



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

## **Masterarbeit**

Jan Napitupulu

Ein System mit skalierbarer Visualisierung zur Entwicklung  
kollaborativer Serious Games

Jan Napitupulu

Ein System mit skalierbarer Visualisierung zur Entwicklung  
kollaborativer Serious Games

Masterarbeit eingereicht im Rahmen der Masterprüfung  
im Studiengang Informatik  
am Studiendepartment Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Kai von Luck  
Zweitgutachter: Prof. Dr. Gunter Klemke

Abgegeben am 20. Juni 2008

**Jan Napitupulu**

**Thema der Masterarbeit**

Ein System mit skalierbarer Visualisierung zur Entwicklung kollaborativer Serious Games

**Stichworte**

CSCW, Serious Games, Powerwall, Collaborative Workspace, Gruppentraining

**Kurzzusammenfassung**

Im Zusammenhang mit dem Begriff Serious Games wird untersucht, wie sich Computerspiele für Anwendungsbereiche einsetzen lassen, in denen nicht (primär) die Unterhaltung im Vordergrund steht. Ein Schwerpunkt liegt dabei auf der Bereitstellung von geeigneten konzeptionellen und technologischen Infrastrukturen, die für eine spielbasierte Aus- bzw. Fortbildung von Personengruppen eingesetzt werden können. An dieser Stelle kommt es zu Überschneidungen mit den Forschungen im Bereich der computergestützten Gruppenarbeit, in denen u.a. die Entwicklung von Co-located Collaborative Workspaces verfolgt wird. Eines der Konzepte zur Unterstützung der Gruppenprozesse in solchen Arbeitsumgebungen ist die Verwendung von großen und hochauflösenden Visualisierungssystemen, die als gemeinsam genutzte Arbeitsbereiche fungieren. Im Rahmen dieser Arbeit wird untersucht, wie sich solche Systeme realisieren und effizient für kollaborative Serious Games einsetzen lassen. Als Ergebnis wird mit dem Co-located Collaborative Training Environment eine computergestützte Umgebung präsentiert, die für spielbasierte Trainingsmaßnahmen von Gruppen optimiert ist.

**Jan Napitupulu**

**Title of the paper**

A system with scalable visualization for the development of collaborative serious games

**Keywords**

CSCW, Serious Games, Powerwall, Collaborative Workspace, Group Training

**Abstract**

The transferability of computer games to areas of application, where entertainment is not (primarily) focused on, is examined in the context of serious games. One of the main areas is the supply of suitable conceptual and technological infrastructures, which can be used for a game-based education or advanced training by groups of people. At this point, there is some overlapping with the research area of computer-supported collaborative work, in which among other things the development of co-located collaborative workspaces is pursued. One of the concepts is to support the group processes in these working environments by large-scaled, high-resolution visualization systems, used as shared workspaces. This thesis examines how these systems can be realized and efficiently used for collaborative serious games. As a result, a co-located collaborative training environment will be presented, which is a computer-supported environment optimized for game-based training programs targeting groups.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
1.1	Motivation . . . . .	3
1.2	Zielsetzung . . . . .	5
1.3	Aufbau der Arbeit . . . . .	5
<b>2</b>	<b>Grundlagen</b>	<b>7</b>
2.1	Computergestützte Gruppenarbeit (CSCW) . . . . .	7
2.1.1	Begriffsabgrenzungen . . . . .	8
2.1.2	Klassifizierung von CSCW . . . . .	9
2.1.3	Co-located Collaborative Workspace (CCW) . . . . .	11
2.1.4	Ambient Labor der HAW Hamburg . . . . .	13
2.2	Skalierbare Visualisierungssysteme . . . . .	15
2.3	Serious Games . . . . .	17
2.3.1	Ausprägungen und Kategorien von Serious Games . . . . .	18
2.3.2	Forschungsschwerpunkte . . . . .	19
2.3.3	Einordnung in das Forschungsgebiet CSCW . . . . .	20
<b>3</b>	<b>Analyse</b>	<b>21</b>
3.1	Serious Games für Teamtrainings . . . . .	22
3.1.1	Spielbasierte Simulationen (Game) . . . . .	23
3.1.2	Prozesse und Modelle aus der Lerntheorie (Pedagogy) . . . . .	25
3.1.3	Theorien zur Mediennutzung bei der Gruppenarbeit (Reality) . . . . .	29
3.1.4	Fazit . . . . .	33
3.2	Effizienter Einsatz von hochauflösenden Powerwalls . . . . .	35
3.2.1	Klassifikation von Nutzungsformen . . . . .	37
3.2.2	Kognitive Vorteile von großen Anzeigemedien . . . . .	39
3.2.3	Interaktionskonzepte . . . . .	41
3.2.4	Fazit . . . . .	46
3.3	Strategien und Verfahren zur skalierbaren Visualisierung . . . . .	47
3.3.1	Strategien zur Synchronisation von Daten . . . . .	48
3.3.2	Verfahren für die Eingabesynchronisation . . . . .	50
3.3.3	Verfahren für die Szenensynchronisation . . . . .	53
3.3.4	Realisierungsansätze: Invasiv vs. Nicht-Invasiv . . . . .	55
3.4	Auswertung und Zusammenführung . . . . .	56
3.4.1	Beispielszenario für Gruppentrainings . . . . .	56
3.4.2	Funktionale Anforderungen . . . . .	60

---

3.4.3	Nicht-funktionale Anforderungen . . . . .	67
<b>4</b>	<b>Design und Realisierung</b>	<b>68</b>
4.1	Konzeptionsansätze für den Trainingsraum . . . . .	68
4.1.1	CCW als Co-located Collaborative Training Environment (CCTE) . . . . .	68
4.1.2	Globale und lokale Sichten im CCTE . . . . .	70
4.1.3	Aufbau der Infrastruktur des CCTE . . . . .	71
4.2	Grundlagen der Systemarchitektur . . . . .	72
4.2.1	MVC-Entwurfsmuster in der Spieleentwicklung . . . . .	73
4.2.2	Client-Server-Modell in der Spieleentwicklung . . . . .	74
4.3	Systemarchitektur . . . . .	77
4.3.1	Systemübersicht . . . . .	77
4.3.2	Basissystem für spielbasierte Simulationen im CCTE . . . . .	79
4.3.3	Komponente Spielsimulation (Model) . . . . .	84
4.3.4	Komponente Spielvisualisierung (View) . . . . .	89
4.3.5	Komponente Spielmechanik (Control) . . . . .	95
4.4	Entwurf eines kollaborativen Spielkonzepts . . . . .	97
4.4.1	Beschreibung des Spielkonzepts . . . . .	97
4.4.2	Berücksichtigung der Designimplikationen . . . . .	99
4.5	Realisierung . . . . .	101
4.5.1	Allgemeines . . . . .	101
4.5.2	Skalierbare Bildschirmwand (Powerwall) . . . . .	103
4.5.3	Basisdienste . . . . .	105
4.5.4	Szenariodaten . . . . .	105
4.5.5	Szenarioverwaltung . . . . .	107
4.5.6	Spielkonfiguration . . . . .	108
4.5.7	Spielbasierte Simulation . . . . .	110
<b>5</b>	<b>Evaluation</b>	<b>113</b>
5.1	Untersuchungsschwerpunkte . . . . .	113
5.2	Aufbau und Ablauf der Untersuchung . . . . .	114
5.3	Ergebnisse der Untersuchung . . . . .	114
5.3.1	Grundlegende Funktionsweise . . . . .	115
5.3.2	Lerneffektivität . . . . .	115
5.3.3	Aufbau der Infrastruktur . . . . .	118
5.3.4	Verwendung der Powerwall . . . . .	119
5.3.5	Weitere Ergebnisse . . . . .	121
<b>6</b>	<b>Resümee</b>	<b>122</b>
6.1	Zusammenfassung . . . . .	122
6.2	Bewertung . . . . .	123
6.3	Ausblick . . . . .	124
6.3.1	Technische und konzeptionelle Weiterentwicklung des CCTE . . . . .	124
6.3.2	Übertragung auf andere Anwendungsbereiche im Bereich CSCW . . . . .	126

---

6.3.3 Die Zukunft von ernsthaften Spielen . . . . .	127
<b>A Anhang</b>	<b>129</b>
A.1 Technische Komponenten der Powerwall . . . . .	129
A.1.1 Displays . . . . .	129
A.1.2 Rechner . . . . .	129
A.2 Inhalt der CD-ROM . . . . .	130
<b>Abbildungsverzeichnis</b>	<b>131</b>
<b>Literaturverzeichnis</b>	<b>134</b>

# 1 Einleitung

Der Begriff *Homo Ludens*<sup>1</sup> ist durch [Huizinga \(1956\)](#) in seiner gleichnamigen Studie bekannt geworden. In seinen Forschungsarbeiten veranschaulichte er, dass das Spielen ein wichtiger und unverzichtbarer Bestandteil im Leben der Menschen ist. Kulturelle Systeme wie Politik, Wissenschaft, Religion oder Recht sind demnach ursprünglich aus spielerischen Verhaltensweisen hervorgegangen, die sich über die Zeit institutionell verfestigt haben. [Huizinga \(1956\)](#) stellte weiterhin fest, dass das Spielen für Menschen eine wichtige Methode sei, um neue und stabile Gruppen zu bilden. Nach [Berne \(1967\)](#) begründet sich das unter anderem darin, dass sich menschliche Beziehungen im Kern auf spielerischen Verhaltensweisen aufbauen.

Der *Akt des Spielens* ist demnach sowohl bei der individuellen Entwicklung von Menschen als auch bei Gruppen- und Sozialisationsprozessen ein sehr wichtiger Faktor. Verschiedene Institutionen und Unternehmen aus Praxis und Forschung sind sich dieser Zusammenhänge bewusst und setzen bei der Ausbildung verstärkt auf spielerische Ansätze. Bei vielen Trainingsmaßnahmen zur Aus- bzw. Fortbildung von Individuen und Gruppen werden daher unter anderem Computerspiele eingesetzt.

Begünstigt wurde dieser Trend nicht nur durch die sozialwissenschaftliche Relevanz von Spielen (für Menschen und deren Beziehungen), sondern vor allem auch durch die technologischen Fortschritte in der Unterhaltungssoftwareindustrie. Nach [Sawyer \(2002\)](#) konnte sich diese Branche in den letzten Jahren insbesondere durch sehr leistungsfähige Entwicklungen in den Bereichen der künstlichen Intelligenz, Visualisierung und Simulationsmodellierung auszeichnen. Die Verwendung von Computerspielen bzw. den zugrundeliegenden Methoden und Technologien für „ernsthafte“ Anwendungsgebiete wird in dem interdisziplinären Forschungsgebiet *Serious Games* betrachtet.

## 1.1 Motivation

In Bezug auf kollaborative Serious Games, bei denen die Ausbildung von Gruppen im Vordergrund steht, beschäftigt sich die Forschung aus der Sicht der Informatik unter anderem mit der Analyse von sogenannten *Massively Multiplayer Online Games* (MMOG). In diesen können sehr viele Benutzer mit sogenannten Avataren<sup>2</sup> in einer persistenten virtuellen 3D-Welt

---

<sup>1</sup>lat.: der spielende Mensch

<sup>2</sup>Ein Avatar ist eine künstliche Person oder ein grafischer Stellvertreter einer echten Person in einem Computerspiel.

gleichzeitig miteinander kommunizieren und interagieren. Mit den zugrundeliegenden Technologien und Interaktionsparadigmen werden spielbasierte Modelle für verteilte virtuelle Lern- und Trainingsumgebungen für Gruppen entwickelt, wie z.B. *Active Worlds Educational Universe* (AWEDU, 2007).

Der Vorteil dieser Umgebungen ist, dass damit sehr viele räumlich verteilte Gruppenmitglieder gleichzeitig erreicht und ausgebildet werden können. Ein Nachteil bei der Ausbildung von solchen *virtuellen Gruppen* ist die fehlende Face-to-Face-Kommunikation<sup>3</sup> und damit die Sichtbarkeit des Kommunikationsverhaltens der Partner. Dies spielt nach [Walser und Graf \(2004\)](#) eine wichtige Rolle bei der Entwicklung von Beziehungen und ist, neben den reinen Sachinhalten, unverzichtbar für das Zustandekommen von Verständnis, Motivation und Produktivität. Ein weiteres Problem ist die Gewährleistung bzw. die Aufrechterhaltung eines *Gruppengewahrseins*, dass zur Orientierung im Gruppenprozess sowie zur Einordnung und Planung der Gruppenaktivitäten benötigt wird ([Schlichter u. a., 2001](#)). Damit eignen sich solche Systeme zwar für die Ausbildung von Individuen einer Gruppe, das gezielte Training von Kooperation und Kollaboration zwischen den einzelnen Gruppenmitgliedern ist jedoch nur schwer umzusetzen.

Für Anwendungsbereiche und Trainingszenarien, die ganz gezielt auf die Förderung der Zusammenarbeit von Gruppen abzielen, ist es daher günstiger und teilweise sogar erforderlich, dass sich die Gruppenmitglieder während der Ausbildung *zur gleichen Zeit am gleichen Ort*<sup>4</sup> befinden. So ist der Einsatz von kollaborativen Serious Games z.B. für solche Anwendungsbereiche interessant, in denen Gruppen die Planung und Durchführung von gemeinsamen Bewegungsabläufen und Vorgehensweisen in unbekanntem und/oder teilweise komplexen räumlichen Umgebungen einüben sollen. Typische Beispiele für solche Anwendungsbereiche finden sich unter anderem bei militärischen Operationen, Sicherheitstrainings in Unternehmen oder Rettungseinsätzen von Polizei und Feuerwehr.

Die Bereitstellung von Systemumgebungen für solche *co-located Teamtrainings* auf der Basis von Serious Games findet derzeit in der Forschung nur wenig Beachtung. In der Praxis werden die Gruppenmitglieder in solchen Fällen daher zwar oftmals in einen gemeinsamen realen Raum zusammengeführt, dabei wird jedoch auf die gleichen Prinzipien wie bei verteilten virtuellen Lern- und Trainingsumgebungen gesetzt: die Spieler agieren autonom an ihren jeweiligen Arbeitsplätzen und interagieren ausschließlich in der virtuellen 3D-Welt miteinander. Aus der physischen Co-Präsenz und der realen Umgebung der Gruppenmitglieder wird kein Nutzen gezogen. Diesbezüglich bieten sich jedoch Möglichkeiten an, die Ausbildung von Gruppen nicht nur softwareseitig, d.h. durch Serious Games, sondern auch durch die Bereitstellung einer geeigneten technischen Infrastruktur zu unterstützen.

Mögliche Ansätze dafür finden sich in dem Forschungsgebiet *Computer Supported Collaborative Work*<sup>5</sup> (CSCW), in das sich die Forschungen zu kollaborativen Serious Games als Unterbereich einordnen lassen. Eines der Ziele im Bereich CSCW ist die Entwicklung von intelligenten

---

<sup>3</sup>zu dt.: von Angesicht zu Angesicht; Die Gruppenmitglieder befinden sich zur gleichen Zeit am gleichen Ort und stehen in unmittelbarer persönlicher Kommunikation zueinander

<sup>4</sup>Im weiteren Verlauf wird diese spezifische Raum/Zeit-Konstellation auch als *co-located* bezeichnet

<sup>5</sup>zu dt.: computergestützte Gruppenarbeit

Umgebungen bzw. Räumen, in denen durch den Einsatz von Informations- und Kommunikationstechnologie die Effektivität und Effizienz von Gruppenarbeit unterstützt werden soll. Diese sogenannten Co-located Collaborative Workspaces (CCWs) bestehen nach Vorbildern wie dem *iRoom* (Johanson u. a., 2004) oder *i-Land* (Fraunhofer-IPSI, 2002) aus einer heterogenen Landschaft verschiedener Systemkomponenten, wie z.B. Computern, Netzwerkgeräten und diversen Ein- und Ausgabemedien.

Eines der Konzepte zur Unterstützung der Gruppenprozesse in CCWs ist u.a. die Verwendung von großen und hochauflösenden Visualisierungssystemen, die als gemeinsame Arbeitsbereiche genutzt werden können. In diesen sollen Benutzer *gemeinsam und gleichzeitig* z.B. mit Dokumenten und Anwendungen arbeiten können. Handelsüblich erhältliche Anzeigemedien sind nur bis zu einer sehr eingeschränkten Größe bzw. Auflösung erhältlich und damit nur bedingt für solche Arbeitsbereiche geeignet. Eine Möglichkeit diese Einschränkung zu umgehen, ohne spezielle und kostspielige Hardware einsetzen zu müssen, ist die Verteilung der Berechnung und Darstellung von Grafiken auf mehrere Rechner und Monitore. Damit lassen sich Visualisierungssysteme realisieren, die theoretisch beliebig skalierbar sind.

## 1.2 Zielsetzung

Im Zusammenhang mit kollaborativen Serious Games stellt sich hier die Frage, inwieweit sich solche Visualisierungssysteme und möglicherweise auch andere Konzepte eines CCW für *spielbasierte co-located Teamtrainings* nutzen lassen, um die Zusammenarbeit der Teammitglieder zu fördern und damit möglicherweise den Lerneffekt bzw. die Qualität der Ausbildung durch „ernsthafte“ Computerspiele zu erhöhen.

In theoretischen Teil wird diesbezüglich analysiert, welche Aspekte bei der Entwicklung eines kollaborativen Serious Game grundsätzlich zu beachten sind, welche Vorteile hochauflösende Visualisierungssysteme mit sich bringen und wie sich solche Systeme realisieren lassen.

Das praktische Ziel dieser Arbeit ist die Entwicklung einer spielbasierten Lern- und Trainingsumgebung für Gruppen, die sich im funktionalen und infrastrukturellen Aufbau an den grundlegenden Eigenschaften und Konzepten eines CCW orientiert, sowie eines exemplarischen Serious Game, auf dessen Basis die Funktionalität dieser Umgebung veranschaulicht und evaluiert werden soll.

## 1.3 Aufbau der Arbeit

Im Anschluss an die Einleitung werden in Kapitel 2 alle Grundlagen erörtert, die für ein Verständnis der weiteren Arbeit notwendig sind. Dazu wird zunächst in Unterkapitel 2.1 das Forschungsgebiet der Computergestützten Gruppenarbeit (CSCW) vorgestellt. Anschliessend wird in Unterkapitel 2.2 skizziert, was unter skalierbaren Visualisierungssystemen zu verstehen ist

und wo diese eingesetzt werden. Das Kapitel Grundlagen wird mit einem Überblick der Forschungen und Projekte im Bereich Serious Games (Unterkapitel 2.3) abgeschlossen.

In Kapitel 3 erfolgt die Analyse für das zu konzipierende System in dieser Arbeit. Zunächst wird in Unterkapitel 3.1 untersucht, wie sich Serious Games für das Training von Gruppen einsetzen lassen. Das Ziel dieser Untersuchung wird es sein, mögliche Designimplikationen abzuleiten, die in der Konzeption des Systems zu berücksichtigen sind. Im darauffolgenden Unterkapitel 3.2 wird analysiert, wie sich hochauflösende Anzeigemedien effizient einsetzen lassen. Der Fokus dieser Betrachtung wird dabei auf der Verwendung von sogenannten *Powerwalls* liegen. Anschließend beschäftigt sich das Unterkapitel 3.3 mit der Frage, welche Methoden zur skalierbaren Visualisierung von Anwendungen eingesetzt werden. Hier werden verschiedene Strategien und Verfahren vorgestellt und hinsichtlich einer möglichen Verwendung in der Systemkonzeption bewertet. Zum Abschluss werden die in der Analyse gewonnenen Erkenntnisse ausgewertet und in ein Beispielszenario für Gruppentrainings in einer spielbasierten Lernumgebung zusammengeführt (Unterkapitel 3.4). Darauf basierend erfolgt eine Analyse der funktionalen und nicht-funktionalen Anforderungen, die das zu konzipierende System leisten muss.

Das Kapitel 4 beschäftigt sich mit dem Design und der Realisierung des Systems dieser Arbeit. Zu Beginn werden in Unterkapitel 4.1 einige Ansätze für die Konzeption eines spielbasierten Trainingsraums vorgestellt. Daran anschließend werden in Unterkapitel 4.2 einige wichtige Grundlagen der Systemarchitektur besprochen, bevor in Unterkapitel 4.3 der eigentliche Systementwurf dieser Arbeit erläutert wird. Im Unterkapitel 4.4 wird ein Spielkonzept für ein kollaboratives Serious Game vorgestellt, anhand dessen die Funktionalität des Systems aufgezeigt werden kann. Abschließend werden in Unterkapitel 4.5 die wesentlichen Aspekte der Realisierung des spielebasierten Lernumgebung und des exemplarischen Serious Game besprochen.

Eine erste Evaluation des Systems wird in Kapitel 5 diskutiert. Im Rahmen der dortigen Ausführungen werden die Schwerpunkte ((Unterkapitel 5.1), Aufbau und Ablauf (Unterkapitel 5.2) sowie die Ergebnisse (Unterkapitel 5.3) der Untersuchung besprochen.

Den Abschluss dieser Arbeit bildet das Resümee in Kapitel 6. Hier werden die Ergebnisse der Arbeit zusammengefasst und bewertet (Unterkapitel 6.1 und 6.2). Ferner wird ein Ausblick auf mögliche Verbesserungen und weiterführende Arbeiten in diesem Gebiet gegeben (Unterkapitel 6.3).

