup>2</sup> Ein Avatar ist ein Vertreter des Benutzers in der virtuellen Umgebung, eine virtuelle Identität des Nutzers.

<sup>3</sup> Unter Einbeziehung der Zeit als vierte Komponente.

<sup>4</sup> Näheres hierzu wurde bereits in anderen Referaten behandelt.

Interoperabilität. VRML-Szenen sollten sowohl von mehreren geographisch voneinander entfernten Nutzern gleichzeitig verwendet werden können (Interpersonalität), als auch von verschiedenen Entwicklern unabhängig erweiterbar sein (Interoperabilität). Als spezielle Komponente zur Verwirklichung dieser Anforderungen wurde das MUTech konstruiert, welches auf jedem Client vorhanden ist und dort die Kommunikation zu den anderen involvierten Clients übernimmt. Außerdem legten die Entwickler großen Wert darauf, dass zur Darstellung verteilter VRML-Anwendung als Betriebsmittel ein normaler VRML 2.0 Browser genügen sollte. [7][8]

### **3.4. X3D**

Extensible 3D, kurz X3D, ist eine Modifikation des VRML97 Standards und gilt als dessen zukunftsweisender Nachfolger. Wie VRML, ist auch X3D speziell für verteilte Anwendungen konstruiert worden. Die Neuerung dabei ist die Möglichkeit, die Beschreibung von 3D-Modellen auf die Syntax der Metasprache XML aufzubauen und damit die Generierung von Multimedia-Daten und deren Austausch über das Internet zu erleichtern. Andererseits bleibt jedoch eine vollständige Abwärtskompatibilität zum VRML97 Standard erhalten, so dass weiterhin VRML97-Anwendungen und Werkzeuge verwendet werden können. [9][10]

## **4. DIVE**

### **4.1. Was ist DIVE?**

DIVE ist ein C geschriebenes Framework zur Erstellung von verteilten virtuellen Welten. DIVE wurde 1991 am Swedish Institut of Computer Science (SICS) entwickelt und als Forschungsprototyp frei zugänglich gemacht. Dies bedeutet allerdings auch, dass es auf eine Wartung oder Weiterentwicklung dieser experimentellen Plattform keinerlei Garantie gibt.



Das Konzept von DIVE beruht auf der Interaktion von Avataren. Diese können sich frei innerhalb einer VR-Welt sowie zwischen verschiedenen Welten bewegen. Mit DIVE erstellte Welten sind skalierbar, d. h. zu jedem Zeitpunkt können Benutzer hinzukommen oder austreten. Eine VR-Welt existiert dabei genau so lange, solange noch mindestens ein User aktiv ist. Ursprünglich wurde DIVE als reine Desktop-VR Anwendung entwickelt, mittlerweile wurde es aber bereits für verschiedene spezielle VR Ein- und Ausgabegeräte, wie z. B. Datenhandschuh, CAVE, usw., erweitert. [11][12]

### **4.2. Vernetzung von DIVE**

Die Vernetzung von DIVE-Applications findet über das Internet statt. Beim erstmaligen Eintreten in eine VR-Welt lädt sich der Benutzer die jeweiligen Bilder und Sounds von einem Server herunter und gründet eine neue Benutzergruppe für diese Welt. Existiert die VR-Welt bereits, lädt sich der User diese von einem anderen User herunter und installiert lokal eine Kopie der Welt. Die Clients kommunizieren nun unabhängig vom Server innerhalb ihrer Gruppe, es entsteht eine peer-to-peer-Architektur. Durch kontinuierliche Updates in periodischen Abständen werden dabei die Welten auf den einzelnen Hosts auf ihren aktuellen Zustand gebracht. Als Übertragungsverfahren wurde Multicast gewählt, so dass jeweils genau die benötigten Informationen an die anfordernden Benutzer versendet werden kann. [11][12]

### **4.3. Aufbau von DIVE**

DIVE-Welten bestehen aus verschiedene geometrischen Objekten, die mit Texturen und Lichtern versehen werden. Organisiert werden diese Objekte mit Hilfe von Szenengraphen

und Knoten. Über ein TCL<sup>5</sup>-Skript kann jedem Objekt ein (oder mehrere) Verhalten zugeordnet werden, das bei Eintritt eines Ereignisses<sup>6</sup> ausgelöst wird.