## 2 Grundlagen

In diesem Kapitel werden verschiedene Themen diskutiert, die für ein umfassendes Verständnis dieser Arbeit erforderlich sind. Zunächst wird in Unterkapitel 2.1 das Forschungsgebiet der computergestützten Gruppenarbeit (CSCW) vorgestellt. Im Anschluss daran wird gezeigt, was unter skalierbaren Visualisierungssystemen zu verstehen ist und wo diese eingesetzt werden (Unterkapitel 2.2). Abschliessend werden in Unterkapitel 2.3 die Forschungsarbeiten im Bereich Serious Games aufgezeigt. Dabei werden verschiedene Ausprägungen und Kategorien von Serious Games vorgestellt, die Forschungsschwerpunkte besprochen und der Bezug zu den Arbeiten im Bereich CSCW hergestellt.

### 2.1 Computergestützte Gruppenarbeit (CSCW)

Die Vernetzung von Arbeitsplatzrechnern wird mittlerweile nicht nur für eine verteilte Datenverarbeitung, sondern auch zur Unterstützung kollaborativen Arbeitens genutzt ([Schlichter u. a., 2001](#)). Das Forschungsgebiet, das sich mit der Nutzung von verteilten Systemen für eine solche computergestützte Gruppenarbeit beschäftigt, wird unter dem Begriff *Computer Supported Cooperative Work* (CSCW) zusammengefasst. Die Entstehung des Begriffs CSCW ist auf Irene Greif (MIT) und Paul Cashman (DEC) zurückzuführen, die damit das gemeinsame Forschungsfeld einer Gruppe von Workshop-Teilnehmern aus sehr verschiedenen Arbeitsbereichen <sup>1</sup> ausgedrückt haben ([Brown u. a., 1985](#)).

Nach [Hasenkamp u. a. \(1994\)](#) lassen sich bei CSCW drei eng miteinander zusammenhängende Forschungsgebiete unterscheiden:

- die Entwicklung eines Verständnisses der Zusammenarbeit und Koordination
- die Entwicklung von Konzepten und Werkzeugen für die Unterstützung arbeitsteiliger Prozesse
- die Bewertung dieser Konzepte und Werkzeuge

Die Forschungen in diesem Bereich gestalten sich dabei aus interdisziplinären Sichtweisen: so spielen neben informatischen Gesichtspunkten, beispielsweise auch Fragestellungen unter soziologischen, psychologischen und wirtschaftswissenschaftlichen Aspekten eine wichtige Rolle. Im Allgemeinen bezeichnet CSCW dabei „[...] die theoretischen Grundlagen bzw. die

---

<sup>1</sup>Ein CSCW Workshop, der 1984 im Endicott House (Massachusetts) stattfand

*Methodologien für Gruppenarbeit und deren Unterstützung durch Rechner*“ (Schlichter u. a., 2001). Eine konkretere Definition für CSCW liefert Wilson (1991):

*„CSCW is a generic term which combines the understanding of the way people work in groups with the enabling technologies of computer networking, and associated hardware, software, services and techniques.“*

Das Ziel sämtlicher Arbeiten im Bereich CSCW ist es demnach, unter Nutzung aller zur Verfügung stehenden Mittel der Informations- und Kommunikationstechnologie, Gruppenprozesse zu untersuchen und dabei die Effektivität und Effizienz der Gruppenarbeit zu erhöhen.

### 2.1.1 Begriffsabgrenzungen

Im Zusammenhang mit dem Forschungsgebiet CSCW werden verschiedene weitere Begriffe in Verbindung gebracht: *Groupware*, *Collaborative Workspace*, *Co-located Collaborative Workspace* und *Collaborative Virtual Environment*.

Die Entstehung des Begriffs **Groupware** (auch als CSCW-System bezeichnet) ist auf Johnson-Lenz und Johnson-Lenz (1981) zurückzuführen. Darunter versteht man Systeme zur Unterstützung von Gruppenarbeit, in deren Realisierung die theoretischen Grundlagen integriert sind, die im Rahmen der Forschungen im Bereich CSCW spezifiziert wurden. Nach Ellis u. a. (1991) handelt es sich bei solchen CSCW-Systemen um rechnerbasierte Systeme, die eine Gruppe von Individuen bei ihren gemeinsamen Aufgaben unterstützen und eine Schnittstelle zu einer gemeinsamen Umgebung bereitstellen. Johansen (1988) beschreibt Groupware wie folgt:

*„Groupware is a generic term for specialised computer aids that are designed for the work of collaborative work groups. Typically, these groups are small project-oriented teams that have important tasks and tight deadlines. Groupware can involve software, hardware, services and/or group process support.“*

Groupware ist demnach eine generische Bezeichnung, unter der sich eine Vielzahl von Systemen mit unterschiedlichster Funktionalität und Komplexität subsummieren lassen. So reicht das Spektrum vom einfachen E-Mail-Client, über Foren und Wikis bis hin zu kompletten Workflow-Management-Systemen.

**Collaborative Workspaces** (CWs) lassen sich im weitesten Sinne zwar auch unter Groupware einordnen, werden dabei aber deutlich konkreter: ein CW bietet eine vollständige computergestützte Arbeitsumgebung, die für projektorientierte Zusammenarbeit mehrerer Personen konzipiert ist. (Roßberger, 2008).

Bei **Collaborative Virtual Environments** (CVEs) handelt es sich um spezielle Ausprägung von CWs. Hier kommunizieren und kooperieren die Gruppenmitglieder in einem *virtuellen Arbeitsraum*, ohne sich dabei notwendigerweise (physisch) am gleichen Ort befinden zu müssen.

Demgegenüber existiert mit den **Co-located Collaborative Workspaces** (CCWs) eine weitere Sonderform von CWs. Das wichtigste Kriterium eines CCW ist, dass sich die Gruppenmitglieder (physisch) zur gleichen Zeit und am gleichen Ort befinden. Diese Form des Collaborative

Workspace spielt für diese Arbeit eine wesentliche Rolle und wird daher in Abschnitt 2.1.3 noch genauer betrachtet.

## 2.1.2 Klassifizierung von CSCW

CSCW-Systeme können in ihren Eigenschaften und ihrem Funktionsumfang höchst unterschiedlich ausfallen. Nach Schlichter u. a. (2001) können solche Systeme jedoch nach verschiedenen Kriterien charakterisiert und klassifiziert werden:

- Klassifizierung nach Raum und Zeit
- Klassifizierung nach dem 3K-Modell
- Klassifizierung nach funktionalen Anwendungsklassen

Die einzelnen Klassifizierungen werden im Folgenden kurz vorgestellt.

### Klassifizierung nach Raum und Zeit

Diese Klassifizierung wurde von Johansen (1988) eingeführt. Hierbei werden CSCW-Systeme und deren Nutzung durch Personengruppen in den zwei Dimensionen *Raum* und *Zeit* unterschieden. Mit dieser Charakterisierung lässt sich beschreiben, ob sich die Mitglieder einer Gruppe am gleichen Ort oder an verschiedenen Orten befinden, und ob zur gleichen Zeit (synchron) oder zu verschiedenen Zeitpunkten (asynchron) interagiert wird.

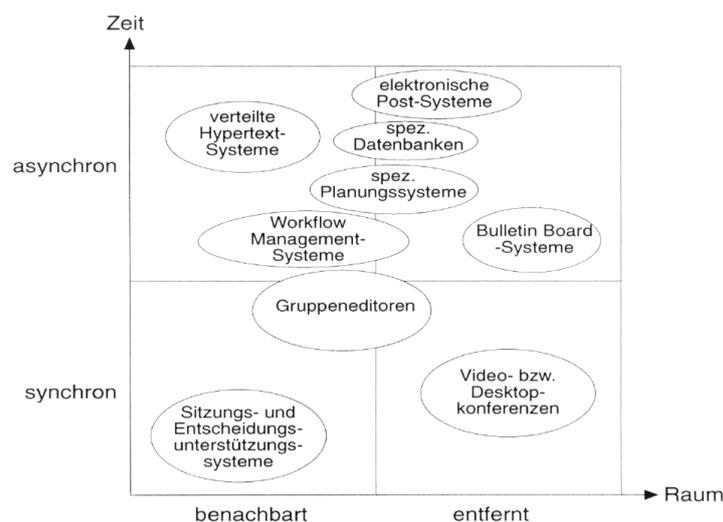


Abb. 2.1: Raum-Zeit-Matrix (Teufel u. a., 1995)

Die Kombination dieser beiden Dimensionen führt grundsätzlich zu vier verschiedenen Situationstypen. Die Darstellung dieser Klassifizierung und die Einordnung von CSCW-Systemen erfolgt üblicherweise in einer Raum-Zeit-Matrix (siehe Abb. 2.1).

Nach [Schlichter u. a. \(2001\)](#) sind die vier Quadranten der Matrix nicht im Sinne einer Eingrenzung und/oder Abgrenzung zu betrachten. Vielmehr sind die Grenzen der einzelnen Kategorien als *fließende Übergänge* zu verstehen. Ein umfassendes CSCW-System muss daher den Anforderungen aller vier Quadranten genügen.

Die Raum-Zeit-Matrix wurde durch [Grudin \(1994\)](#) um eine Koordinationskomponente erweitert. Damit lässt sich die räumliche und zeitliche Verteilung bezüglich der *Vorhersehbarkeit verschiedener Orte/Zeiten* differenzierter betrachten. In der Kategorie „verschieden“ wird daher zusätzlich nach „verschieden, aber vorhersehbar“ und nach „verschieden und nicht vorhersehbar“ unterschieden.

### Klassifizierung nach dem 3K-Modell

[Teufel u. a. \(1995\)](#) klassifizieren CSCW-Systeme nach dem Grad ihrer Unterstützung bezüglich der grundlegenden Eckpfeiler der Gruppenarbeit. Als Basis dafür wird bei der Zusammenarbeit innerhalb einer Gruppe zwischen *Kommunikation*, *Koordination* und *Kooperation* unterschieden. Bei der Kommunikation steht der Informationsaustausch zwischen Personen im Vordergrund. Die Koordination dient zur Planung, Steuerung und Überwachung der Gruppenressourcen, während die Kooperation die Verfolgung gemeinsamer Ziele fordert.

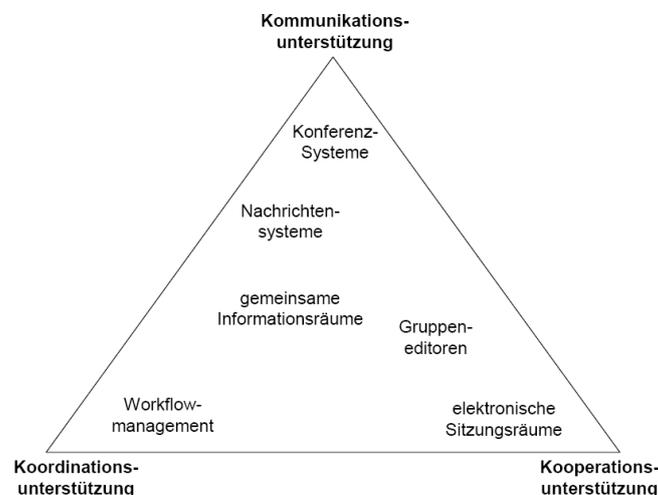


Abb. 2.2: 3K-Klassifizierung nach [Teufel u. a. \(1995\)](#)

Mit dem 3K-Modell lassen sich auf diese Weise verschiedene CSCW-Systeme nach ihren jeweiligen Unterstützungsfunktionen in einem Dreieck einordnen (siehe [Abb. 2.2](#)).

### Klassifizierung nach Anwendungsklassen

In [Abbildung 2.2](#) wurden verschiedene Anwendungstypen eingeordnet. Sie entstammen einer Klassifizierung nach anwendungsorientierten Funktionsklassen. Nach [Ellis u. a. \(1991\)](#) lassen

sich beispielsweise folgende Anwendungstypen unterscheiden:

**Nachrichtensysteme** (z.B. E-Mail-Systeme) unterstützen den asynchronen Austausch von Textnachrichten und multimedialen Informationen (z.B. Bilder, Ton oder Video) zwischen Gruppenmitgliedern.

**Gruppeneditoren** ermöglichen das gemeinsame Bearbeiten von Dokumenten. Die Benutzer werden dabei nicht voneinander isoliert, sondern gegenseitig über die stattfindenden Aktivitäten informiert.

**Elektronische Sitzungsräume** unterstützen den Diskussionsverlauf und die Gruppendynamik klassischer Sitzungsräume für Face-to-Face-Sitzungen. Dies erfolgt durch den Einsatz rechnergestützter Werkzeuge für die Entscheidungsfindung und zur Erforschung von unstrukturierten Problemen.

**Konferenzsysteme** werden nach *Nicht-Realzeitkonferenzen* und *Realzeitkonferenzen* unterschieden. Bei Nicht-Realzeitkonferenzen kommunizieren die Gruppenteilnehmer asynchron über ihren Arbeitsplatzrechner (z.B. über E-Mails oder Newsgroups). Bei Realzeitkonferenzen sind alle Sitzungsmitglieder zur selben Zeit anwesend und alle Beiträge werden direkt übermittelt.

**Gemeinsame Informationsräume** stellen die systemorganisierte Verwaltung von Gruppendokumenten in den Vordergrund. Sie erlauben persistente Speicherung der Daten und stellen geeignete Zugriffsmechanismen zur Verfügung.

**Koordinationsysteme** werden auch als *Workflow-Management-Systeme* bezeichnet. Sie werden zur Koordinierung der verschiedenen (meist asynchronen) Tätigkeiten einzelner Gruppenteilnehmer eingesetzt, die zur Vollendung des gemeinsamen Ziels notwendig sind.

Die Klassifizierung nach Anwendungsklassen ist ebenfalls nicht als statisch zu begreifen. Auch hier gibt es fließende Übergänge und Überschneidungen zwischen den verschiedenen Anwendungstypen. Des Weiteren gibt es auch Variationen in der Anzahl, den Bezeichnungen und den Beschreibungen der einzelnen Anwendungsklassen (z.B. [Teufel u. a. \(1995\)](#)).

### 2.1.3 Co-located Collaborative Workspace (CCW)

Wie bereits in Abschnitt 2.1.1 erwähnt wurde, handelt es sich bei Co-located Collaborative Workspaces (CCWs) um eine Spezialisierung von Collaborative Workspaces (CWs). CCWs bieten eine computergestützte Arbeitsumgebung für eine projektorientierte Zusammenarbeit mehrerer Personen, die gemeinsam und gleichzeitig in einem speziell dafür ausgestatteten Besprechungs- oder Arbeitsraum arbeiten. Gemäß der in Abschnitt 2.1.2 vorgestellten Klassifizierung von CSCW-Systemen in einer Raum-Zeit-Matrix beschränken sich CCWs damit auf den ersten Quadranten „*gleicher Ort und gleiche Zeit*“ (siehe Abb. 2.1).

Als Vorbild für einen CCW werden typischerweise konventionelle Besprechungs- oder Versammlungsräume genommen, wie man sie insbesondere aus Unternehmen kennt. Der konkrete Anwendungskontext solcher Räume kann je nach Bedarf höchst unterschiedlich ausfallen: kurze Projektbesprechungen, kreative Konzeptionsprozesse (z.B. Brainstormings), Vertragsabschlüsse oder Fortbildungsmaßnahmen sind hier exemplarisch zu nennen. Eine Gemeinsamkeit aller Nutzungsformen ist jedoch, dass in der Regel ein Informationsfluss in Form von Dokumenten zwischen den Gruppenmitgliedern in dem Versammlungsraum stattfindet. Dies impliziert, dass Dokumente vorhanden sind (z.B. Vortragsfolien), für die Gruppe sichtbar gemacht werden können (z.B. durch eine Beamerprojektion der Vortragsfolien) und bei Bedarf an die Gruppe verteilt werden können (z.B. als Handouts oder als digitale Kopien).



Abb. 2.3: i-LAND: Oben die erste Konzeption und einzelne Roomware-Komponenten, darunter ein mögliches Einsatzszenario (Fraunhofer-IPSI, 2002)

Ein CCW optimiert die Gruppenarbeit und den Informationsfluss zwischen den Gruppenmitgliedern, indem eine komplette Infrastruktur für die Verwaltung, Visualisierung und Verteilung von digitalen Dokumenten bereitgestellt wird. Dafür ist eine solche kollaborative Arbeitsumgebung in der Regel mit großen und hochauflösenden Bildschirmen ausgestattet, die zum Teil über berührungsempfindliche Oberflächen zur Steuerung von Anwendungen und Dokumenten verfügen (Touchscreens). Darüber hinaus können mobile Geräte (z.B. PDAs oder Notebooks) der beteiligten Gruppenmitglieder direkt in die bestehende Infrastruktur eingebunden werden.

Die Idee, konventionelle Arbeits- oder Versammlungsräume mit Technologie anzureichern, ist auf die von Weiser (1991) formulierte Vision des *Ubiquitous Computing* zurückzuführen: Computertechnologie ist allgegenwärtig und unterstützt die Menschen bei ihren Aufgaben,

verschwindet aber selber in den Hintergrund. Im Bereich der Forschungen zur Gestaltung, Konzeption und Nutzung von CCWs eröffnet das Ubiquitous Computing ein weites Feld an Vertiefungsschwerpunkten.

So wurde z.B. bei dem vom Fraunhofer Institut entwickelten *i-LAND* (Streitz u. a., 1999) ein Schwerpunkt auf das Verschwinden von Technologie gelegt, indem Computertechnologien (z.B. Monitore) in Wände, Türen oder Möbel integriert wurden (sogenannte Roomware, siehe Abb. 2.3).

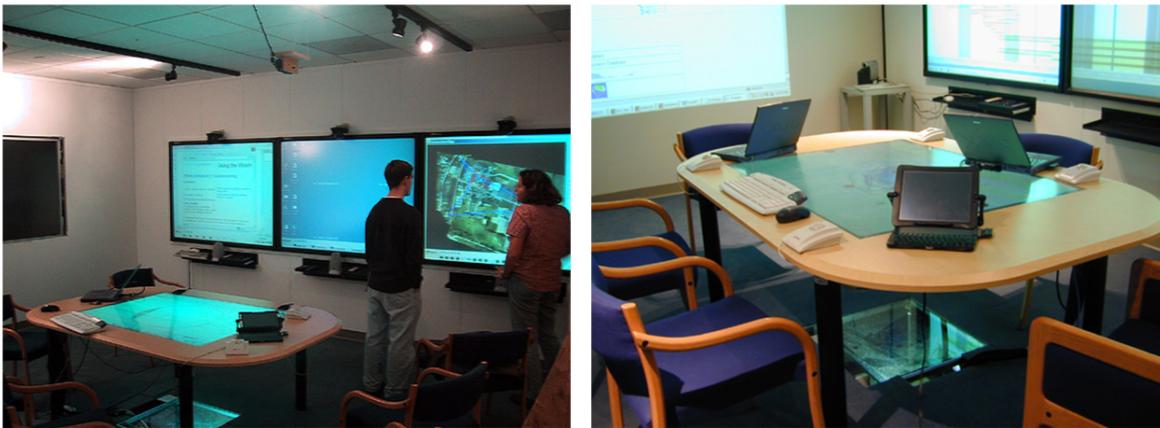


Abb. 2.4: iRoom der Stanford University (Johanson u. a., 2004)

Beim *iRoom* (Fox u. a., 2000) der Stanford University liegt der Fokus dagegen größtenteils auf der *Human-Computer-Interaction*<sup>2</sup> (HCI), insbesondere in Bezug auf Fragestellungen und Problemen zur Interaktion mit großen, hochauflösenden Bildschirmen (siehe Abb. 2.4).

Auch an der HAW Hamburg wird im Rahmen der Forschungen im Bereich Ubiquitous Computing im Ambient Labor die Konzeption und Realisierung von Co-located Collaborative Workspaces verfolgt. Im nachfolgenden Abschnitt werden das Labor sowie die dort bisher gelaufenen Arbeiten und Schwerpunkte kurz vorgestellt.

### 2.1.4 Ambient Labor der HAW Hamburg

Bereits seit 2002 werden an der HAW Hamburg im Fachbereich Informatik durch verschiedene Projekte sowie Seminar- und Abschlussarbeiten unterschiedlichste Aspekte und Probleme aus dem Bereich Ubiquitous Computing behandelt. Alle dabei entstandenen Beiträge und Ergebnisse werden in dem übergeordneten Projekt *UbiComp* zusammengetragen (HAW-Hamburg, 2002).

<sup>2</sup>zu dt.: Mensch-Maschine-Interaktion; Ein Teilgebiet der Informatik, das sich mit der benutzergerechten Gestaltung von interaktiven Systemen und ihren Mensch-Maschine-Schnittstellen beschäftigt.

Zunächst lag der Fokus der Arbeiten insbesondere auf Software-Architekturen für verteilte, ubiquitäre Systeme. Seit 2006 hat sich der Forschungsschwerpunkt zunehmend auf das Gebiet *Ambient Intelligence* verlagert, in dem die Entwicklung einer intelligenten Umgebung verfolgt wird. Darauf basierend ist das Ambient-Labor entstanden, in dem zur Zeit ein Co-located Colaborative Workspace konzipiert und realisiert wird.

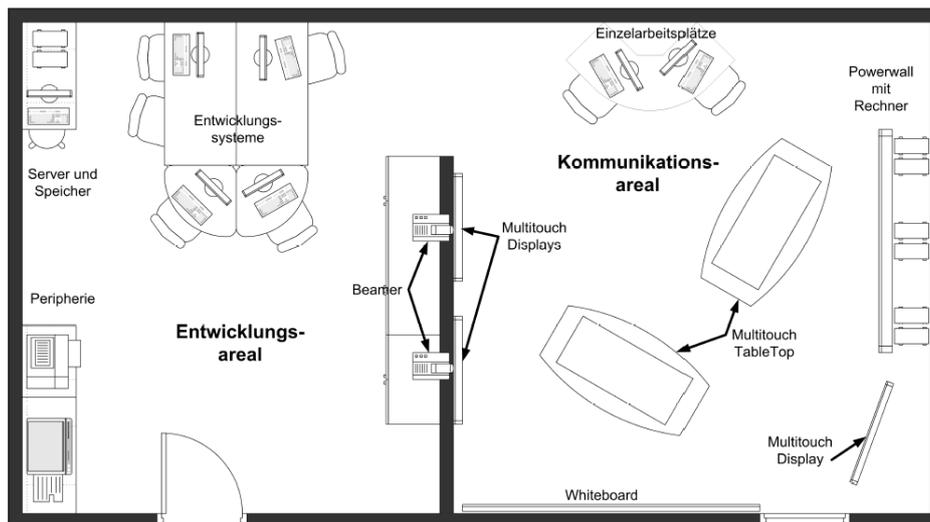


Abb. 2.5: Systeminfrastruktur des Ambient-Labors der HAW Hamburg (nach Fischer (2007))

Die Abbildung 2.5 zeigt eine schematische Übersicht der Infrastruktur des Ambient-Labors. Grundsätzlich ist hier eine Einteilung in zwei Bereiche vorgesehen: das Entwicklungsareal und das Kommunikationsareal. Während das Kommunikationsareal den eigentlichen CCW darstellt, werden im Entwicklungsareal die dafür notwendigen Technologien realisiert und gesteuert.

Wie schon zuvor in Abschnitt 2.1.3 dargelegt, lassen sich die Untersuchungen und Entwicklungen im CCW in verschiedene Richtungen lenken. So hat beispielsweise Fischer (2007) die multimodalen Interaktionsmöglichkeiten im CCW untersucht und dafür ein prototypisches Eingabe- und Zeigegerät entwickelt. Roßberger (2008) hat an das Konzept der *Seamless Interaction*<sup>3</sup> angeknüpft und darauf basierend ein physikbasiertes Interaktionssystem, insbesondere für Tabletops<sup>4</sup>.

Einer der weiteren Untersuchungsschwerpunkte im Ambient-Labor ist es, die gesamte Infrastruktur als *einen gemeinsamen Anwendungskontext* zur Verfügung zu stellen. Ein Ansatz dafür ist, die reale Infrastruktur mit einem virtuellen Systemmodell zu unterlegen, welches alle Geräte zu einer logischen Einheit verbindet (Kindberg und Fox, 2002). Insbesondere die Eingabe- und Ausgabemedien sollen damit „über die Gerätegrenzen hinaus“ benutzbar gemacht werden.

<sup>3</sup>zu dt.: nahtlose Interaktion; Damit wird der nahtlose Übergang von Interaktionen mit Objekten aus der realen und virtuellen Umgebung bezeichnet.

<sup>4</sup>Tabletops bezeichnen horizontale Interaktionsflächen, die in Tische eingelassen sind bzw. darauf projiziert werden.

Das Ziel ist hier z.B. die Bereitstellung eines verteilten 3D-Desktops (Köckritz, 2007), in dem Dokumente und Anwendungsfenster zwischen den zur Verfügung stehenden Anzeigeflächen verschoben werden können.

In dieser Arbeit wird an den zuvor erwähnten Ansatz angeknüpft. Der CCW mit seinen zahlreichen Anzeigeräten wird hier als *System mit skalierbarer Visualisierung* begriffen. Ein Teilaspekt bildet dabei die Entwicklung und Verwendung von hochauflösenden Bildschirmwänden (Powerwalls). Diese werden sowohl zur *Darstellung von* als auch zur *Interaktion mit* unterschiedlichsten Informationen genutzt. Im Kontext eines CCW fungieren diese Bildschirmwände aufgrund ihrer Größe häufig als gemeinsame Arbeitsbereiche (z.B. Johanson u. a. (2002)), um die Gruppenprozesse bei bestimmten Aufgaben zu unterstützen. Die Fragestellungen und Probleme, die sich beim Einsatz solcher Anzeigemedien ergeben, z.B. in Bezug auf mögliche Realisierungskonzepte, Interaktionsmöglichkeiten und der Einbindung in einen gemeinsamen Anwendungskontext, werden in Kapitel 3 erörtert.

## 2.2 Skalierbare Visualisierungssysteme

In vielen Anwendungsbereichen aus Industrie und Wissenschaft ist die dreidimensionale Visualisierung von Informationen notwendig, die einen sehr hohen Detaillierungsgrad erfordern. Beispiele dafür finden sich u.a. in der Konstruktion und Planung von großen Gebäudeanlagen (z.B. Fabriken) oder in der Stadtplanung. Das Echtzeit-Rendering<sup>5</sup> von solchen 3D-Daten

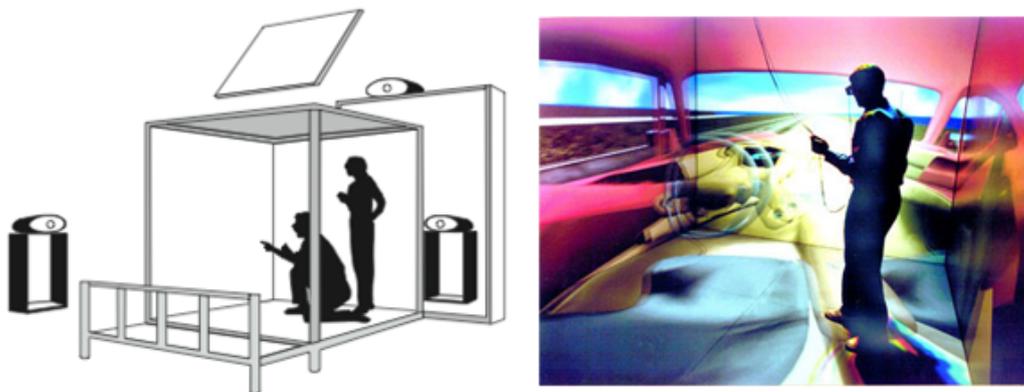


Abb. 2.6: Links der konzeptionelle Aufbau eines CAVE-Systems (Robert, 2006) und daneben eine Realisierung im 3D Lab der University Michigan (UM3D, 2008)

übersteigt dabei oftmals die maximale Anzahl von visuell darstellbaren Bildpunkten, die bei der *klassischen Visualisierung* mit einer Workstation zu erreichen sind. Um große Datenmengen dennoch realitätsgetreu und in Echtzeit visualisieren zu können, werden für diese Zwecke

<sup>5</sup>Echtzeit-Rendering bezeichnet in der 3D-Computergrafik die kontinuierliche Berechnung und unmittelbare Ausgabe eines Bildes auf der Basis einer dreidimensionalen Szenenbeschreibung.

Systeme mit skalierbarer Visualisierung eingesetzt, die oftmals auch als *Immersive Environments* bezeichnet werden. Diese bestehen in der Regel aus einem Cluster von Rechnern, in dem jeder Knoten (Rechner + Ausgabemedium) für die Visualisierung eines Teilbereichs der darzustellenden Bildpunkte zuständig ist. Auf diese Weise können mehrere Ausgabemedien zu einem gemeinsamen, hochauflösenden Visualisierungskontext zusammengeschlossen werden.

Die Bezeichnung „skalierbare Visualisierung“ ist bei solchen Systemem darauf zurückzuführen, dass neben geeigneter Hardware insbesondere effiziente Skalierungsstrategien zur Verteilung des Visualisierungskontextes benötigt werden. Nach [Robert \(2006\)](#) bezieht sich *skalierbare Visualisierung* auf:

- die parallele Ausführung von lokalen und/oder verteilten Anwendungen
- die Verwendung von mehreren Anzeigemedien als ein einheitliches Anzeigemedium
- die Verwendung von mehreren Anzeigemedien mit unterschiedlichen Perspektiven
- die effiziente Handhabung großer Datenmengen
- die Möglichkeit zu einer flexiblen Konfiguration der ausgeführten Anwendungen

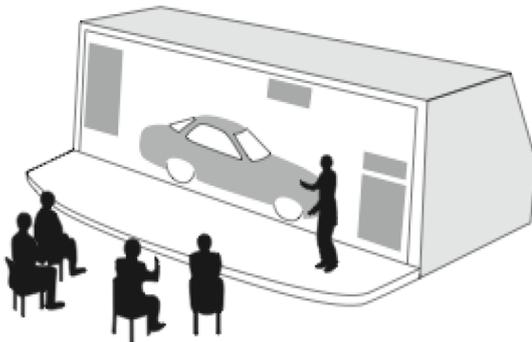


Abb. 2.7: Powerwall als skalierbares Visualisierungssystem (links von [Robert \(2006\)](#) und rechts von [Mechdyne \(2007b\)](#))

Mit den konkreten Anwendungsszenarien, in denen solche Visualisierungssysteme zum Einsatz kommen, unterscheiden sich auch die Art ihrer Installationen. Die sogenannten CAVE-Systeme (Abb. 2.6) sind insbesondere für Applikationen aus dem Bereich *Virtual Reality* konzipiert. Dabei handelt es sich um einen würfelförmigen Raum, bei dem die Wände als Projektionsfläche für die Visualisierung verwendet werden. Eine andere gängige Ausprägung von *Immersive Environments* sind großflächige Wände aus Bildschirmen oder Projektionsleinwänden, die auch als *Walls* oder *Powerwalls* bezeichnet werden (Abb. 2.7). Eine Kompromisslösung zwischen der zweidimensionalen Darstellungsfläche einer Powerwall und der den Benutzer umgebenden Darstellung in einem CAVE-System, sind die sogenannten Curved Screens (Abb. 2.8). Bei diesen handelt es sich im Wesentlichen um leicht gewölbte Powerwalls, die für den Betrachter damit einen „eindringlicheren“ visuellen Effekt erzeugen sollen.

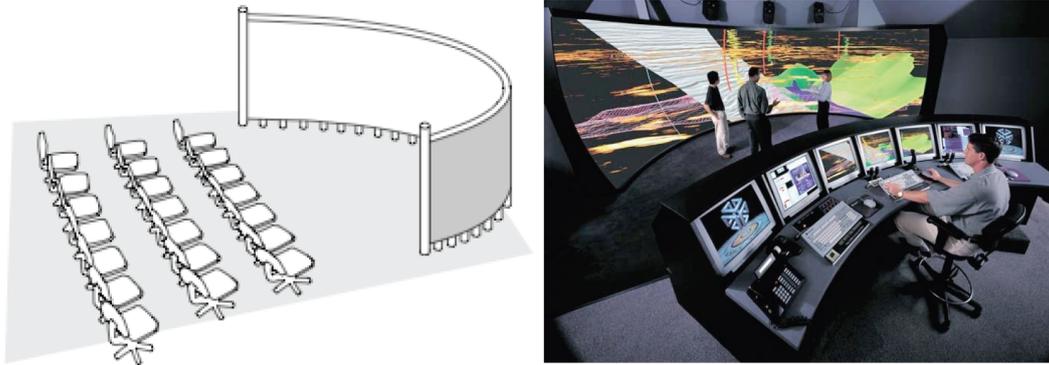


Abb. 2.8: Curved Screens bieten durch die Wölbung ähnliche Raumeffekte wie in CAVE-Systemen (Robert, 2006)

## 2.3 Serious Games

Computerspiele dienen heutzutage nicht mehr nur der alleinigen Unterhaltung. Spielesoftware und spielbasierte Simulationen werden bereits jetzt in vielen Industriezweigen und öffentlichen Einrichtungen eingesetzt, z.B. für Trainingsmaßnahmen in den Bereichen Managementwissenschaften, Ökonomie, Psychologie, Soziologie, interkultureller Kommunikation, Politikwissenschaften, Militärstrategien, Teamfähigkeit oder Bildung allgemein (Raybourn und Bos, 2005).

Motiviert wurde dieser Trend vor allem durch den technologischen Vorsprung, den sich die Spieleindustrie in den letzten Jahren insbesondere in den Punkten künstliche Intelligenz, Visualisierung und Simulationsmodellierung erarbeitet hat (Sawyer, 2002). Dabei wurden nach Stone (2005) ausgereifte Methoden und leistungsfähige Werkzeuge entwickelt, die die Erstellung neuer Spielmodelle unterstützen und dabei sehr kosteneffizient sind. Ein Beispiel dafür sind sogenannte 3D-Engines, die die Erstellung interaktiver 3D-Applikationen vereinfachen und dabei teilweise kostenlos verfügbar sind (z.B. OGRE3D (2007) oder Irrlicht (2008)).

Die Verwendung von diesen Methoden und Technologien aus dem Computerspielbereich für ernsthafte Inhalte wird unter dem Begriff *Serious Games* (SG)<sup>6</sup> betrachtet. *Serious Games* bezeichnet dabei sowohl den Forschungsbereich als auch die Produkte bzw. Projekte, die aus diesem hervorgehen. Die Forschungs- und Interessengemeinschaft für SG hat sich in den letzten Jahren insbesondere durch die Bestrebungen der *Serious Games Initiative* (SGIOrg, 2002) formiert, die 2002 am Woodrow Wilson International Center for Scholars (WWICS) ins Leben gerufen wurde. Der Begriff *Serious Games* wurde durch die beiden Begründer der SGI, Ben Sawyer (Digital Mills) und David Rejeski (WWICS), geprägt, um die Zusammenführung zweier Wirtschaftszweige zu beschreiben: der *klassischen Industrie* auf der einen Seite und die *Spielerbranche* auf der anderen.

Eine allgemeingültige Definition für *Serious Games* existiert nicht. Sawyer (2002) beschreibt *Serious Games* beispielsweise wie folgt:

<sup>6</sup>zu dt.: ernsthafte Spiele

„*Serious Games [...] are interactive applications that extend far beyond the traditional videogame market, including: training, policy exploration, analytics, visualisation, simulation, education, health and therapy.*“

Das Konzept von „Lernspielen“ ist nicht neu: schon seit den 1980er Jahren stellt die sogenannte *Entertainment-Education* (auch kurz als Edutainment bezeichnet) den Versuch dar, gesellschaftlich erwünschte Ergebnisse - wie z.B. Lernen - durch den Einsatz von Unterhaltungsmedien zu erzielen (Wong u. a., 2007). Während die Ergebnisse dieser Bestrebungen jedoch hauptsächlich für ein sehr junges Publikum konzipiert wurden, z.B. zur Förderung von Lesen und Schreiben, erweitern SG das Konzept auf eine größere Zielgruppe und ermöglichen damit den Einsatz für ein breit gefächertes Anwendungsfeld. So können Serious Games beispielsweise auch bei der Maschinensteuerung, Anlagenplanung, Sicherheitstrainings oder Marketingkampagnen zum Einsatz kommen, da es unter konzeptionellen und technologischen Aspekten eine Vielzahl von Parallelen gibt (Fraunhofer-IuK, 2007).

### 2.3.1 Ausprägungen und Kategorien von Serious Games

Durch die verschiedenen Einsatzmöglichkeiten von Serious Games, sowohl in Bezug auf die thematischen Inhalte als auch auf den erwünschten Nutzen, gestalten sich die aktuellen Projekte sehr vielfältig. Für eine inhaltliche und intentionale Klassifizierung von SG wurden von Alvarez u. a. (2007) daher folgende Kategorien identifiziert:

**Advergaming:** Damit bezeichnet man die Benutzung von Spielen für die Bewerbung von Produkten, Personen, Organisationen oder Standpunkten. Einige Beispiele dafür sind *The Coke Zero Game*<sup>7</sup> oder *Moorhuhn Jagd*<sup>8</sup>.

**Edutainment:** Wie bereits in Abschnitt 2.3 erwähnt wurde, handelt es sich hierbei um Lernspiele, die in erster Linie für Kinder und Jugendliche konzipiert werden, um Wissen und Informationen auf eine motivierende, spielerische Art und Weise zu vermitteln. Der Anspruch an die zu transferierenden Inhalte kann dabei von „grundlegend“ (z.B. Lesen, Schreiben, Rechnen) bis hin zu „wissenschaftlich komplex“ reichen, wie z.B. in *Ruhemasse Null*<sup>9</sup>, in dem die Spielenden an die Astronomie und Astrophysik herangeführt werden.

**Edumarket Games:** Spiele dieser Kategorie sind eine Symbiose aus Advergaming und Edutainment: sie erhöhen den Wert eines Produkts, einer Institution, eines Konzepts oder einer Ideologie und haben dabei gleichzeitig einen Bildungswert. Ein Vertreter dieser Kategorie ist beispielsweise *Food Force*<sup>10</sup> von dem *United Nations World Food Programme* (WFP). Spieler übernehmen dabei Missionen, in denen sie Hilfsgüterlieferungen durchführen und beim Wiederaufbau von Krisenregionen mitwirken. Gleichzeitig lernen sie mehr über die aktuelle Hungersnot und über die Arbeit des WFP in der realen Welt.

<sup>7</sup><http://www.cokezerogame.de>

<sup>8</sup><http://www.moorhuhn.de>

<sup>9</sup><http://www.ruhemasse-null.de>

<sup>10</sup><http://www.food-force.com>

**Diverted Games:** Häufig werden *Diverted Games* auch als *News Games* bezeichnet. In solchen Spielen werden aktuelle politische und geopolitische Probleme diskutiert, wie z.B. in *September 12th*<sup>11</sup>. Bei diesem Spiel wird auf eine einfache Art und Weise demonstriert, wie der Einsatz von Gewalt zu einer politischen Eskalation führen kann.

**Simulation Games:** In solchen Spielen werden Aspekte der Realität simuliert, um zum Teil komplexe Prozessabläufe, Wechselwirkungen, Möglichkeiten und Risiken für bestimmte Anwendungsbereiche aufzuzeigen. Zum einen liegt die erwünschte Lernintention darin, den Spielenden eine bestimmte Thematik zu vermitteln, wie z.B. Aspekte der Stadtplanung. Zum anderen sollen durch die Simulation Strategien und Methoden entdeckt, ausprobiert und optimiert werden, die möglicherweise auch in der Realität zu einem optimalen Ergebnis führen könnten. So beschäftigt sich beispielsweise das Spiel *Distributor Game* (Verbraeck und van Houten, 2005) mit Liefer- und Versorgungsketten im Zeitalter der Globalisierung. Dabei übernimmt man die Rolle eines Großhändlers, der seine Handelswaren in einem globalen Netzwerk aus Anbietern und Kunden optimal verwalten muss.

Über diese Klassifizierung hinaus, gibt es weitere Ausprägungen nach denen sich SG differenzieren lassen. So gibt es z.B. Unterschiede in den Zielgruppen (Kinder, Erwachsene, Fachkräfte,...), in der Art der Distribution (Online- oder Offline-Spiele) oder in der Mehrbenutzerfähigkeit (Einzel- oder Mehrspieler).

Für Industrie und Forschung ist die Kategorie *Simulation Games* besonders interessant, da man sich hier den größten wirtschaftlichen und wissenschaftlichen Nutzen verspricht (van der Zee und Slomp, 2005). Auch in der Analyse und dem Design dieser Arbeit wird der Fokus auf simulationsbasierte Spiele gelegt werden.

### 2.3.2 Forschungsschwerpunkte

Ähnlich wie der Bereich CSCW beschreibt auch Serious Games ein interdisziplinäres Forschungsfeld, das von verschiedenen Interessengruppen, z.B. aus Informatik, Soziologie, Pädagogik, Wirtschaftswissenschaften, Management, Politik und Bildungswesen, verfolgt wird.

Ein gemeinsamer Forschungsschwerpunkt besteht dabei nach Raybourn und Bos (2005) darin, mit den Designprinzipien und Technologien aus der Unterhaltungsbranche *Umgebungen für aktives, kritisches und nachhaltiges Lernen* bereitzustellen. Das Ziel ist, den Spielenden Möglichkeiten zum gleichzeitigen Lernen auf mehreren Ebenen zu eröffnen: Spieler lernen aus eingebetteten, kontextabhängigen Informationen, den Prozesszusammenhängen in der Spielmechanik, sowie durch Risiken, Nutzen, Kosten, Ausgänge und Belohnungen für alternative Strategien, die als Ergebnis aus der Entscheidungsfindung resultieren (Raybourn und Waern, 2004).

Die Erforschung solcher Umgebungen wird durch die unterschiedlichen Fachdisziplinen naturgemäß auch unter verschiedenen Aspekten angegangen: während beispielsweise aus sozial-

<sup>11</sup><http://september-12.freeonlinegames.com>

und bildungspolitischer Perspektive ein Hauptaugenmerk auf die Stärkung sozialer Kompetenzen (Raybourn und Waern, 2004) und die Untersuchung der Lerneffektivität von interaktiven Medien gelegt wird (Wong u. a., 2007), werden aus wirtschaftswissenschaftlicher Sicht mögliche Synergieeffekte untersucht, die die Entwicklung viel versprechender Geschäftsmodelle für Unternehmen ermöglichen (Fraunhofer-IuK, 2007).