Die Übertragung der Szenen auf den Blickwinkel des Anwenders (Rendering) übernimmt in DIVE der sogenannte VISUALIZER. Der VISUALIZER liest die verschiedenen I/O-Geräte aus und berechnet entsprechend den physischen Bewegungen die logische Tätigkeit des Benutzers in der DIVE-Umgebung. [11][12]

#### 4.4. DIVE-Conference

Die DIVE-Conference ist ein Anwendungsbeispiel, das aufbauend auf DIVE erstellt wurde. Das Szenario bildet einen virtuellen Konferenzraum ab, in dem sich mehrere Nutzer in Form von Avataren treffen können. Als Interaktionsmöglichkeiten stehen Aktionen wie gehen, zeigen oder nicken zur Verfügung. [11]



Abb. 2: DIVE-Conference [11]

## 5. Anwendungsbeispiele

### 5.1. Virtual Training: Cosimir VR

Verteilte virtuelle Umgebungen können dazu verwendet werden, um für geographisch voneinander entfernte Menschen ein gemeinsames kollaboratives, virtuelles Arbeitsumfeld zu simulieren. Cosimir VR ist eine virtuelle Trainingssimulation für Astronauten, die es den Astronauten ermöglicht von verschiedenen Orten aus imaginäre Szenarien virtuell miteinander durchzuspielen. Es können dabei mehrere Benutzer gleichzeitig in derselben virtuellen



Abb. 3: Multiuser-Anwendung Cosimir [13]

Umgebung arbeiten und sich dort, ohne akute Gefährdung der Personen, mit alltäglichen als auch weniger alltäglichen Situationen vertraut machen. So können beispielsweise wichtige interaktive Arbeitsvorgänge trainiert oder Notfallsituationen simuliert werden. [13]

### 5.2. Virtual Meetings: Starbright-Projekt

Menschen können mit Hilfe der VR-Technik in virtuellen Räumen zusammentreffen und dort kommunizieren und interagieren. Virtual Meetings sind im gesellschaftlichen wie industriellen Bereich bereits verbreitet, so kann man sich bereits im Internet in Virtuellen Kommunen treffen oder Stände auf virtuellen Messen besuchen.

Steven Spielberg entwickelte in Zusammenarbeit mit der Starbright Foundation ein Projekt, das es schwer kranken Kindern, die durch lange Krankenhausaufenthalte von ihrem sozialen Umfeld abgeschnitten werden, ermöglicht sich in einer virtuellen Umgebung mit Leidensgenossen zu treffen. Spielberg kreierte hierfür drei verschiedenen Fantasie-Welten (Tropical-, Cave-, Skyworld), in denen sich die Kinder treffen und miteinander spielen

<sup>5</sup> TCL ist eine erweiterbare Programmiersprache. TCL organisiert in DIVE das Verhalten und die Verteilung von Objekten.

<sup>6</sup> Ereignisse sind z. B.: Interaktion, Kollision, Timer,...

können. Zusätzlich lässt sich über ein Videokonferenzsystem direkt miteinander kommunizieren. Ziel dieses mit 20 Mio. Dollar geförderten Projekts ist es, den Kindern eine Ablenkung zum tristen Krankenhaus-Alltag zu geben und dadurch den Heilungsprozess zu beschleunigen. Das System wurde bereits an einigen Hospitälern in den USA installiert, inwiefern es seinen Erwartungen gerecht werden wird, wird sich zeigen. [14]



Abb 4: Starbright Projekt, Eingabe [14]



Abb 5: Starbright Projekt, Ausgabe [14]

### 5.3. Multiplayer Games: Quake III

Bereits weit verbreitet sind 3D-Spiele die über das Internet zu mehreren Personen gespielt werden können. Es handelt sich dabei zumeist um Desktop VR-Anwendungen, die über einen handelsüblichen PC und mit klassischen Ein-/Ausgabegeräten ohne Probleme gespielt werden können. Oft werden sogenannte Egoshoooter genutzt, bei denen man sich in einem Reich von bösen Kreaturen und Mitspielern befindet und als Ziel deren Auslöschung hat. Ein Beispiel ist das 1999 von idSoftware vorgestellte QuakeIII, das speziell als Multiplayer-Game für viele Anwender designed wurde.