Die Informatik beschäftigt sich in diesem Rahmen, neben der Identifikation und Adaption von geeigneten Designimplikationen sowie Entwicklungswerkzeugen, unter anderem mit *Massively Multiplayer Online Role-Playing Games* (MMORPGs), wie z.B. *World of Warcraft*<sup>12</sup> oder *Second Life*<sup>13</sup>. Mit diesen ist es möglich, dass sehr viele Benutzer gleichzeitig in einer persistenten Spielwelt miteinander interagieren können. Untersucht wird hier insbesondere, wie solche Plattformen als virtuelle Lern- und Kollaborationsumgebungen für mehrbenutzerfähige Serious Games einsetzbar sind.

### 2.3.3 Einordnung in das Forschungsgebiet CSCW

Die Untersuchung und Bereitstellung von „Umgebungen für aktives, kritisches und nachhaltiges Lernen“ sowie „virtuellen Lern- und Kollaborationsumgebungen“ (vgl. Abschnitt 2.3.2) führt zu Überschneidungen mit den Interessen des Forschungsgebiets CSCW. Nach Brown und Bell (2004) handelt es sich beispielsweise bei einer Vielzahl von Online-Multiplayer-Spielen um *Collaborative Virtual Environments* (vgl. Abschnitt 2.1.1). Die Merkmale dafür sind die Zusammenführung einer Gruppe in einen gemeinschaftlichen (virtuellen) Raum, die Verfolgung gemeinsamer Missionsziele und die Herstellung komplexer sozialer Strukturen durch Möglichkeiten zur Kommunikation und Interaktion. Serious Games, die diesen Ansprüchen genügen, lassen sich daher als eine konkrete Ausprägung von CSCW betrachten.

Eine ähnliche Sichtweise ergibt sich für kollaborative Serious Games, bei denen sich die Gruppenmitglieder nicht nur virtuell, sondern auch physisch „zur gleichen Zeit, am gleichen Ort befinden“. Solche Serious Games sind ebenfalls als eine konkrete Ausprägung von CSCW anzusehen. Diese lassen sich hierbei jedoch nicht als *Collaborative Virtual Environments*, sondern als Anwendungen in *Co-located Collaborative Workspaces* einordnen. Die weiteren Ausführungen in dieser Arbeit werden dieser Betrachtungsweise folgen.

---

<sup>12</sup><http://wow-europe.com>

<sup>13</sup><http://de.secondlife.com>

## 3 Analyse

In vielen Anwendungsbereichen ist es erforderlich, dass Mitarbeiter in kleinen Gruppen bzw. Teams die strategische Planung und Durchführung von gemeinsamen Bewegungsabläufen und Vorgehensweisen in unbekanntem und/oder teilweise komplexen räumlichen Umgebungen einüben sollen. Typische Beispiele für solche Anwendungsbereiche finden sich unter anderem bei militärischen Operationen, Sicherheitstrainings in Unternehmen oder Rettungseinsätzen von Polizei und Feuerwehr.

Die Ausgangssituation dieser Arbeit geht davon aus, dass diese Trainingsmaßnahmen in einem speziell dafür konzipierten Trainingsraum mit den Eigenschaften eines *Co-located Collaborative Workplace* (CCW) stattfinden (vgl. Abschnitt 2.1.3). Die Wissensvermittlung soll bei den hier geplanten Trainingsmaßnahmen jedoch nicht über ein klassisches Seminar oder eine Lehrveranstaltung, sondern über ein *interaktives und kollaboratives Serious Game* erfolgen.

Motiviert durch die Forschungsarbeiten im Bereich CSCW (siehe Unterkapitel 2.1) und durch Projekte wie dem *iRoom* (Johanson u. a., 2004) oder *i-Land* (Fraunhofer-IPSI, 2002) sind für die Gestaltung des Trainingsraums in dieser Arbeit insbesondere die folgenden Eigenschaften eines CCW interessant:

**Bereitstellung eines gemeinsamen Arbeitsbereichs:** Der gemeinsame Arbeitsbereich ist für alle Gruppenmitglieder im CCW zugänglich. In diesem können z.B. sehr detaillierte Informationen, Anwendungen oder Dokumente dargestellt und bearbeitet werden, die für alle Beteiligten interessant sind. Als Basis dafür ist ein großformatiges und hochauflösendes Anzeigemedium erforderlich, wie z.B. eine Powerwall (vgl. Unterkapitel 2.2).

**Bereitstellung eines gemeinsamen Anwendungskontextes:** Aus technischer Sicht besteht der CCW aus einer Vielzahl von unterschiedlichen Geräten und beschreibt damit eine sehr uneinheitliche Infrastruktur. Für eine optimale Unterstützung der Gruppenprozesse ist es notwendig, die Heterogenität der verschiedenen Geräte, insbesondere der Eingabe- und Ausgabemedien, durch die Bereitstellung eines gemeinsamen Anwendungskontextes zu „eliminieren“. Auf diese Weise können beispielsweise in der Metapher eines verteilten 3D-Desktops (Köckritz, 2007) Dokumente vom lokalen Arbeitsplatz (z.B. von einem Notebook oder Tabletop) auf den gemeinsamen Arbeitsbereich verschoben bzw. umgekehrt Dokumente oder sogar Anwendungsfenster von dem gemeinsamen Arbeitsbereich in den persönlichen Arbeitsbereich geholt werden.

Die Serious Games, die hier zur Mitarbeiterschulung eingesetzt werden sollen, sind als interaktive Lernanwendungen im CCW zu begreifen.

In diesem Kapitel wird zunächst untersucht, welche Designimplikationen sich bei der Entwicklung von Serious Games ergeben, welche lerntheoretischen Modelle für die Ausbildung von Individuen bzw. Gruppen zur Anwendung gebracht werden können und welche Art von Medien zur Unterstützung von gruppenbasierten Fortbildungsmaßnahmen geeignet sind (Unterkapitel 3.1). In Unterkapitel 3.2 wird daraufhin analysiert, wie sich hochauflösende Powerwalls effizient einsetzen lassen und welche Vorteile sich bei deren Verwendung ergeben. Im Anschluss daran wird in Unterkapitel 3.3 dargelegt, welche Strategien und Verfahren zur skalierbaren Visualisierung in dem Trainingsraum eingesetzt werden können, um die oben definierten Eigenschaften eines CCW erfüllen zu können. Zum Abschluss des Kapitels werden in Abschnitt 3.4 die Ergebnisse der Analyse zusammengeführt und ein Anwendungsszenario für ein spielbasiertes Teamtraining vorgestellt, auf dessen Basis eine Anforderungsanalyse für das Design und die Realisierung des Systems in Kapitel 4 erfolgt.

### 3.1 Serious Games für Teamtrainings

Alle Computerspiele beinhalten in irgendeiner Form einen Lernaspekt. So werden z.B. die Fähigkeiten in der Hand-Augen-Koordination oder in der räumlichen Wahrnehmung trainiert (Castel u. a., 2005). Aber nicht alle Spiele mit solchen Lernaspekten weisen dabei auch einen (Aus-)Bildungscharakter auf. Die wichtigste Herausforderung bei der Entwicklung von Serious Games und gleichzeitig der größte Unterschied zur klassischen Spieleindustrie ist nach Hartevelde u. a. (2007), die Ausgewogenheit zwischen den bildungsrelevanten Inhalten und den eigentlichen Spielelementen zu schaffen.

Nach Hartevelde u. a. (2007) müssen daher drei „gleichberechtigte“ Hauptziele beim Design von Serious Games berücksichtigt werden:

**Game:** Dieses Ziel beinhaltet alle Aspekte, die den Aufbau, den Ablauf und die Umsetzung des Spiels betreffen. Dazu zählen z.B. die Spielmechanik, die verwendeten Technologien und ggfs. der Unterhaltungswert.

**Pedagogy:** Für den Bildungscharakter eines Serious Game müssen pädagogische Methoden und Theorien angewendet werden, um eine Wissensvermittlung zu garantieren.

**Reality:** Der Wahrheitsgehalt bezieht sich auf den thematischen Inhalt des Spiels. Es müssen relevante Inhalte gelehrt werden, die auf die „reale“ Welt übertragen werden können. Bei dem Design muss darauf geachtet werden, inwiefern diese relevanten Inhalte zu den verwendeten pädagogischen Methoden passen und gleichzeitig in die Spielmechanik integriert werden können, ohne an Wahrheitsgehalt zu verlieren.

Das praktische Ziel dieser Arbeit ist die Entwicklung einer Umgebung für spielbasierte co-located Teamtrainings, sowie eines exemplarischen Serious Game, welches die Funktionalität dieser Umgebung veranschaulichen soll. Um dem Anspruch eines Serious Game nach Hartevelde u. a. (2007) zu genügen und gleichzeitig die Entwicklung unterschiedlicher Ausprägungen

von kollaborativen Serious Games für diese Umgebung zu unterstützen, müssen die oben genannten Hauptziele auch in der Konzeption dieser Arbeit berücksichtigt werden.

In diesem Unterkapitel wird daher untersucht, welche Designimplikationen sich für eine solche Trainingsumgebung und das Serious Game identifizieren lassen, die zur Erreichung der Hauptziele herangezogen werden können. In Abschnitt 3.1.1 wird das Hauptziel „Game“ mit dem Schwerpunkt auf spielbasierten Simulationen untersucht, da man sich bei diesen nach [van der Zee und Slomp \(2005\)](#) in Industrie und Forschung den größten wirtschaftlichen und wissenschaftlichen Nutzen verspricht (vgl. Abschnitt 2.3.1). Für das Hauptziel „Pedagogy“ werden in Abschnitt 3.1.2 Prozesse und Modelle aus der Lerntheorie untersucht, die einen erfolgreichen Wissenstransfer begünstigen können. Das Hauptziel „Reality“ wird in erster Linie von dem jeweiligen Anwendungskontext bestimmt. Die Identifikation von allgemeingültigen Designimplikationen ist bei den unterschiedlichen thematischen Inhalten nur in Bezug auf die Gruppenprozesse und deren Übertragbarkeit auf die Realität möglich. In Abschnitt 3.1.3 werden daher Theorien zur Mediennutzung bei der Gruppenarbeit betrachtet. Durch die Vorgabe der Ausgangssituation, dass der zu konzipierende Trainingsraum nach dem Vorbild eines CCW gestaltet sein soll, lassen sich hier Rückschlüsse darauf ziehen, für welche Art von Aufgaben das spielbasierte Teamtraining geeignet ist. In Abschnitt 3.1.4 werden die in diesem Unterkapitel identifizierten Designimplikationen für die Konzeption zusammengefasst.

### 3.1.1 Spielbasierte Simulationen (Game)

Zur Vorbereitung auf reale Aktionen oder Einsätze sind z.B. bei Sicherheitstrainings, militärischen Operationen oder bei Rettungseinsätzen von Polizei und Feuerwehr Trainingsmaßnahmen erforderlich. Die Durchführung dieser Trainingsmaßnahmen erfolgt „klassischerweise“ durch theoretische Lerneinheiten (z.B. Seminare oder Vorträge) und wird durch praktische Übungen in realen Trainingsumgebungen (z.B. in einem Übungsgebäude bzw. auf einem Übungsgelände) oder mit speziellen (realistischen) Simulationssystemen ergänzt. Der Aufbau von realen Trainingsumgebungen, die alle Aspekte des tatsächlichen Einsatzes berücksichtigen, ist in der Regel aus organisatorischen Gründen sehr schwierig bzw. nicht möglich, weil z.B. der Aufbau eines Übungsgebäudes nicht dem realen Einsatzort entspricht.

Diesbezüglich bieten spezielle Simulationssysteme bessere Möglichkeiten. Hier lassen sich Trainingsszenarien zur Verfügung stellen, die dem tatsächlichen Aufbau des realen Einsatzortes nachempfunden sind. Der Nachteil bei solchen Systemen ist oftmals, dass für die Visualisierung spezielle Hard- und Software-Komponenten erforderlich sind und sie damit einen sehr hohen Kostenfaktor mit sich bringen. Zudem handelt es sich bei diesen Systemen oftmals um Individualanfertigungen, so dass für den Einsatz in unterschiedlichen Anwendungsbereichen spezielle Adaptionen oder sogar vollständige Neuentwicklungen erforderlich sind. Damit sind solche Systeme zumeist nur Institutionen oder Unternehmen mit umfangreichen finanziellen Mitteln vorbehalten, wie z.B. dem Militär.

Der Einsatz von *spielbasierten Simulationen* als eine Ausprägung von Serious Games (vgl. Abschnitt 2.3.1) stellt hier eine sehr kostengünstige und beliebig anpassbare Alternative gegen-

über kostenintensiver Simulationssysteme dar. So erlauben beispielsweise moderne Grafik-Engines sehr detaillierte Nachbildungen von real existierenden Umgebungen, die mit Avataren in Echtzeit „durchwandert“ werden können. Unter Einbeziehung einer geeigneten Spielmechanik, computergesteuerten Nicht-Spieler-Charakteren (NSCs) und der Unterstützung von mehreren (verteilten) Spielern gleichzeitig, lassen sich sehr „lebendige und realistische Simulationsumgebungen“ (Sawyer, 2002) erschaffen. In diesen *spielbasierten Simulationen* können Gruppen miteinander interagieren, kooperieren und kommunizieren, um den Einsatz in der realen Vorlage der Simulationsumgebung bzw. „den Ernstfall“ zu trainieren.

Die Bezeichnung Serious Games impliziert nach Hartevelde u. a. (2007), dass ein gewisser Unterhaltungswert gegeben ist. Schon zu Beginn der *Serious Games Initiative* hat Sawyer (2002) die Allgemeingültigkeit dieser Auffassung in Bezug auf spielbasierte Simulationen relativiert. Grundsätzlich gibt es nach Sawyer (2002) nämlich zwei verschiedene Ansätze, um Technologien und Praktiken aus der Spieleindustrie in das weitere Feld von Simulationen zu integrieren:

- Aufwertung einer bestehenden Simulation
- Erstellung einer tatsächlichen spielbasierten Simulation

Jeder der beiden Ansätze verfolgt verschiedene Ziele und bringt unterschiedliche Anforderungen mit sich.

### **Aufwertung einer bestehenden Simulation**

Sawyer (2002) zeigt mit diesem Ansatz auf, dass Simulationen nicht wirklich Spiele sein müssen, um von den Technologien und Prinzipien der Unterhaltungsbranche profitieren zu können. Hier besteht das Ziel darin, verbesserte interaktive Versionen von bestehenden Simulationen (z.B. aus Industrie oder Wissenschaft) zu ermöglichen. Bei dem Einsatz der zur Verfügung stehenden Mittel für die Erweiterung der Simulationen ist es wichtig, den wissenschaftlich geprüften Teil des zugrundeliegenden Modells nicht zu beeinträchtigen oder gar zu gefährden. Nach Sawyer (2002) können z.B. folgende Fähigkeiten und Technologien zu dieser Erweiterung beitragen:

- Die Bereitstellung einer intuitiven grafischen Benutzerschnittstelle, die eine schnelle und einfache Interaktion mit der Simulation erlaubt.
- Die Verwendung verschiedener Techniken aus dem Bereich der künstlichen Intelligenz, um der Simulation ein realistisches bzw. realitätsnahes Eigenleben zu verschaffen oder parameterbasierte Reaktionen hervorzurufen.
- Die Verwendung aktueller Grafiktechnologien, um das visuelle Feedback und die Darstellungsgenauigkeit zu maximieren.

Simulationen, die auf diese Weise um die Eigenschaften eines kommerziellen Unterhaltungsprodukt ergänzt werden, haben nach Sawyer (2002) Chancen auf eine größere Verbreitung, Benutzung und Wirkungskraft.

### Erstellung einer tatsächlichen spielbasierten Simulation

Der typische Ansatz zur Erstellung eines Serious Game ist, Simulationen „als Spiele zu verpacken“ (Sawyer, 2002). Dafür ist erforderlich, dass das zugrundeliegende Modell interaktiv ist und die Schlüsselemente eines Spiels aufweist. Um diesen Ansprüchen zu genügen muss eine Simulation nach Sawyer (2002) die folgenden Eigenschaften aufweisen:

- Der Ausgang des Spiels muss durch die Benutzer steuerbar sein.
- Es muss ein übergeordnetes Ziel und mehrere Teilziele geben.
- Zu bewältigende Herausforderungen müssen auf Basis der Aktionen der Benutzer in positiven und negativen Ergebnissen münden.
- Es müssen mentale und/oder physische Fähigkeiten erforderlich sein.
- Der Ausgang des Spiels muss zu Beginn offen sein.
- Die Benutzer müssen für ein erfolgreiche Bewältigung des Spiels Strategien entwickeln können. Diese Strategien müssen zu Beginn des Spiels nicht offensichtlich sein. Das „Element der Entdeckung“ ist eine der wichtigsten Stärken von spielbasierten Simulationen.
- Es müssen mehrere Wege zum Erfolg führen. Lineare Spiele haben oftmals den Charakter eines Puzzles. Dabei geht es in der Regel um die Klärung eines speziellen Problems, jedoch nicht unbedingt um die Entwicklung einer (generalisierbaren) Strategie.
- Die Benutzer müssen alle Hindernisse im Spiel überwinden können. Nur in sehr seltenen Fällen kann es sinnvoll sein, Spiele zu konstruieren, die an einem gewissen Punkt nicht „gewinnbar“ sind.
- Das Spiel sollte interessant sein und einen gewissen Unterhaltungswert bieten, um zu wiederholtem Spielen zu motivieren.

Dieser Ansatz eignet sich nach Sawyer (2002) sehr gut dafür, Simulationen in bestimmte Trainingssituationen zu übertragen. Der Einsatz von spielbasierten Simulationen führt zu Ausbildungsanwendungen, die beispielsweise zur Förderung strategischen Denkens oder zur Stärkung der von kooperativen Fähigkeiten eingesetzt werden können. Der Unterhaltungswert führt zudem zu einer wiederholten Benutzung und fördert die Motivation im Lernprozess. In der Konzeption und Realisierung des Systems (Kapitel 4) wird dieser Ansatz Verwendung finden.

#### 3.1.2 Prozesse und Modelle aus der Lerntheorie (Pedagogy)

Wie in den Grundlagen in Abschnitt 2.3.2 erläutert wurde, bezeichnet *Serious Games* ein interdisziplinäres Forschungsfeld. Der wesentliche Schwerpunkt dieser Arbeit liegt auf der informatischen Sichtweise. Dennoch ist es für die Konzeption und das Design wichtig, auch den

pädagogischen Aspekt zu berücksichtigen (Harteveld u. a., 2007), um die erwünschte Wissensvermittlung durch Serious Games zu gewährleisten. Als Grundlage hierfür können Prozesse und Modelle aus der Lerntheorie der Psychologie herangezogen werden.

Das Lernen bezeichnet einen komplexen Prozess, in dessen Verlauf Wissen, Einstellungen und Verhaltensweisen erzeugt werden. Nach Sacher (1990) lassen sich grundsätzlich zwei Arten des Lernens unterscheiden: konzipierendes Lernen und rezipierendes Lernen.

Beim **konzipierendem Lernen** lernt der Mensch durch praktische Anwendung. Er beschäftigt sich mit ausgewählten Elementen seiner Umwelt und erarbeitet sich neues Wissen durch explorative Experimente mit diesen Elementen. Die Gefahr bei dieser Form des Lernens, dass aufgrund von fehlendem oder teilweise auch fehlerhaftem Hintergrundwissen falsche Schlussfolgerungen gezogen werden und der Mensch sich somit falsches Wissen aneignet. Nach Haag (1995) ist konzipierendes Lernen daher nur dann sinnvoll, wenn bereits *korrektes Wissen* vorhanden ist, in welches das neue Wissen integriert werden kann.

Das **rezipierende Lernen** ist die Grundlage der Weiterentwicklung der menschlichen Kultur. Bei dieser Art des Lernens lenkt der Mensch seine Aufmerksamkeit auf bestehendes Wissen bzw. auf die Ergebnisse eines anderen Lernprozesses, wie z.B. durch eine Analyse bestehender wissenschaftlicher Arbeiten. Durch die Verwendung der Erkenntnisse und dem (kompakten) Wissen anderer Menschen kann eine zeitaufwendige explorative Erforschung übersprungen und sich schnell neues Wissen angeeignet werden. Die Schwierigkeit ist hierbei *das Verstehen* und *die Integration* des neuen Wissens in das bereits vorhandene Wissen. Obwohl durch rezipierendes Lernen aneignetes Wissen in der Regel vielfach durchdacht und bewährt ist, besteht die Gefahr, dass ein *falsches Verstehen* wiederum zu *falschem Wissen* führt (Haag, 1995).

Nach Sacher (1990) sind neue Erkenntnisse nur durch konzipierendes Lernen möglich. Daher kommt es bei der menschlichen Entwicklung sehr auf das richtige Verhältnis zwischen konzipierendem und rezipierendem Lernen an.

Bislang konnte wissenschaftlich noch nicht einwandfrei geklärt werden, auf welche Weise *deklaratives Wissen*<sup>1</sup> im menschlichen Gehirn tatsächlich „gespeichert“ wird. Mit der genauen Kenntnis dessen, wie diese „Kodierung von Informationen“ funktioniert, könnte der Lernprozess darauf aufgestimmt werden, indem beispielsweise Lernstrategien mit verstärktem Einsatz von Bildern oder sprachlichen Komponenten gewählt werden (Baumgartner und Payr, 1999). Bei der Konzeption von Software mit bildungsrelevanten Inhalten sollten Informationen daher nach Möglichkeit auf verschiedene Arten präsentiert werden. Damit kann den Benutzern offen gestellt werden, in welcher Form sie diese aufnehmen können (Haag, 1995).

Im Verlauf der Forschungen zum Verständnis der menschlichen Wissensaufnahme wurden verschiedene Theorien entwickelt, was unter Lernen zu verstehen ist, nach welchen Gesetzmäßigkeiten es funktioniert und wie es gezielt unterstützt werden kann. Im Folgenden werden

---

<sup>1</sup>Wissen wird als *deklarativ* bezeichnet, wenn sich die thematischen Inhalte auf Fakten beziehen und sprachlich in Form von Aussagesätzen beschrieben werden können.

die drei bekanntesten Lernparadigmen vorgestellt, bewertet und deren Verwendbarkeit in Bildungssoftware skizziert.

### Behaviorismus

Der Behaviorismus ist eines der ersten Lernparadigma aus der Psychologie. Hier stehen die konkreten internen Vorgänge des Gehirns im Hintergrund. Es wird vielmehr als eine Black-Box betrachtet. Bei dieser Theorie wird davon ausgegangen, dass das menschliche Gehirn nur auf eine geeignete Art und Weise gereizt werden muss, um eine bestimmte Reaktion hervorzurufen. Die Schwierigkeiten den Behaviorismus didaktisch umzusetzen, bestehen nach [Baumgartner und Payr \(1999\)](#) insbesondere darin, *geeignete Stimuli* und *passende Resonanzen* (d.h. „Belohnungen“ und „Strafen“) zu identifizieren, um die erwünschte Verhaltensweise zu verstärken.

Diese Theorie ist mittlerweile stark in die Kritik geraten, da die Vereinfachung zu einem *Stimulus-Response-Modell* ([Skinner, 1953](#)) die Komplexität der menschlichen Lernprozesse nach modernen Auffassungen nicht komplett erschliessen kann ([Schink, 1993](#)). Für die Aneignung von physischen Fertigkeiten konnten mit dem Behaviorismus allerdings gute Erfolge erzielt werden, so dass dieser für solche Zwecke auch heute noch eingesetzt wird.

Ein auf diesem Paradigma basierendes computergestütztes Lernsystem ist nach [Faulhaber \(1996\)](#) „[...] eine absolute Autorität, welche beurteilt und weiß, was richtig und falsch ist“. Das zu vermittelnde Wissen muss dazu in sehr feingranulare Elemente gegliedert werden. Diese Elemente werden sodann einzeln präsentiert und unmittelbar mit Kontrollfragen überprüft. Problematisch ist hierbei, dass es mitunter sehr schwierig sein kann, komplexe Zusammenhänge geeignet zu gliedern und darzustellen.

### Kognitivismus

Beim Kognitivismus stehen, als Gegenreaktion zum Behaviorismus, die im Gehirn stattfindenden Prozesse im Vordergrund, die eine Eingabe zu einer Ausgabe verarbeiten. Diese Verarbeitungsprozesse werden in verschiedenen Ausprägungen des Kognitivismus untersucht, unterschieden und in ihrer jeweiligen Funktion miteinander in Beziehung gesetzt. Im Allgemeinen wird Lernen dabei als ein *Informationsverarbeitungsprozess* angesehen, dessen genaue Funktionalität für jedes Problem gefunden werden muss. Anders als beim Behaviorismus geht es hierbei also um die *Identifikation von geeigneten Methoden und Verfahren*, deren Anwendung dann zur Ausgabe von richtigen Antworten führen.

Auch der Kognitivismus ist ein umstrittenes Lernparadigma. Der Fokus liegt vornehmlich auf der Erklärung geistiger Verarbeitungsprozesse. Die Aneignung physischer Fertigkeiten lässt sich auf Basis der Identifikation von Methoden und Verfahren nur sehr schwer nachweisen. Wie auch schon beim Behaviorismus bezieht sich ein weiterer Kritikpunkt an diesem Paradigma auf die Vereinfachung und einseitige Betrachtung der menschlichen Lernprozesse. Es

wird von einer einzigen, objektiven und erkennbaren Realität ausgegangen. Demnach ist die Lösung eines Problems „eine Frage der Zeit“, wenn ein Problem objektiv gegeben ist und repräsentiert werden kann. Beim Kognitivismus wird durch diese Realitätswahrnehmung vernachlässigt, dass Probleme erst einmal entdeckt (d.h. identifiziert bzw. konstruiert) werden müssen, bevor sie gelöst werden können.

Computergestützte Lernsysteme, die auf den Kognitivismus aufbauen, beschreiben den Benutzern nach [Faulhaber \(1996\)](#) bestimmte Situationen auf der Basis verschiedener kennzeichnender Faktoren und die zu lösenden Probleme in dieser Situation. Das Ziel sollte dabei nicht sein, eine „einzig wahre“ Lösung zu finden, sondern verschiedene Verfahren zu erkunden, die zu optimalen Ergebnissen führen. Die Benutzer stehen in einem aktiven Dialog mit dem System, so dass es bei Bedarf erforderliche Hilfestellungen zur Problemlösung geben kann. Die Rolle des Systems ist nach [Faulhaber \(1996\)](#) „ein beobachtender und helfender Tutor“.

### Konstruktivismus

Der Konstruktivismus berücksichtigt auch den Aspekt der Problemidentifizierung bzw. der Problemkonstruktion. Dieses Paradigma betrachtet die Erarbeitung von Wissen als subjektiven Konstruktionsprozess. Dadurch wird der Beobachter (Subjekt) auch mit dem Kontext bzw. der Situation (Objekt) verknüpft, in welcher Wissen erworben wird. Es gibt keine „einzig wahre“ objektive Realität. Die gleiche „Situation“ wird demnach beispielsweise von zwei Individuen auch aus zwei verschiedenen Perspektiven betrachtet und wahrgenommen. Durch die Verbindung zu diesem kontextuellen Wissen werden deklaratives und prozedurales Wissen „situier“.

Im Konstruktivismus ist das Lernen ein aktiver, selbstgesteuerter, konstruktiver, situativer und sozialer Prozess ([Gerstenmaier und Mandl, 1994](#)). In komplexen realen Lebenssituationen konstruieren Menschen ihr Wissen in Beziehungen zu früheren Erfahrungen. Konstruktivistische Lehr- und Lernmodelle verlangen dementsprechend *situieretes Lernen*. Im Kontext von situierem Lernen haben sich verschiedene Ansätze entwickelt, die sich auf die theoretischen Annahmen zur Situiertheit von Wissen und Lernen beziehen. Bei der Gestaltung der Lernumgebungen sind nach [Mandl u. a. \(1997\)](#) die folgenden grundlegenden Prinzipien, die allen Ansätzen des situiereten Lernens gemeinsam sind, zu berücksichtigen:

- die Darstellung komplexer Ausgangsprobleme und sozialer Realitäten
- die authentischen und situiereten Aktivitäten von Lernenden
- die Präsentation multipler (statt einfacher) Perspektiven auf Probleme
- die Artikulation und Reflexion von Vorgehensweisen und Problemen
- das Lernen im sozialen Austausch

Die bekanntesten Ansätze zum situiereten Lernen sind die *Cognitive Flexibility Theory*, der *Cognitive Apprenticeship Ansatz* und der *Anchored Instruction Ansatz*. Diese Instruktionsansätze legen ihren Fokus auf Lernen durch aktives Lösen von Problemen. Vertiefende Informationen zu diesen Ansätzen finden sich in [Mandl u. a. \(1997\)](#).

Nach [Faulhaber \(1996\)](#) sind Simulationen gute Beispiele für computergestützte Lernsysteme nach dem konstruktivistischen Ansatz. In diesen lassen sich Situationen bzw. Szenarien aufbauen, die eine Reihe von Faktoren beinhalten, welche sich wiederum wechselseitig beeinflussen. Simulationen sind in der Regel „offen“ für Problemstellungen, d.h. es werden keine expliziten Probleme vorgegeben bzw. die Probleme sind noch gar nicht bekannt. Der Lernende muss diese erst aus der Gesamtsituation ermitteln und anschließend lösen. Durch die Veränderung der Faktoren und der Reaktion des Simulationsmodells können auch komplexe Zusammenhänge und Probleme erlernt werden. In diesem Fall nimmt das System nach [Faulhaber \(1996\)](#) die Rolle „eines Coaches oder Moderators“ ein und ist damit auch nicht mehr *unfehlbar* ([Baumgartner und Payr, 1999](#)).

### 3.1.3 Theorien zur Mediennutzung bei der Gruppenarbeit (Reality)

Neben den, durch den konkreten Anwendungskontext gegebenen, thematischen Inhalten des Serious Game, steht bei dem in dieser Arbeit zu konzipierenden Trainingsraum die Förderung von Kommunikation, Koordination und Kollaboration der Teammitglieder im Vordergrund. Wie zu Beginn dieses Kapitels beschrieben, soll die Infrastruktur des Trainingsraums dazu die Eigenschaften eines CCW aufweisen.

Der Wahrheitsgehalt der thematischen Inhalte eines Serious Game muss bei der Entwicklung eines jeden Spiels neu geprüft werden. Für die Übertragbarkeit der trainierten Gruppenprozesse auf die Realität lassen sich jedoch gemeinsame Muster erkennen. Als Grundlage dafür lassen sich Theorien zur Mediennutzung bei der Gruppenarbeit heranziehen. Diese Theorien bilden Anhaltspunkte dafür, für welche Art von Aufgaben oder Problemstellungen, welche Medien benötigt werden. Da die Infrastruktur, und damit die eingesetzten Medien, durch die Eigenschaften eines CCW vorgegeben sind, werden diese Theorien in diesem Abschnitt daher rückwärts interpretiert. Damit soll festgestellt werden, für welche Art von Aufgaben die co-located Teamtrainings in dem zu konzipierenden Trainingsraum geeignet sind.

#### Media-Richness-Theorie

Die *Media-Richness-Theorie* ([Daft und Lengel, 1986](#)) stellt eine Verbindung zwischen der Auswahl der Medien und der Aufgabe, die die beteiligten Personen (Akteure) gemeinsam lösen wollen, her. Dabei erfolgt eine Aufteilung in *unsichere Aufgaben* und *mehrdeutige Aufgaben*.

Eine optimale Bewältigung *unsicherer Aufgaben* ist möglich, wenn alle benötigten Informationen vorhanden wären. *Mehrdeutige Aufgaben* sind dagegen durch die alleinige Bereitstellung von sehr vielen Informationen nicht lösbar. Sie erfordern vielmehr die Interpretationsfähigkeit der Akteure, die zu einem gemeinsamen Verständnis eines Sachverhalts kommen müssen. So ist nach [Schwabe \(2001\)](#) beispielsweise die Formulierung einer Unternehmensstrategie eine mehrdeutige Aufgabe, weil sich die Akteure dabei auf ein gemeinsames Verständnis der Rolle des Unternehmens im Markt verständigen müssen. Als Grundlage dafür muss zunächst eine

Definition der relevanten Einflussfaktoren erfolgen. Bei mehrdeutigen Aufgaben werden demnach *Variablen* gesucht, während bei unsicheren Aufgaben die *Werte von Variablen* gesucht werden (Schwabe, 2001).

Daft und Lengel (1986) empfehlen, für unsichere Aufgaben Medien zu verwenden, die viele Informationen vermitteln (z.B. schriftliche Berichte). Für mehrdeutige Aufgaben sind „reiche“ Medien erforderlich (z.B. Sitzungen). Das Maß für den Reichtum (Richness) eines Mediums kann man nach Schwabe (2001) an der Unmittelbarkeit des Feedbacks, der Anzahl der Informationskanäle, der Menge an Informationen, der Persönlichkeit der Kommunikation und der Vielfältigkeit der vermittelten Sprache messen. Nach Daft und Lengel (1986) führt der Einsatz von *besser geeigneten* Medien zu einer höheren Effektivität der Aufgabenerfüllung.

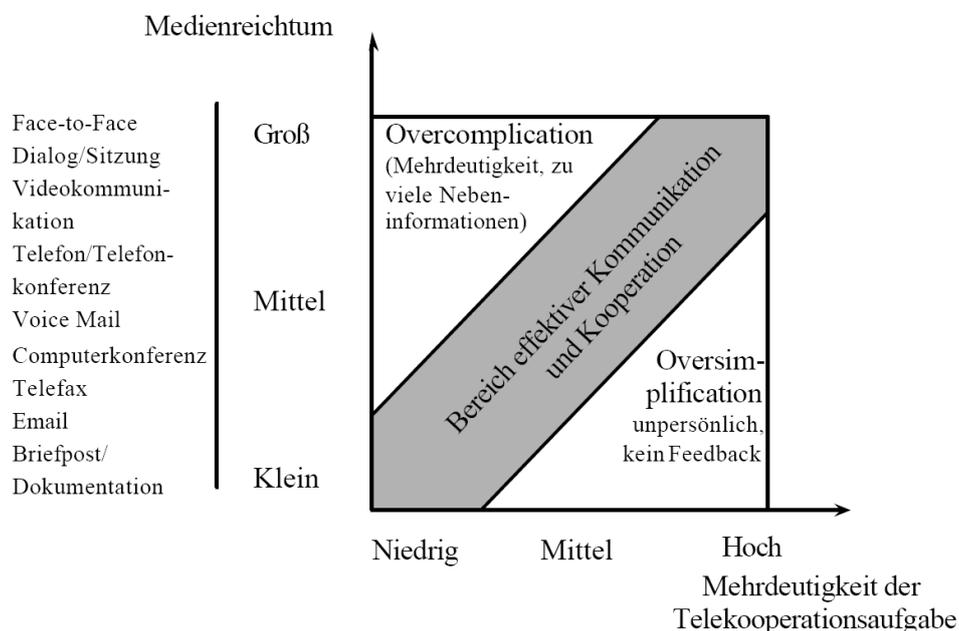


Abb. 3.1: Das Media-Richness-Modell für die Telekooperation (nach Reichwald u. a. (1998))

Die ursprüngliche Fassung der Media-Richness-Theorie von Daft und Lengel (1986) wurde von Rice (1992) für neue Medien weiterentwickelt. Darauf basierend wurde ein Media-Richness-Modell für die Telekooperation konzipiert (Reichwald u. a., 1998). In diesem haben Face-to-Face-Dialoge bzw. Sitzungen den größten Medienreichtum. Dagegen haben beispielsweise Briefpost und schriftliche Dokumentationen den kleinsten Medienreichtum. Die Wahl geeigneter Medien steht in direkter Abhängigkeit zu dem Maß an Mehrdeutigkeit einer Aufgabe. Es gibt nach Reichwald u. a. (1998) einen Bereich *effektiver Kommunikation* (siehe Abb. 3.1).

Zu „reiche“ Medien erhöhen die eigentliche Komplexität der Aufgabensituation („Overcomplication“). Die Akteure werden durch den Reichtum der Medien abgelenkt und können sich nicht mehr auf die eigentliche Aufgabe konzentrieren. Dabei wird nach Schwabe (2001) Mehrdeutigkeit möglicherweise künstlich erzeugt. Zu „arme“ Medien führen dagegen zu einer *vereinfachten* Darstellung der Aufgabensituation („Oversimplification“). Die eingesetzten Medien

eignen sich nur für eine Informationssuche, obwohl ein gemeinsames Verständnis durch gemeinsame Interpretation gefragt ist.

Nach der Media-Richness-Theorie ist ein Co-located Collaborative Workspace (CCW) mit der Möglichkeit zur Face-to-Face-Kommunikation und der Menge an darstellbaren Informationen als eine Umgebung mit „reichen“ Medien einzuschätzen. Um einen Bereich effektiver Kommunikation und Kooperation zu erreichen, eignet sich der CCW damit insbesondere für mehrdeutige Aufgaben. Hinsichtlich der Konzeption eines Serious Games in dieser Umgebung empfiehlt es sich daher, auf Themengebiete zu setzen, die nicht nur allein durch eine Vielzahl von Informationen darstellbar sind, sondern auch die Interpretationsfähigkeit der beteiligten Akteure erfordern.

### Media-Synchronicity-Theorie

In der Media-Richness-Theorie wird davon ausgegangen, dass die Eigenschaften der Aufgabe und die Anforderungen des Anwendungskontextes eine optimale Medienwahl bestimmen. Nach [Dennis und Valacich \(1999\)](#) ist dieser Grundansatz der Media-Richness-Theorie zu ungenau. Vielmehr wird die Mediennutzung durch die *Art des Kooperationsprozesses* und dessen *Anforderung an die Informationsverarbeitungskapazität* eines Mediums geprägt.

Darauf basierend wurde die Media-Synchronicity-Theorie ([Dennis und Valacich, 1999](#)) entwickelt, in der zwei generische Kooperationsprozesse unterschieden werden: konvergente Prozesse und divergente Prozesse. In divergenten Prozessen erfolgt eine Informationsverteilung, während in konvergenten Prozessen eine Informationsverdichtung stattfindet, um ein gemeinsames Verständnis zu erlangen. Nach [Schwabe \(2001\)](#) sind divergente Prozesse damit für die Reduktion von Unsicherheit und konvergente Prozesse zur Reduktion von Mehrdeutigkeit geeignet.

Gegenüber der Media-Richness-Theorie erweitert die Media-Synchronicity-Theorie die ausschlaggebenden Eigenschaften eines Mediums. Danach ist nicht (nur) der „Reichtum“ eines Mediums von Relevanz, sondern (vor allem auch) dessen Synchronität. [Dennis und Valacich \(1999\)](#) beschreiben die Mediensynchronität als „[...] das Ausmaß, in dem Individuen an der gleichen Aufgabe zur gleichen Zeit zusammenarbeiten, d.h. einen gemeinsamen Fokus haben“. Dieses Ausmaß wird in der Media-Synchronicity-Theorie anhand von fünf Faktoren beschrieben (in Anlehnung an [Schwabe \(2001\)](#)):

**Geschwindigkeit des Feedbacks**, d.h. die Geschwindigkeit, mit der ein Kooperationspartner auf Nachrichten antworten bzw. reagieren kann.

**Symbolvarietät**, d.h. die Vielfältigkeit der Informationsübermittlung und die Menge an Informationen.

**Parallelität**, d.h. die Anzahl der Kanäle, über die Akteure gleichzeitig kooperieren bzw. kommunizieren können.

**Überarbeitbarkeit**, d.h. die Möglichkeit, Informationen (z.B. einen Arbeitsbeitrag) vor der Verteilung zu überarbeiten.

**Wiederverwendbarkeit**, d.h. die Möglichkeit, Informationen (z.B. Arbeitsbeitrag von Kooperationspartnern) geeignet wiederverwenden zu können.

Die Abbildung 3.2 zeigt eine Einordnung dieser Faktoren in die Kommunikation und Kooperation zwischen den Akteuren (Sender und Empfänger).

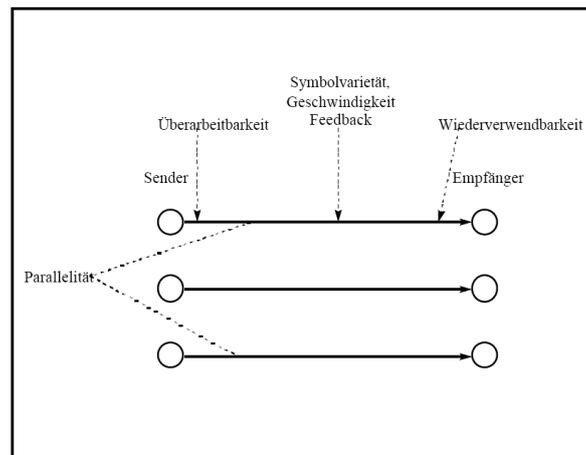


Abb. 3.2: Medieneigenschaften in der Media-Synchronicity-Theorie (Schwabe, 2001)

Für die Betrachtung der Mediensynchronität haben die beiden Faktoren *Geschwindigkeit des Feedbacks* und *Parallelität* das stärkste Gewicht. Medien mit der Möglichkeit zu einem schnellen Feedback und geringer Parallelität ermöglichen eine hohe Synchronität, während Medien mit langsamen Feedback und hoher Parallelität eine geringe Synchronität zur Folge haben. Im Zusammenhang mit den zuvor dargelegten generischen Kooperationsprozessen gilt nach Schwabe (2001), dass konvergente Prozesse Medien mit hoher Synchronität und divergente Prozesse entsprechend Medien mit geringer Synchronität erfordern. Weiterhin zeigt Schwabe (2001) auf, dass Medien mit einer höheren Überarbeitbarkeit zu einer besseren Leistung führen können. Speziell in konvergenten Prozessen, also bei Prozessen mit Informationsverdichtung, schlagen sich Medien mit höherer Wiederverwendbarkeit in einer gesteigerten Leistung nieder. Der Bedarf an Synchronität steht in unmittelbarem Zusammenhang mit der gemeinsamen Erfahrung der Gruppe. Eingespielte Gruppen benötigen weniger Synchronität als solche Gruppen, die sich gerade neu formiert haben.

Nach der Media-Synchronicity-Theorie ist ein Co-located Collaborative Workspace (CCW) mit der Möglichkeit zum direkten Dialog (Face-to-Face) und geringer Parallelität (in der Regel spricht dabei nur eine Person) als eine Umgebung mit hoher Mediensynchronität einzuschätzen. Damit eignet sich der CCW in diesem Zusammenhang für konvergente Prozesse. Hinsichtlich der Konzeption eines Serious Games in dieser Umgebung empfiehlt es sich daher, auf Themengebiete zu setzen, bei denen ein gemeinsames Verständnis der gesamten Gruppe durch eine Verdichtung von Informationen erreicht werden kann.