VR-Multiplayer-Spiele erfreuen sich immer größerer Beliebtheit, so dass bei den Neuentwicklungen kein Ende abzusehen ist. Mit großer Erwartung blickt man in der entwickelnden Industrie als auch bei den Usern auf die kommerzielle Einführung von brauchbaren VR-Devices, die einem ein tieferes Eintauchen in die imaginären Spielwelten ermöglichen werden. [15][16]



Abb. 6: Quake III, verschieden Spielsituationen [13]

#### **5.4. Weitere Anwendungsgebiete**

Auf heute bereits wichtige Anwendungen von verteilten virtuellen Systemen trifft man in der Medizin, man denke dabei an die Möglichkeit von Fern-Operationen, bei denen ein Arzt geographisch vom Patienten entfernt ist und diesen über spezielle Interfaces und Maschinen behandeln kann. Besonders wichtig und darum auch die Technik vorantreibend ist der Einsatz von verteilten und kollaborativen VR-Anwendungen beim Militär. Es lassen sich so beispielsweise virtuelle Schlachtfelder modellieren auf denen strategische Manöver interaktiv trainiert werden können. Man könnte die Liste von Anwendungsmöglichkeiten verteilter VR-Technologien hier nun ins Unermessliche fortsetzen, dies sei an dieser Stelle aber der Eigenrecherche und/oder Fantasie des Lesers überlassen.

#### **6. Fazit und Ausblick**

Abschließend ist zu sagen, dass verteilte und kooperative VR-Anwendungen bereits heute ein Standbein in viele bereiche unserer Gesellschaft gesetzt haben und sich dort auch durchaus schon profiliert haben. Im Rahmen der Desktop-VR ist dabei sicherlich mit einer schnellen Verbreitung zu rechnen. Die gesellschaftlich Einführung voll-immersiver VR-Anwendungen stellt sich hingegen weitaus problematischer dar, da hierbei immer ein sehr hohes Budget von Nöten ist. So ist mittelfristig wohl noch nicht mit einer Einführung solcher Anwendungen in unser soziales Umfeld zu rechnen.

## 7. Quellenangabe

### Netzwerke

- [1] Vorlesungsskript: Informatik III, Universität München, Prof. Linnhoff-Popien  
<http://www.nm.informatik.uni-muenchen.de/Vorlesungen/ws0102/info3.shtml#Skript>
- [2] Vorlesungsskript: Einführung in die Informatik B, Universität Karlsruhe, Prof. Studer  
<http://www.aifb.uni-karlsruhe.de/Lehrangebot/Winter1998-99/Info-B/k43.pdf>
- [3] Vorlesung Spezielle Kapitel aus Computergraphik - Verteilte Graphische Applikationen, Universität Linz, Christoph Anthes  
<http://www.gup.uni-linz.ac.at/vga/VGA040609-06-CVE.pdf>

### VRML

- [4] Vorlesungsskript: Multimedia-Systeme, Universität Oldenburg, Dietrich Bole  
<http://www-is.informatik.uni-oldenburg.de/~dibo/teaching/mm97/script/node33.html>
- [5] VRML am ICA/CSV, Universität Stuttgart  
<http://www.csv.ica.uni-stuttgart.de/vrml/>
- [6] Holodeck im Web, VRML 2.0: lebendiges 3D, Andreas Steinbuch  
<http://www.heise.de/ct/96/07/043/>
- [7] golem.de, IT-News für Profis  
<http://www.golem.de/0408/32932.html>
- [8] Living Worlds Technical Tutorial, Mitra Technology Consulting  
[http://www.mitra.biz/vrml/lw/lw\\_tutor.htm](http://www.mitra.biz/vrml/lw/lw_tutor.htm)
- [9] Net-Lexikon von akademie.de, X3D  
<http://www.lexikon-definition.de/X3D.html>
- [10] X3D, Diplomarbeit  
[http://www.technik-vermitteln.de/x3d/extended\\_3d.pdf](http://www.technik-vermitteln.de/x3d/extended_3d.pdf)

### DIVE

- [11] The DIVE Home Page, Swedish Institut of Computer Science (SICS)  
<http://www.sics.se/dive/>
- [12] DIVE – ein internet-basiertes Virtual-Reality-System für viele Nutzer, TU Chemnitz, Stefan Ziegler  
<http://archiv.tu-chemnitz.de/pub/2003/0074/data/>

### Anwendungsbeispiele

- [13] EF-Robertertechnik, COSIMIR-VR, Multimedia Presentation  
[http://www.cosimir.com/VR/German/VirtualTraining/weltraum/weltraum\\_multiuser.htm](http://www.cosimir.com/VR/German/VirtualTraining/weltraum/weltraum_multiuser.htm)
- [14] Medien Praktisch, aus Heft 3/97, Seite 25-29: Cyberspace selbst gestalten, Virtual Worlds für Kinder  
<http://www.medienpraktisch.de/amedienp/mp3-97/3-97schind.htm>
- [15] Quake und Doom  
<http://www.quake.de/>
- [16] Wikipedia, die freie Enzyklopädie, Quake  
<http://de.wikipedia.org/wiki/Quake>