### 3.1.4 Fazit

Für die Konzeption der in dieser Arbeit geplanten Lernumgebung ist es wichtig, die wesentlichen Eigenschaften eines Serious Game nach [Harteveld u. a. \(2007\)](#) zu berücksichtigen und damit gleichzeitig die Entwicklung unterschiedlicher Ausprägungen von kollaborativen Serious Games für diese Umgebung zu ermöglichen. Daher werden in diesem Abschnitt einige Designimplikationen für die Entwicklung von Serious Games vorgestellt, die aus der zuvor dargelegten Analyse in diesem Unterkapitel abgeleitet werden konnten.

**Game:** Die in Abschnitt [3.1.1](#) vorgestellten Eigenschaften von spielbasierten Simulationen lassen sich zu folgenden Designimplikationen zusammenfassen:

- **Interaktivität:** Das zugrundeliegende Modell muss interaktiv sein und den Benutzern damit die Möglichkeit bieten, den weiteren Spielverlauf und den Spielausgang beeinflussen zu können.
- **Modellplausibilität:** Damit die spielbasierte Simulation Eingaben und Informationen der Benutzer richtig interpretieren kann, muss sich das zugrundeliegende Modell konsistent und kohärent verhalten.
- **Unsicherheit:** Der Spielverlauf und der Spielausgang kann eingeschätzt werden, darf aber zu keinem Zeitpunkt sicher bestimmbar sein.
- **Motivation:** Die Aufmerksamkeit der Benutzer muss kontinuierlich durch ein Belohnungssystem, bei dem zu bewältigende Herausforderungen auf Basis der Aktionen der Benutzer in positiven und negativen Ergebnissen münden, aufrecht erhalten werden.
- **Spielfluss:** Die Benutzer müssen zu jeder Zeit darüber aufgeklärt sein, welche Aufgaben sie zu erfüllen haben. Der Schwierigkeitsgrad muss angemessen sein und das Spiel darf dabei keine unüberwindbaren Hindernisse aufweisen.

Die Aspekte bezüglich der technischen Realisierungsmöglichkeiten wurden in dieser Analyse nicht berücksichtigt und erst in Kapitel [4](#) (Design und Realisierung) betrachtet.

**Pedagogy:** Wie in Abschnitt [3.1.2](#) dargelegt, ist der konstruktivistische Ansatz ein anerkanntes und zeitgemäßes lerntheoretisches Modell, das sich durch situiertes Lernen auf Basis von (spielbasierten) Simulationen realisieren lässt. Diesbezüglich lassen sich folgende Designimplikationen zusammenfassen:

- **Reflektion:** Die Artikulation und Reflexion von Vorgehensweisen und Problemen ist wichtig, um von konkreten Situationen zu allgemeingültigeren Abstraktionen zu gelangen ([Harteveld u. a., 2007](#)). Reflektion kann durch einen Ausbilder stimuliert oder direkt als Teil des Spiels integriert werden.
- **Perspektiven:** Probleme werden von Menschen stets subjektiv wahrgenommen. Um die vielfältigere (objektivere) Sicht auf die Problematik zu ermöglichen, sollte diese aus unterschiedlichen Perspektiven präsentiert werden.

- **Angemessenheit:** Faktoren, wie z.B. eine „Informationsflut“ oder Zeitdruck in einer Spielumgebung, können - wenn sie nicht Teil des Lerninhaltes sind - zu einer kognitiven Überbelastung führen oder die Benutzer dazu bringen, wichtige Informationen zu übersehen.
- **Erkundung:** In realen Lebenssituationen erkunden Menschen ihre Umgebung, um sich zu orientieren, Probleme zu identifizieren und entsprechende Handlungsalternativen zu entwickeln. Analog sollte auch eine spielbasierte Lernumgebung diese Möglichkeit bieten und den Benutzer nicht in enge Handlungsvorgaben zwingen.
- **Inkrementalität:** In komplexen realen Lebenssituationen konstruieren Menschen ihr Wissen in Beziehung zu früheren Erfahrungen. Dementsprechend sollten auch die Inhalte eines Spiels inkrementell gesteigert werden, damit die Benutzer auf eine bestehende Erfahrungsbasis zurückgreifen können.

Es ist nicht ausgeschlossen, auch Teilaspekte anderer Lernmodelle implizit oder explizit einfließen zu lassen. Die Designimplikation *Motivation* des Bestandteils *Game* impliziert beispielsweise ein Belohnungssystem, welches wiederum eine einfache Form des Stimulus-Response-Modell des Behaviorismus darstellt.

**Reality:** Der Bestandteil Reality wird in erster Linie von dem jeweiligen Anwendungskontext bestimmt. In dieser Analyse wurde die Betrachtung daher in Abschnitt 3.1.3 auf die Art der möglichen Aufgabensituationen und Gruppenprozesse eingeschränkt. Hier lassen sich folgende Designimplikationen festhalten:

- **Mehrdeutige Aufgabensituationen:** Nach der Media-Richness-Theorie ist der CCW eine Umgebung mit „reichen“ Medien und eignet sich damit zur Darstellung von Themengebieten mit mehrdeutigen und komplexen Aufgabensituationen für die Gruppe. Dies korreliert unmittelbar mit einem der Grundansätze des situierten Lernens, bei dem die Darstellung komplexer Ausgangsprobleme und sozialer Realitäten in einer konstruktivistischen Lernumgebung möglich sein sollte (Mandl u. a., 1997).
- **Unterstützung konvergenter Gruppenprozesse:** Nach der Media-Synchronicity-Theorie ist der CCW eine Umgebung mit hoher Mediensynchronität und eignet sich damit für konvergente Gruppenprozesse, die zu einer Reduktion der Mehrdeutigkeit einer Aufgabe verwendet werden können. Hierfür müssen entsprechende Möglichkeiten zur Verfügung stehen, damit die Gruppenmitglieder die Informationen einer Aufgabensituation sukzessive verdichten können. In einer konstruktivistischen Lernumgebung können diese Aspekte durch die zuvor dargestellten Designimplikationen *Reflektion*, *Perspektiven* und *Erkundung* unterstützt werden.

Der Wahrheitsgehalt der thematischen Inhalte eines Serious Game, wie z.B. die korrekte Abbildung der realen Prozesse oder die konkreten Lernziele, muss bei der Entwicklung eines jeden Spiels neu geprüft werden.

Die hier dargestellte Aufstellung von Designimplikationen ist als empfohlene Richtlinie zu verstehen, die nicht unbedingt vollständig in die Konzeption einfließen muss.

## 3.2 Effizienter Einsatz von hochauflösenden Powerwalls

Während die Realisierung von hochauflösenden Powerwalls vor einigen Jahren aufgrund spezieller Hardwareanforderungen noch sehr kostenintensiv war, ermöglichen heutzutage softwarebasierte skalierbare Visualisierungsstrategien<sup>2</sup> eine verhältnismäßig günstige Umsetzung mit Standardkomponenten (Bezerianos und Balakrishnan, 2005). Dies führte dazu, dass in den letzten Jahren der Einsatz von Powerwalls verstärkt ins Interesse der Forschung gerückt ist. Die Schwerpunkte liegen dabei auf der Untersuchung der Effekte in Bezug auf Produktivität sowie Techniken, um den Umgang mit Powerwalls effizienter zu gestalten.

Ein Ansatz für diese Untersuchungen liefern solche Anwendungsbereiche, in denen der Bedarf an dem Einsatz von Powerwalls durch die konsequente Weiterentwicklung von traditionellen Arbeitstechniken und vertrauten Interaktionskonzepten „historisch gewachsen“ ist (Buxton u. a., 2000). Eine solche „evolutionäre Bedarfsentwicklung“ lässt sich zum Beispiel an den klassischen Anforderungen bzw. Vorgehensweisen in der Automobilkonstruktion aufzeigen:



Abb. 3.3: Traditionelle Arbeitstechniken und Kollaboration an großformatigen, senkrechten Flächen in der Automobilkonstruktion (links von Mini (2006), rechts von Volkswagen-AG (2007))

**Großformatige Konstruktionszeichnungen:** Gerade in der frühen Designphase neuer Automobilmodelle ist es wichtig, alle Details zu erfassen. Konstruktionszeichnungen mit originalgetreuen, aber kleineren Maßstäben sind dafür ungeeignet. Daher erfolgt das Design bei der Automobilkonstruktion in der Regel in sehr großen Maßstäben, teilweise sogar bis zu einem 1:1-Maßstab. Dies geschieht entweder durch die Zusammenführung mehrerer Teilbilder an großen pinnwandähnlichen Flächen oder mittels dem sogenannten *Tape Drawing* (Abb. 3.3, links). Damit wird eine Technik bezeichnet, bei der die Konstruktion unmittelbar auf einer großformatigen, senkrechten Fläche unter Verwendung eines speziellen Klebebands vorgenommen wird.

<sup>2</sup>Konkrete Strategien und Verfahren zur skalierbaren Visualisierung werden in Unterkapitel 3.3 besprochen.

**Realitätsgetreue 3D-Modelle:** Um ein Gefühl für das angestrebte Produkt zu bekommen, reichen weder zweidimensionale Konstruktionszeichnungen noch dreidimensionale CAD-Modelle an Workstations mit klassischen (kleinen) Anzeigemedien aus. Hier wird in der Automobilkonstruktion auf die Herstellung von sehr großen, dreidimensionalen Modellen aus einer speziellen Modelliermasse gesetzt. Auch hier ist der Maßstab sehr wichtig, um sämtliche Details berücksichtigen zu können. Bei entsprechender Gestaltung der Oberflächen bekommen alle an der Entwicklung beteiligten Personen auf diese Weise ein lebensnahes Modell, das als Diskussionsgrundlage für konzeptionelle Veränderungen bzw. Verbesserungen herangezogen werden kann.

**Kollaboration im Designprozess:** Ein weiterer wichtiger Faktor in der Automobilkonstruktion ist die kollaborative Natur des Designprozesses. Bis zur endgültigen Genehmigung eines Designs müssen die Designer kontinuierlich ihre Ideen vermitteln, Ergebnisse bzw. Teilergebnisse präsentieren und diesbezügliche Resonanzen bei ihren Kollegen und dem Management einholen. Nach [Buxton u. a. \(2000\)](#) veranschlagt allein der im Genehmigungsprozess eines Designs notwendige Kommunikationsbedarf in etwa ein Drittel der Zeit des gesamten Designzyklus. Für eine effektive Zusammenarbeit haben sich hier große, pinnwandähnliche Flächen bewährt (Abb. 3.3, rechts). Diese dienen als Verwahrungsort für Ideen, Skizzen und Konzepte sowie als „interaktive Kollaborationsplattform“. Mehrere Designer haben somit einen Gesamtüberblick über die aktuellen Fortschritte, können gemeinsam vor dieser Wand ihre Gedanken austauschen und gleichzeitig mit den Inhalten interagieren, z.B. durch Hinzufügen, Entfernen oder Umsortieren von Konstruktionsskizzen.

Weitere Anwendungsbereiche mit ähnlichen historisch gewachsenen Bedürfnissen finden sich zum Beispiel bei der Analyse von hochauflösenden Satellitenkarten in geowissenschaftlichen Instituten ([LambdaVision, 2007](#)) oder in der Medizin, z.B. für die Darstellung von Röntgenaufnahmen oder Computer-Tomographien ([Robert, 2006](#)).



Abb. 3.4: Powerwalls in der Automobilkonstruktion: Links für das Design eines Fahrzeuginnenraums ([Daimler-AG, 2007](#)) und daneben zur Visualisierung eines komplexen CAD-Modells ([Barco, 2007](#))

Die Arbeit und Interaktion mit großformatigen, senkrechten Flächen bzw. mit großformatigen Modellen ist demnach ein für den Menschen vertrautes Konzept, dass in vielen Anwendungsbereichen Verwendung findet. Ein Vergleich der Forschungsarbeiten zeigt, dass hier zwei Wege beschritten werden können, um sich diesen Umstand zunutze zu machen:

**Vorwärtsanalyse (Anwendungsbereich  $\Rightarrow$  Einsatz einer Powerwall):** Ausgehend von den Anforderungen des jeweiligen Anwendungsbereich werden die traditionellen Arbeitsmethodiken und Interaktionskonzepte für die Arbeit auf bzw. mit Powerwalls adaptiert. So werden nach [Buxton u. a. \(2000\)](#) z.B. bei nahezu allen modernen Automobilkonstruktoren zur Unterstützung bzw. Durchführung der oben gezeigten Vorgehensweisen Powerwalls eingesetzt, die in Abhängigkeit von den Anforderungen des Konstruktionsbereiches in physikalischer Größe und Umfang der Interaktionsmöglichkeiten variieren können (Abb. 3.4).

**Rückwärtsanalyse (Einsatz einer Powerwall  $\Rightarrow$  Anwendungsbereich):** Ausgehend von bestehenden Einsatz- und Nutzungsformen von Powerwalls werden bei dieser Vorgehensweise die kennzeichnenden Eigenschaften von Powerwalls analysiert. Das Ziel ist hierbei, die Einsatzmöglichkeit dieser Visualisierungssysteme für Anwendungsbereiche zu evaluieren, in denen nicht unbedingt ein „historisch gewachsenes“ Bedürfnis an der Verwendung von Powerwalls besteht.

Für einen effizienten Einsatz der Powerwall als gemeinsamer Arbeitsbereich im CCW wird in den folgenden Abschnitten dieses Unterkapitels nach dem Prinzip der Rückwärtsanalyse vorgegangen. Dazu wird hier untersucht, welche Nutzungsformen von Powerwalls grundsätzlich existieren (Abschnitt 3.2.1), welche kognitiven Vorteile sich bei deren Verwendung ergeben (Abschnitt 3.2.2) und welche Paradigmen bzw. Konzepte für eine effiziente Interaktion zur Anwendung gebracht werden können (Abschnitt 3.2.3). Abschließend werden in Abschnitt 3.2.4 die für diese Arbeit interessanten vorteilhaften Eigenschaften von Powerwalls zusammengefasst.

### 3.2.1 Klassifikation von Nutzungsformen

Als Grundlage für die Rückwärtsanalyse wurden verschiedene Nutzungsformen von Powerwalls untersucht, die aus unterschiedlichen Forschungsarbeiten und Projekten hervorgegangen sind. Der Schwerpunkt der Untersuchung lag dabei auf der Fragestellung, aus welchen Eigenschaften bei den jeweiligen Nutzungsformen Vorteile gezogen wurden. Darauf basierend wurde eine Klassifikation von Nutzungsformen gebildet, um die Einordnung und den Vergleich der vielfältigen Arbeiten zu ermöglichen und in den folgenden Ausführungen strukturiert darlegen zu können.

Bei der vergleichenden Betrachtung der Nutzungsformen wurden zwei unterschiedliche Merkmale identifiziert, nach denen eine Gruppierung möglich ist: der *Interaktionsgrad* und die *Form der Informationsdarstellung*. Beide Klassifikationen werden im Folgenden vorgestellt:

### Klassifikation nach Form der Informationsdarstellung

**Einzelne Anwendung:** Die gesamte Darstellungsfläche der Powerwall wird hierbei lediglich zur Anzeige einer einzelnen Anwendung eingesetzt. Ein Fensterkonzept ist hier nicht vorgesehen. Damit stehen nur Informationen zur Verfügung, die innerhalb der dargestellten Anwendung verfügbar sind. Je nach konzeptioneller Realisierung der Powerwall kann es sich dabei um eine Standardanwendung oder um eine speziell für die Powerwall entwickelte Anwendung handeln (vgl. Abschnitt 3.3.4). Diese Nutzungsform eignet sich dazu, spezielle Informationen sehr detailliert darzustellen und ist je nach Anwendungsbereich sowohl von Einzelpersonen als auch von Gruppen effizient einsetzbar.

**Desktopumgebung:** Hier wird eine komplette Desktopumgebung dargestellt, wie es auch an normalen Arbeitsplätzen üblich ist. Es können mehrere Anwendungen in verschiedenen Fenstern gleichzeitig betrieben werden. Damit steht eine Vielzahl von Informationen zur Verfügung, die je nach den Bedürfnissen der Betrachter aus mehreren Anwendungskontexten zusammengestellt werden können. Das Hauptaugenmerk dieser Nutzungsform liegt also auf der Ausnutzung der physikalischen Ausmaße und der hohen Auflösung, um einen großen Informationsraum verwenden zu können. Die Informationsvielfältigkeit impliziert zumeist, dass die Powerwall von einer Gruppe von Benutzern verwendet wird.

**Virtuelle Umgebung:** Bei dieser Nutzungsform wird die Darstellung mit der Powerwall als virtuelle Umgebung eingesetzt. Die Informationen sollen den Betrachtern hierbei durch die physikalischen Ausmaße und den damit verbundenen, nahezu vollständig ausgefüllten Sichtbereich „plastisch“ nahegebracht werden. Diese Nutzungsform ist insbesondere für Anwendungsbereiche sinnvoll, in denen die Darstellung von „realitätsgetreuen“ 3D-Modellen erforderlich ist, wie z.B. in der Automobilkonstruktion oder Fabrikplanung.

### Klassifikation nach Interaktionsgrad

**Keine Interaktion:** Die Powerwall wird hierbei nur zur Visualisierung eingesetzt. Es erfolgt keine unmittelbare Interaktion mit den dargestellten Inhalten. Tatsächlich ist diese Auffassung jedoch lediglich auf die explizite technologische Unterstützung der Interaktion zu beschränken. Wie in Abschnitt 3.2.3 noch im Detail dargestellt werden wird, ist mit jedem großen Anzeigemedium stets eine sogenannte *physische Navigation* möglich. Diese Nutzungsform ist für Anwendungsbereiche interessant, in denen großflächige und hochauflösende Anzeigemedien benötigt werden, die ausschließlich „zum Informieren“ einer größeren Benutzergruppe eingesetzt werden (z.B. im halb-öffentlichen oder öffentlichen Raum).

**Einfache Interaktion:** Die Möglichkeit zu einer einfachen Interaktion mit den dargestellten Inhalten ist die verbreitetste Nutzungsform einer Powerwall. Wie bei einem herkömmlichen Arbeitsplatzrechner werden die Informationen über ein Eingabegerät gesteuert. Informationen, wie z.B. digitale Dokumente oder Anwendungen, können angefordert, platziert,

bearbeitet und entfernt werden. Um diesen Interaktionsgrad zu ermöglichen können verschiedene Technologien zum Einsatz kommen. Gängige Technologien im Kontext einer Powerwall sind z.B. die Verwendung spezieller Zeigegeräte oder touch-basierter Anzeigemedien. Aber auch die Verwendung von „klassischen“ Eingabegeräten, d.h. Maus und Tastatur, ist nicht vollständig auszuschließen. Diese Nutzungsform eignet sich zur Bereitstellung eines skalierbaren Arbeitsbereiches und ist sowohl von Einzelpersonen als auch von Gruppen effizient einsetzbar.

**Komplexe Interaktion:** Die komplexe Interaktion ist eine Spezialisierung der einfachen Interaktion. Zusätzlich zu den zuvor dargestellten Eigenschaften sind bei dieser Nutzungsform Interaktions- und Arbeitstechniken integriert, die nur für einen spezifischen Anwendungsbereich erforderlich sind. Diese Nutzungsform findet sich daher in der Regel bei solchen Anwendungsbereichen, in denen der Bedarf an einer Powerwall „*historisch gewachsen*“ ist (vgl. Abschnitt 3.2).

Die beiden Klassifikationen verlaufen parallel und schließen sich nicht gegenseitig aus. Eine Powerwall kann somit eine *Desktopumgebung* darstellen und Möglichkeiten für eine *einfache Interaktion* zur Verfügung stellen.

Die hier dargestellten Klassifikationen sind als eine Zusammenfassung und grobe Übersicht über die möglichen Nutzungsformen von Powerwalls zu verstehen, die als Grundlage für die weitere Analyse in diesem Unterkapitel dienen.

### 3.2.2 Kognitive Vorteile von großen Anzeigemedien

Herkömmliche Monitore nehmen nur einen geringen Teil des natürlichen Sichtfeldes eines Menschen (ohne Sehbeeinträchtigung) ein. Mit großformatigen Visualisierungssystemen ist es dagegen möglich einen sehr großen, teilweise sogar den vollständigen Bereich des Sichtfeldes abzudecken. In Abbildung 3.5 wird schematisch dargestellt, welche Bereiche des menschlichen Sichtfeldes durch verschiedene exemplarische Visualisierungssysteme ausgefüllt werden.

Die Tatsache, dass große Anzeigemedien einen umfassenden Teil des Sichtfeldes beanspruchen, führt zu der Frage, welche Auswirkungen das auf die Aufnahme und Verarbeitung von Informationen hat. Dieser Fragestellung wurde in zahlreichen Forschungsbeiträgen nachgegangen. Dabei wurden insbesondere die folgenden kognitiven Vorteile identifiziert:

**Unterstützung des Erinnerungsvermögen:** Der erhöhte Wahrnehmungsraum führt nach Untersuchungen von Czerwinski u. a. (2003) dazu, dass Benutzer sich besser an Informationen erinnern können, wenn sie diese über große und sichtfeldeinnehmende Anzeigemedien aufgenommen haben. Nach Tan u. a. (2004) ist dies auf das „Gefühl der Gegenwart“ (sense of presence) zurückzuführen. Die Benutzer nehmen sich selbst als Teil des Informationsraums wahr: Informationen werden nicht mehr „von außen“ betrachtet, sondern stellen Objekte in einem Informationsraum dar, die eine bestimmte Position in

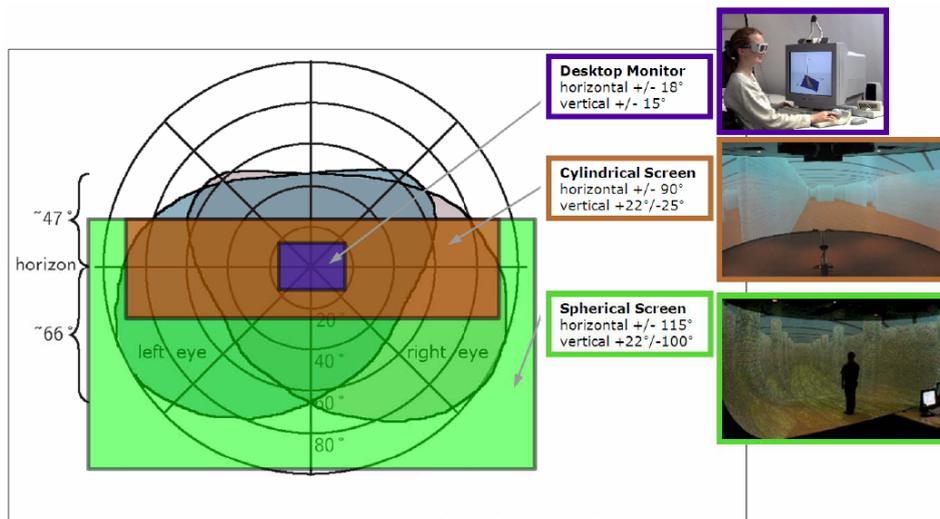


Abb. 3.5: Vergleich verschiedener Visualisierungssysteme im natürlichen Sichtfeld (Campos u. a., 2007)

Relation zu dem Benutzer haben. Dies führt dazu, dass nicht nur die konkreten Inhalte von Informationen, sondern auch ihre Darstellungsform und relative Position zu den Benutzern in *kognitiven Karten*<sup>3</sup> „gespeichert“ werden.

**Ausnutzung der peripheren Sicht:** Das menschliche Sichtfeld lässt sich grob in zwei Bereiche einteilen: die *zentrierte Sicht* und die *periphere Sicht*. Die zentrierte Sicht bezeichnet die Fokussierung und bewusste Wahrnehmung eines Objekts bzw. eines Bildausschnitts. In diesem Bereich hat das menschliche Auge die höchste Auflösung und Farbdifferenzierung. Die periphere Sicht bezieht sich auf den Bereich außerhalb der zentrierten Sicht. Die Unterscheidung von Farben und detaillierten Formen ist in diesem Bereich zwar geringer als bei der zentrierten Sicht, bietet den Menschen jedoch die Möglichkeit, Objekte und Bewegungsabläufe *unterbewusst* wahrnehmen und interpretieren zu können. Bei der Arbeit mit kleinen Bildschirmen nutzt man ausschließlich die zentrierte Sicht und damit nur die bewusste Wahrnehmung. Die Verwendung von großen Bildschirmen führt nach Grudin (2001) dazu, dass sowohl die zentrierte als auch die periphere Sicht eingesetzt werden. Durch dieses natürliche Sichtfeld lässt sich somit auch die unbewusste Wahrnehmung von Informationen ausnutzen (Czerwinski u. a., 2003).

**Unterstützung der räumlichen Orientierung:** Auf Basis der beiden zuvor genannten kognitiven Vorteile, wurden Untersuchungen durchgeführt, wie sich der Einsatz großer Anzeigemedien auf die Orientierung und Navigation in dreidimensionalen virtuellen Umgebungen auswirkt. Arbeiten, wie z.B. von Bakdash u. a. (2006) oder Tan u. a. (2006), zeigen in diesem Zusammenhang auf, dass den Benutzern durch das erweiterte Sichtfeld die räumliche Orientierung in solchen Umgebungen leichter fällt, als bei kleineren

<sup>3</sup>Als kognitive Karte bezeichnet man die mentale Repräsentation eines geographischen Raumes oder räumlich vorstellbarer logischer Zusammenhänge.

Anzeigemedien. Dies liegt unter anderem daran, dass die Orientierung und Navigation der Benutzer nicht nur *virtuell*, sondern auch *physisch* stattfindet (Ball und North, 2005). Das bedeutet, dass die Benutzer auch durch den Einsatz ihres Körpers Teilbereiche der virtuellen Umgebung erforschen können. Beispiele dafür sind das Drehen des Kopfes, um weiter nach links oder nach rechts zu schauen, sowie das Heran- und Zurücktreteten, um bestimmte Bereiche der virtuellen Umgebung zu vergrößern bzw. zu verkleinern. Guy u. a. (2005) zeigen auf, dass die räumliche Orientierung zusätzlich erhöht werden kann, wenn die Darstellung und Betrachtung der virtuellen Umgebung aus unterschiedlichen Perspektiven erfolgt.

Der tatsächliche Wirkungsgrad der hier gezeigten kognitiven Vorteile ist von einer Reihe von Faktoren abhängig. Neben den technischen Aspekten der Visualisierungssysteme (z.B. die physikalischen Ausmaße der Darstellungsfläche und die Bildauflösungen) spielen nach Tyndiuk u. a. (2005) insbesondere auch *die Beschaffenheit der zu bewältigenden Aufgaben* und *die kognitiven Profile* der Benutzer eine wesentliche Rolle. So wird z.B. die Unterstützung der räumlichen Orientierung und die Interaktion in einer virtuellen Umgebung wirksamer, wenn die Aufgaben komplex sind und die Benutzer generell eine niedrige *visuelle Aufmerksamkeit* haben. Die visuelle Aufmerksamkeit bezeichnet dabei die Fähigkeit, Informationen visuell wahrzunehmen und kognitiv zu verarbeiten.

Ein einfacher Test, um die visuelle Aufmerksamkeit eines Menschen zu quantifizieren, ist z.B. der *Test for Attentional Performance* (TAP) von Zimmermann und Fimm (1994). In diesem wird die Zeit gemessen, die ein Mensch benötigt, um ein bestimmtes Symbol innerhalb einer Menge von ähnlichen Symbolen zu identifizieren. Niedrige Zeiten sind ein Maß für eine hohe visuelle Aufmerksamkeit, während hohe Zeiten einer niedrigen visuellen Aufmerksamkeit entsprechen.

### 3.2.3 Interaktionskonzepte

In den letzten Jahren wurden in einer Reihe von Forschungsarbeiten verschiedene Ansätze zur Gestaltung der Interaktion mit großen und hochauflösenden Anzeigemedien untersucht. Dabei ist auffällig, dass es oftmals keine einheitliche Vorstellung dessen gibt, welche Aufgaben bzw. Prozesse unter Interaktion zu verstehen sind. Um dieser Uneinheitlichkeit im Rahmen dieser Arbeit vorzubeugen, wird der Begriff „Interaktion“ im Zusammenhang mit Powerwalls in den weiteren Ausführungen in Anlehnung an Bowman (1999) in zwei grundlegende Aufgaben aufgeteilt:

1. Navigation durch bzw. über Inhalte
2. Selektion und Manipulation von Inhalten

Wie in der Einleitung zu diesem Unterkapitel beschrieben, entspringen Powerwalls einem den Menschen vertrauten Konzept: der Arbeit und Interaktion mit großformatigen, senkrechten Flächen. Das Bestreben der Forschung ist daher, natürliche und intuitive Interaktionstechniken bei der Verwendung von Powerwalls einzusetzen. Aufgrund der sehr großen Darstellungsfläche,

der Vielzahl von Informationen und des oftmals kollaborativen Einsatzgebietes ist die „klassische“ direkte Interaktion über Maus und Tastatur nicht mehr möglich bzw. sinnvoll. Hier sind andere Konzepte erforderlich, die für die Interaktion mit solchen Anzeigemedien optimiert sind. Die wesentlichen Herausforderungen, die es dabei zu bewältigen gibt, wurden von [Ni u. a. \(2006\)](#) zusammengefasst:

**Erreichbarkeit entfernter Ziele:** Mit den großen physikalischen Ausmaßen einer Powerwall ist es für die Benutzer mitunter sehr schwierig, Objekte an „entlegenden“ Stellen der Anzeigefläche zu erreichen ([Robertson u. a., 2005](#)). Dies ist insbesondere dann der Fall, wenn aus nächster Nähe mit dem Medium gearbeitet wird.

**Verfolgung des Eingabezeigers:** Bei der Steuerung mit herkömmlichen Eingabegeräten, wie z.B. der Maus, muss die Beschleunigung des Eingabezeigers erhöht werden, um größere Entfernungen auf hochauflösenden Anzeigemedien zu überbrücken. Mit zunehmender Geschwindigkeit ist es für Benutzer jedoch schwierig, den Eingabezeiger stets mitverfolgen zu können. Gleichmaßen kann es sehr problematisch sein, einen unbewegten Eingabezeiger auf einer Powerwall zu lokalisieren, z.B. wenn durch dessen Positionierung die Aufmerksamkeit der Betrachter auf etwas gelenkt werden soll ([Khan u. a., 2005](#)).

**Kontextwechsel und Layoutmanagement:** Insbesondere bei der Nutzung von fensterbasierten Anwendungen wirft die Interaktion mit hochauflösenden Anzeigemedien eine Reihe von Fragen bezüglich eines geeigneten Layouts für die Darstellungsfläche auf. So werden z.B. durch die Größe der zur Verfügung stehenden Fläche klassische Überlappungen von Fenstern weniger relevant. Gleichzeitig liegen viele Informationen bei der Arbeit aus nächster Nähe im peripheren Sichtfeld (vgl. [3.2.2](#)). Hier sind alternative Konzepte zum effizienten Kontextwechsel notwendig ([Bezerianos und Balakrishnan, 2005](#)).

**Übergang zwischen Interaktionen:** Die Interaktion kann in zwei breitere Paradigmen kategorisiert werden. Für Aufgaben, die die Verwendung von detaillierten Informationen erfordern, ist die Arbeit aus nächster Nähe zur Powerwall sinnvoll. Andere Aufgaben sind besser aus der Entfernung zu bewältigen, wie z.B. die Sortierung von Dokumenten oder die Präsentation von großformatigen Zeichnungen ([Vogel und Balakrishnan, 2005](#)). Für beide Paradigmen sind unterschiedliche Interaktionskonzepte erforderlich, die sich im Idealfall miteinander kombinieren lassen und einen fließenden Übergang zwischen der Arbeit aus nächster Nähe sowie aus der Entfernung ermöglichen.

Wie aus dem letzten Punkt hervorgeht, ist bei der Gestaltung von konkreten Interaktionskonzepten oftmals die Frage entscheidend, aus welcher Entfernung die Powerwall genutzt werden soll. Im Folgenden werden daher zunächst Interaktionsebenen und Nutzungsbereiche von Powerwalls eingeführt. Darauf basierend werden im Anschluss daran verschiedene Konzepte zur Interaktion vorgestellt.

### Interaktionsebenen und Kontextbereiche

Es existiert kein allgemeingültiges Interaktionskonzept für große, hochauflösende Anzeigemedien. Die Anforderungen an die Interaktion stehen beispielsweise in Abhängigkeit von den Anforderungen des jeweiligen Einsatzgebietes, der Nutzungsform oder der Anzahl der Benutzer, die gleichzeitig Zugriff bzw. Einsicht auf die Inhalte des Systems haben.

Um dieser Vielfältigkeit Ausdruck zu verleihen, werden im Rahmen einiger Arbeiten Interaktionsebenen eingeführt, um verschiedene Ausprägungen von Interaktionsmöglichkeiten zu gliedern. [Streitz u. a. \(2003\)](#) haben beispielsweise ein 3-Zonen-Modell für den Einsatz in privaten oder halb-öffentlichen Umgebungen vorgeschlagen. Der zur Verfügung stehende (physikalische) Raum vor dem Visualisierungssystem wird dabei konzeptionell in drei Zonen bzw. Ebenen eingeteilt: *Ambient Zone*, *Notification Zone* und *Interactive Zone* (siehe [Abb. 3.6](#), links).

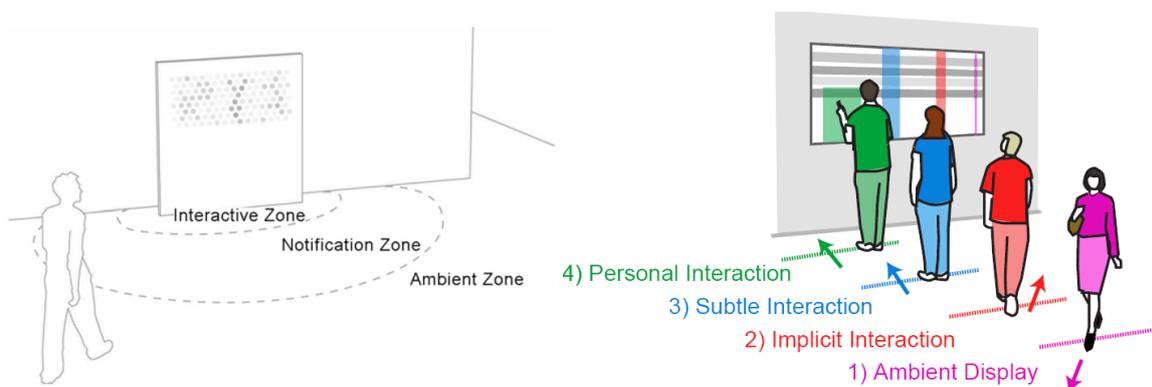


Abb. 3.6: Interaktionsebenen nach [Streitz u. a. \(2003\)](#) (links) und nach [Vogel und Balakrishnan \(2004\)](#) (rechts)

Die Übergänge zwischen den einzelnen Ebenen sind als fließend zu betrachten. In der *Interactive Zone* findet eine Interaktion aus nächster Nähe statt. Hier sind Detailinformationen einsehbar und die Benutzer agieren unmittelbar mit der Oberfläche des Anzeigemediums. In der *Notification Zone* und in der *Ambient Zone* erfolgt die Interaktion aus der Distanz. Während die Interaktion in der *Notification Zone* noch „aktiv“ ist, d.h. Benutzer selektieren und manipulieren Inhalte, ist die Interaktion in der *Ambient Zone* „passiv“. Hier erfolgt keine direkte Manipulation von Inhalten. Diese Ebene dient zur Neuorientierung bzw. zur Verschaffung eines Gesamtüberblicks über alle zur Verfügung stehenden Inhalte.

[Vogel und Balakrishnan \(2004\)](#) erweitern dieses Modell für den Einsatz in öffentlichen Umgebungen und setzen dabei auf vier Ebenen ([Abb. 3.6](#), rechts). Die grundsätzliche Argumentation für die Nutzung der Ebenen ähnelt jedoch den Ausführungen von [Streitz u. a. \(2003\)](#). Zum einen beschreiben die Ebenen für den Benutzer einen Kontextwechsel, d.h. von dem Gesamtüberblick zur Detailansicht und umgekehrt. Zum anderen implizieren die Ebenen unterschiedliche Interaktionskonzepte für die Arbeit mit der Powerwall *aus nächster Nähe* und *aus größerer Entfernung*.

Die Interaktionsebenen dienen also sowohl zur Beschreibung des Informationskontextes, in dem sich die Benutzer befinden (*Detail* vs. *Global*), als auch zur Beschreibung der Interaktionsart (*Close-range* vs. *Distant*). Weitere Arbeiten in diesem Bereich, wie z.B. [Bezerianos und Balakrishnan \(2005\)](#) oder [Robertson u. a. \(2005\)](#), zeigen ähnliche Ergebnisse auf.

Auf Basis dieser Untersuchungen wurde für diese Arbeit eine Generalisierung der verschiedenen Modelle für Interaktionsebenen erstellt. Dabei werden Interaktionsebenen (*Close-range Interaction* und *Distant Interaction*) und Kontextbereiche (*Detail Context*, *Context Switch* und *Global Context*) getrennt betrachtet (siehe Abb. 3.7).

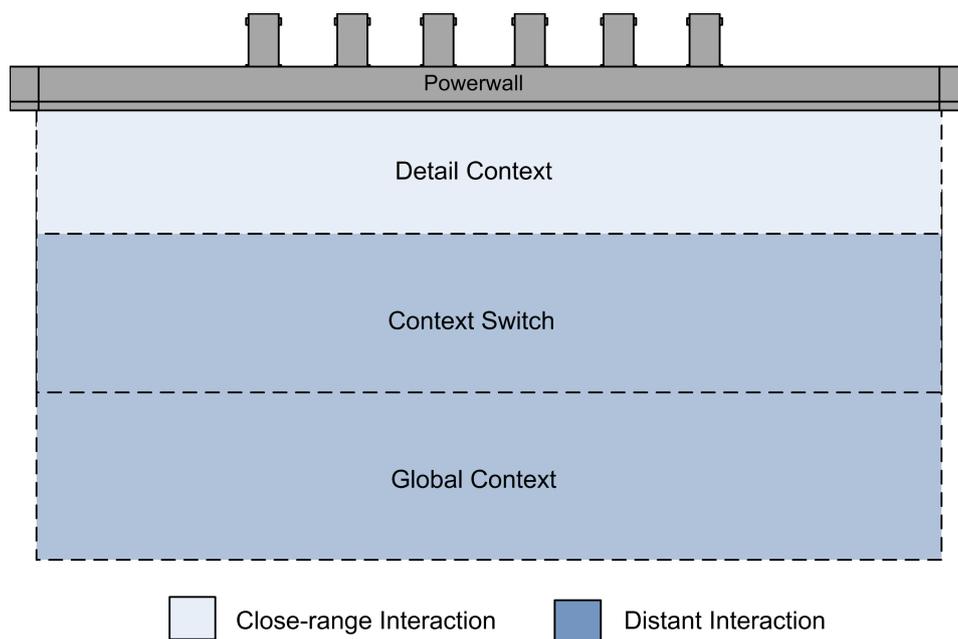


Abb. 3.7: Generalisiertes Modell für Interaktionsebenen und Nutzungsbereiche

Im Folgenden wird zunächst die Benutzung der Kontextbereiche beschrieben. Anschließend werden einige Beispiele für die *Close-range* und die *Distant Interaction* aus bestehenden Arbeiten vorgestellt.

### Zoom und Kontextwechsel

Bei der Verwendung von klassischen Arbeitsumgebungen, d.h. mit kleinen Anzeigemedien, gibt es im Wesentlichen zwei Funktionen, um über die dargestellten Informationen zu navigieren: *Zoom* und *Scroll*. Die Zoom-Funktion ermöglicht es, Informationen nach Bedarf zu vergrößern bzw. zu verkleinern. Mit der Scroll-Funktion kann die Bildschirmansicht in eine beliebige Richtung verschoben werden, um Informationen, die außerhalb des aktuellen Darstellungsbereichs liegen, in den Benutzungskontext zu holen.

Ball und North (2005) haben durch eine Reihe von Untersuchungen gezeigt, dass Benutzer bei der Verwendung großer Anzeigemedien von solchen Funktionen in der Regel seltener Gebrauch machen. Sie setzen intuitiv auf eine *physische Navigation*. Die Vergrößerung von Informationen bis zur Erkennung von feingranularen Details erfolgt durch das Herantreten an das Medium (*Detail Context* in Abb. 3.7). Durch das Zurücktreten von dem Anzeigemedium werden die Informationen bis zu einer globalen Gesamtansicht verkleinert (*Global Context* in Abb. 3.7).

Eine ähnliche Vorgehensweise erfolgt entsprechend für die klassische Scroll-Funktion, d.h. der Änderung des Benutzungskontextes. In der Regel arbeiten die Benutzer entweder im *Global Context* (z.B. bei Gruppenbesprechungen) oder unmittelbar an der Powerwall, d.h. im *Detail Context*. Um im letzteren Fall den Benutzungskontext zu ändern, um z.B. mit anderen Informationen auf der Darstellungsfläche zu arbeiten, gehen die Benutzer „einen Schritt zurück“ (*Context Switch* in Abb. 3.7). Hier erfolgt eine kurze Neuorientierung und Lokalisierung von Informationen, um anschließend entweder in der detaillierten bzw. in der globalen Sicht weiterzuarbeiten.

### Close-range Interaction

Die Arbeit direkt am bzw. mit dem Medium ist für Benutzer eine „natürliche“ Form der Interaktion. Sie entspringt vertrauten Konzepten, z.B. der Arbeit mit Pinnwänden oder Whiteboards. Möchte man bei Pinnwänden beispielsweise ein Dokument umordnen, dann greift man es mit den Händen und hängt es an anderer Stelle wieder auf.

Um eine analoge Vorgehensweise auch auf Anzeigemedien zu ermöglichen, werden in der Regel berührungsempfindliche Systeme (Touchscreens) eingesetzt. Mit diesen ist es möglich, wie in der Realität zu agieren (Knöpfle, 2007). Je nach Realisierung erfolgt die Interaktion direkt mit den Händen oder mit speziellen Stiften auf der Oberfläche des Anzeigemediums.

### Distant Interaction

Die Interaktion aus der Distanz ist nach Knöpfle (2007) eine „intuitive“ Form der Interaktion. Man agiert nicht mehr nach natürlichen Konzepten, verwendet aber Techniken die plausibel erscheinen. Zur Veranschaulichung lässt sich das oben genannte Beispiel mit der Pinnwand erneut aufgreifen. Das zu verschiebende Dokument wird nicht mehr mit den Händen gegriffen. Stattdessen deutet man auf das Dokument und anschließend auf die Position, an die es verschoben werden soll.

Hierfür kommen in der Regel spezielle Zeigegeräte zum Einsatz. Die Selektion eines Objekts erfolgt auf diese Weise z.B. durch das „Zeigen auf das Objekt“. Auch hier unterscheiden sich die konkreten Realisierungen. Ballagas u. a. (2005) und Jiang u. a. (2006) setzen beispielsweise Mobiltelefone mit integrierten Kameras als Zeigegeräte ein. Diese fungieren als Ersatz für eine optische Maus. Die aktuelle Position wird dabei über spezielle Bildmusteralgorithmen

berechnet. Andere Realisierungen setzen auf eine Kombination von Infrarot und Kameraerkennung. Bei [König u. a. \(2007\)](#) werden Kameras an den Rändern der Powerwall installiert und auf die Benutzer ausgerichtet. Die Interaktion erfolgt dann mit einem leicht modifizierten Laserpointer. Auf Basis der Kamerabilder wird die aktuelle Zeigerichtung des Laserpointers berechnet und entsprechend auf das Anzeigesystem übertragen. Das von [Fischer \(2007\)](#) entwickelte Zeigergerät funktioniert genau in die umgekehrte Richtung. Dort werden Infrarotdioden an der Powerwall angebracht und in der Hand tragen die Benutzer eine Kamera. Die zugrundeliegenden Berechnungsprinzipien ähneln dem System von [König u. a. \(2007\)](#). Auf Basis der Kamerabilder wird die aktuelle Zeigerichtung bestimmt.

Neben der Verwendung von Zeigergeräten gibt es weitere Konzepte für die entfernte Interaktion. So lassen sich Selektionen und Manipulationen z.B. mit Gesten und Bewegungen ([Ahn u. a., 2004](#)) oder unter Berücksichtigung der Benutzerposition ([Vogel und Balakrishnan, 2004](#)) durchführen. Diese Konzepte bringen jedoch in der Regel einen erhöhten Lernaufwand mit sich, um z.B. die Auswirkungen bestimmter Gesten oder Bewegungen zu erlernen. Dadurch ist es oftmals sehr schwierig, die Intuitivität der Interaktion für diese Konzepte zu gewährleisten.

### 3.2.4 Fazit

Für die Konzeption des in dieser Arbeit geplanten Systems ist es wichtig, die Vorteile zu kennen, die sich durch den Einsatz einer Powerwall ergeben. Daher werden in diesem Abschnitt die wesentlichen vorteilhaften Eigenschaften von Powerwalls zusammengefasst, die in der zuvor dargelegten Analyse identifiziert werden konnten.

**Realitätsgetreue Auflösung:** Mit kleinen Anzeigemedien können bestimmte Daten und Informationen, wie z.B. Satellitenkarten oder 3D-Modelle, nur sehr detailarm wiedergegeben werden. Dies liegt unter anderem daran, dass die Auflösung bei normaler Distanz zum Bildschirm im Durchschnitt nur einem Drittel des Auflösungsvermögens des menschlichen Auges entspricht ([Knöpfle, 2007](#)). Mit dem Einsatz einer (nahezu) beliebig skalierbaren Powerwall, lassen sich Auflösungen erreichen, die oberhalb des Auflösungsvermögens des menschlichen Auges liegen. Damit sind sehr detaillierte und realitätsgetreue Visualisierungen möglich.

**Kognitive Vorteile:** Die Auflösung und die physikalische Größe einer Powerwall führen zu einem sehr großen Informationsraum. Die Möglichkeiten des menschlichen Sichtfeldes (zentrierte und periphere Sicht) lassen sich komplett ausnutzen. Informationen können dabei nicht nur bewusst, sondern auch unbewusst aufgenommen werden. Dies kann zur Unterstützung des Erinnerungsvermögens und der räumlichen Orientierung in dreidimensionalen Umgebungen verwendet werden.

**Natürliche und intuitive Interaktion:** Der Einsatz einer großformatigen und senkrecht stehenden Arbeitsfläche ist für Menschen ein vertrautes Interaktionskonzept (z.B. Whiteboard, Flipchart oder Pinnwand). Für die Interaktion können natürliche und intuitive Konzepte eingesetzt werden.

**Eignung für Gruppen:** Durch den großen Informationsraum sind Powerwalls sehr gut für die Gruppenarbeit geeignet. Daten und Informationen können für alle sichtbar miteinander geteilt und/oder gleichzeitig bearbeitet werden. Die physikalischen Ausmaße führen dazu, dass sich die Gruppenmitglieder nicht gegenseitig behindern oder das Sichtfeld einschränken.

**Globale und detaillierte Sichtweise:** Die physikalische Größe und die hohe Auslösung ermöglichen es, dass Powerwalls gleichzeitig eine globale und auch detaillierte Sichtweise bieten können. Durch die physische Navigation bei großen Anzeigemedien, können Benutzer beliebig in den Informationsraum „hineinzoomen“ (d.h. herangehen) und „herauszoomen“ (d.h. zurücktreten).

**Beliebige Skalierbarkeit der Größe:** Theoretisch lassen sich die Ausmaße von Powerwalls beliebig skalieren, wenn sie mit geeigneten Verteilungsstrategien und parallelen Bildberechnungsverfahren realisiert wurden (vgl. Abschnitt 2.2).

**Realisierung mit Standardkomponenten:** Während vor einigen Jahren noch spezielle Grafik-Hardware für die Realisierung von Powerwalls benötigt wurde, ist dies heutzutage mit Standardkomponenten zu bewerkstelligen.

Diese Aufstellung stellt keinen Anspruch auf Vollständigkeit. Sie basiert lediglich auf den Forschungsarbeiten und Projekten, die im Rahmen der Analyse in diesem Unterkapitel betrachtet wurden. Es ist nicht auszuschließen, dass auf Basis anderer Arbeiten weitere Eigenschaften identifizierbar sind, die sich beim Einsatz von Powerwalls als vorteilhaft erweisen können.

### 3.3 Strategien und Verfahren zur skalierbaren Visualisierung

In den Grundlagen dieser Arbeit wurden in Unterkapitel 2.2 verschiedene Ausprägungen skalierbarer Visualisierungssysteme vorgestellt. Da der zu konzipierende Trainingsraum einen gemeinsamen Arbeitsbereich in Form einer Powerwall aufweisen soll, wurden im vorangegangenen Unterkapitel 3.2 die Vorteile analysiert, die sich beim Einsatz dieser Ausprägung eines Visualisierungssystems ergeben.

Bislang ungeklärt blieb die Frage, mit welchen Strategien und Verfahren die Visualisierung von Anwendungen skalierbar gemacht werden können. Diese Frage wird in diesem Unterkapitel nachgegangen. Zunächst werden dazu in Abschnitt 3.3.1 allgemeine Strategien zur Synchronisation von Daten vorgestellt. In den darauffolgenden Abschnitten werden mit der Eingabesynchronisation (3.3.2) und der Szenensynchronisation (3.3.3) zwei dieser Strategien spezifischer betrachtet. Für jede Synchronisationsstrategie werden verschiedene Verfahren vorgestellt, die für die Konzeption eines skalierbaren Visualisierungssystems eingesetzt werden können. Abschließend werden in Abschnitt 3.3.4 zwei grundlegende Ansätze vorgestellt, nach denen sich die hier aufgezeigten Strategien und Verfahren in konkreten Realisierungen umsetzen lassen.

### 3.3.1 Strategien zur Synchronisation von Daten

Allen Systemen mit skalierbarer Visualisierung liegt zugrunde, dass eine parallele Berechnung von 3D-Szenen in einem Cluster von Rechnern erforderlich ist. Die einzelnen Knoten des Clusters müssen hierbei so miteinander synchronisiert werden, dass alle Knoten gemeinsam aus der gleichen Datenbasis ein konsistentes Bild berechnen. Nach Roth (2005) müssen für eine solche Synchronisation *Informationseinheiten* zwischen den parallelen Instanzen ausgetauscht werden. In Bezug auf eine (maximale) Skalierbarkeit muss für jedes verteilte Visualisierungssystem geklärt werden, welche Informationseinheiten zur Synchronisation verwendet werden und wie sich diese Informationseinheiten zwischen den verteilten Instanzen austauschen lassen.

In klassischen Anwendungen, die nur auf einer einzelnen Workstation laufen, werden Eingaben eines Benutzers mit Interaktionsgeräten über die Benutzeroberfläche durchgeführt. Dabei entstehen Eingabeereignisse aus der die Anwendung neue Primitiven für die Anzeige generiert (z.B. neuer Text oder ein neues 3D-Objekt). Das Grafik-Subsystem rasterisiert diese Primitiven in einem sogenannten *Framebuffer*<sup>4</sup>, der dann auf einem Anzeigemedium ausgegeben wird.

Die Parallelisierung solcher Applikationen wurde von Chen u. a. (2001a) untersucht. Der Schwerpunkt lag dabei auf den Fragestellungen, wie die Daten zwischen den Komponenten in einem verteilten Visualisierungssystem verfügbar gemacht werden und wie die Ausführung der parallelen Instanzen synchronisiert werden kann. Die Anwendungen werden dazu konzeptionell in eine Client-/Server-Architektur übertragen. Der Client-Teil dient als Interaktionsschnittstelle für die Benutzer, während der Server-Teil für die Ansteuerung und Verwaltung der Anzeigemedien zuständig ist. Mehrere Instanzen beider Teile werden parallel ausgeführt. Die Art der benötigten Informationseinheiten zwischen den parallel laufenden Instanzen ist von dem Zeitpunkt der Synchronisation abhängig. Chen u. a. (2001a) identifizieren dabei drei grundsätzliche Möglichkeiten: Steuerelemente, Primitive und Pixel (siehe Abb. 3.8).

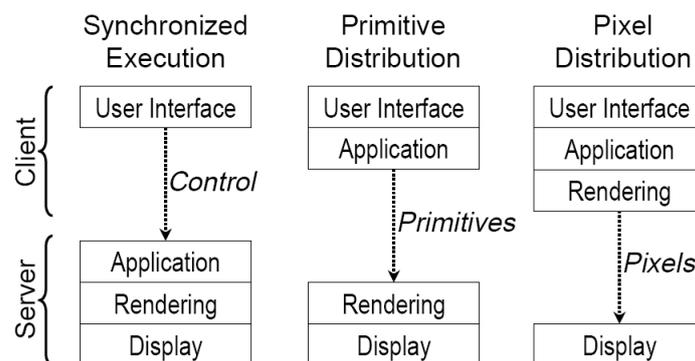


Abb. 3.8: Informationseinheiten bei verschiedenen Synchronisationsstrategien in einem verteilten Visualisierungssystem (Roth, 2005)

<sup>4</sup>zu dt.: Bildspeicher; Der Framebuffer bezeichnet ein Teil des Grafikspeichers von Computern und entspricht einer digitalen Kopie des Monitorbildes.

Darauf basierend schlägt [Roth \(2005\)](#) drei verschiedene Strategien für die Verteilung und Synchronisation der Bildberechnungsdaten in einem Visualisierungssystem vor:

**Eingabesynchronisation:** Auf allen Rechnern im Cluster wird die gleiche Applikation ausgeführt. Die Eingaben des Anwenders werden an alle Instanzen der Applikation propagiert. Solange alle Eingaben in der gleichen Reihenfolge empfangen werden und unter der Voraussetzung, dass der interne Zustand einer Applikation nur durch externe Eingaben bestimmt wird, befinden sich alle parallelen Instanzen in einem identischen Zustand. Diese Form der Synchronisation erfordert nur einen sehr geringen Kommunikationsaufwand und skaliert damit sehr gut. Nach [Roth \(2005\)](#) wird die Eingabesynchronisation häufig in fehlertoleranten Systemen verwendet, um mehrfach vorhandene Instanzen konsistent zu halten. Auch in kollaborativen *Virtual-Reality-Systemen* werden häufig nur Ereignisse, die durch Eingaben ausgelöst werden, synchronisiert. Konkrete Umsetzungen dieser Strategie finden sich u.a. in [Chen u. a. \(2001b\)](#) und [McCahill und Lombardi \(2004\)](#).

**Szenensynchronisation:** Bei dieser Strategie wird die Beschreibung der 3D-Szene zwischen den Instanzen synchronisiert. Es können sowohl grafische Primitive als auch Objekte eines Szenengraphen synchronisiert werden. Ein Szenengraph ist eine objektorientierte Datenstruktur mit der die logische und räumliche Anordnung der darzustellenden zwei- oder dreidimensionalen Szene sehr effizient beschrieben werden kann. Beispiele für die Verteilung von grafischen Primitiven sind *WireGL* ([Humphreys u. a., 2001](#)) und dessen Nachfolger *Chromium* ([Humphreys u. a., 2002](#)). Die Synchronisation über primitive Objekte des Szenengraphen wird u.a. bei den Szenengraphsystemen *Open Inventor* ([OI, 2006](#)) oder *OpenSceneGraph* ([OSG, 2007](#)) eingesetzt.

**Bildsynchronisation:** Bei dieser Synchronisationsstrategie wird lediglich die Anzeige parallelisiert und zwischen den parallelen Instanzen verteilt. Die Client-Applikation führt hier die komplette Bildberechnung durch. Anschließend wird das fertige Bild aufgeteilt und über das Netzwerk an die Server-Rechner gesendet, die die Ausgabemedien verwalten. Ein Beispiel dafür ist das *Distributed Multihead X Project* ([XDMX, 2003](#)) in Kombination mit der X-Server Erweiterung *Xinerama*. Nähere Informationen zum Einsatz dieser Technologien sind in den Vorarbeiten zu dieser Arbeit in [Napitupulu \(2007\)](#) verfügbar.

Die Voraussetzung für den Einsatz aller Strategien ist, dass die parallel ablaufenden Programmteile auf eine gemeinsame und konsistente Datenbasis zugreifen müssen. Zudem stellt [Roth \(2005\)](#) fest, dass unabhängig von der Art der eingesetzten Synchronisation ein schnelles Kommunikationsmedium unerlässlich ist.

Nach [Chen u. a. \(2001a\)](#) ist die Auswahl des Synchronisationsverfahrens von den Anforderungen an das Visualisierungssystem bzw. von den Eigenschaften der Applikationen, die mit diesem betrieben werden sollen, abhängig. Wenn beispielsweise ein Partikelsystem für Forschungen im Bereich der Astronomie visualisiert werden soll, so muss bei der Eingabesynchronisation lediglich die Zeit zwischen den parallel betriebenen Partikelsimulationssystemen synchronisiert werden. Bei der Szenensynchronisation müssen hingegen die genauen Positionen aller Partikel übertragen werden. Dabei würde ein wesentlich höherer Kommunikationsaufwand zu verzeichnen sein. Demgegenüber muss jedoch bei der Eingabesynchronisation

die Simulation von allen parallelen Instanzen durchgeführt werden. Dies führt zu einer deutlich höheren Belastung der insgesamt verfügbaren Ressourcen.

Für die parallele Bildberechnung in einem verteilten Visualisierungssystem ist die Bildsynchronisation nur bedingt geeignet, weil mit dieser nur die Darstellung parallelisiert wird. Die eigentliche Bildberechnung erfolgt weiterhin sequentiell. Diese Strategie skaliert bereits für zweidimensionale Anwendungen nicht sehr gut. Für dreidimensionale Anwendungen ist sie gar nicht einsetzbar. Aus diesem Grund werden zur Synchronisation in der Regel Eingabe- oder Szenensynchronisation eingesetzt. Auch hybride Formen dieser beiden Strategien sind möglich.

### 3.3.2 Verfahren für die Eingabesynchronisation

Die Verfahren für die Eingabesynchronisation basieren auf dem *Synchronized Execution Model* (Chen u. a., 2001b). Die Grundidee ist hierbei, mehrere Instanzen derselben Anwendung auf den Knoten im Cluster laufen zu lassen, die für die Ansteuerung der Anzeigemedien zuständig sind. Die Ausführung dieser Instanzen werden ab einer festgelegten Synchronisationsgrenze (synchronization level) abgeglichen. Innerhalb dieser Grenzen müssen die Instanzen identische Handlungsweisen annehmen.

Für die Definition der Systemgrenze, wird eine Anwendung konzeptionell in zwei Komponenten aufgeteilt: *Szenenverwaltung* (scene management) und *Szenenberechnung* (scene rendering). Die Szenenverwaltung wird auf allen Rechnern im Cluster identisch repliziert. Sie reagiert auf die Eingaben in der Systemumgebung der Anwendung, wie z.B. Dateizugriffe oder Tastatur- bzw. Mauseingaben, und passt ihr internes Verhalten entsprechend an. Die Eingaben werden mit jeder parallel laufenden Instanz synchronisiert. Die Szenenberechnung erhält darauf von der Szenenverwaltung sogenannte *Szenenbeschreibungen*, auf deren Basis dann Grafik-Primitiven zur Ausgabe auf einem Anzeigemedium generiert werden. Die Abbildung 3.9 zeigt eine Übersicht des Zusammenspiels zwischen den Komponenten.

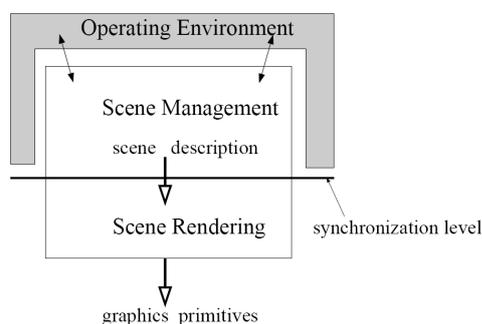


Abb. 3.9: Synchronized Execution Model nach Chen u. a. (2001b)

Infolge der Synchronisation erzeugen alle Instanzen dieselben Szenenbeschreibungen. Hierbei lassen sich zwei Szenarien unterscheiden: die *vollständige Replikation* und die *partielle Replikation*. Bei der vollständigen Replikation wird auch die Komponente Szenenberechnung

auf allen Rechnern identisch repliziert. Bei den Szenenbeschreibungen handelt es sich hier um identische Grafik-Primitiven (z.B. OpenGL-Instruktionen), mit denen jeder Knoten im Rechnercluster alle Objekte der Szene in der Bildberechnung berücksichtigen muss. Bei der partiellen Replikation können sich die parallel laufenden Instanzen der Szenenberechnung unterscheiden und zusätzliche Informationen mit sich tragen, wie z.B. eine Szenengraph-Datenstruktur. Die Szenenbeschreibung erfolgt dann auf einem höheren Abstraktionsniveau (z.B. Manipulationsoperatoren auf Szenengraphen), so dass die Knoten z.B. nur Objekte in einem vorher zugeteilten Teilausschnitt berechnen müssen (siehe Abb. 3.10).

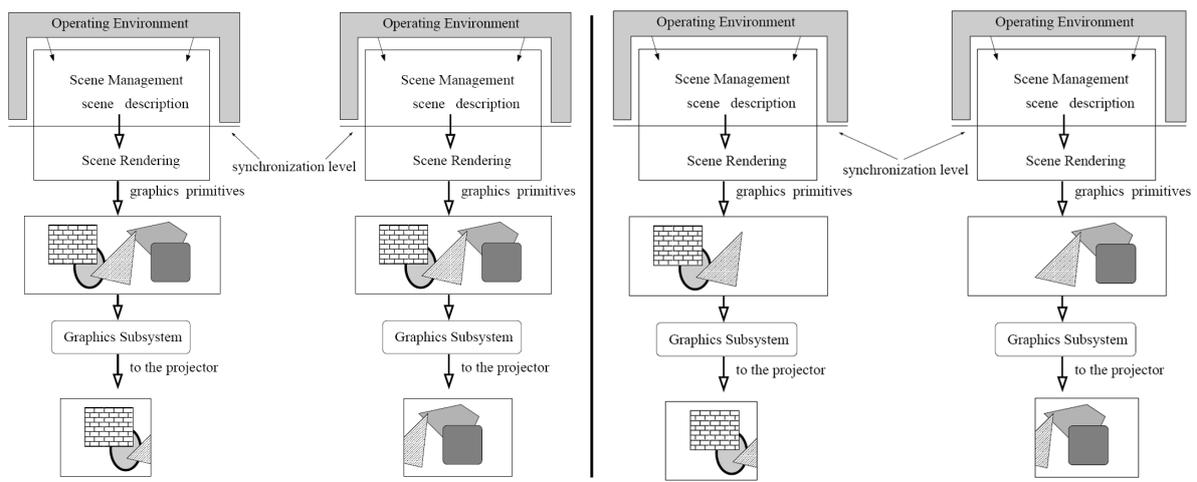


Abb. 3.10: Vollständige Replikation (links) und partielle Replikation (rechts) im Synchronized Execution Model (Chen u. a., 2001b):

Die Platzierung der Synchronisationsgrenze beim *Synchronized Execution Model* eröffnet nach Chen u. a. (2001b) zwei Verfahren für die konkrete Realisierung. Bei diesen handelt es sich dabei um die *System-level program synchronization* und um die *Application-level program synchronization*. Beide Verfahren werden im Folgenden kurz dargestellt und bewertet.

### System-level program synchronization

Das Ziel bei diesem Verfahren ist eine transparente Replikation der Anwendungen auf die verschiedenen Rechner im Cluster. Damit soll es möglich sein, bestehende Anwendungen ohne Modifikationen in einem Visualisierungssystem betreiben zu können.

Die Synchronisationsgrenze wird dazu zwischen dem Betriebssystem und der Anwendung platziert. Sämtliche Systemaufrufe vom und zum Betriebssystem, die eine Zustandsänderung der Anwendung bewirken, werden mit allen Rechnern des Clusters synchronisiert (siehe Abb. 3.11).

Bei *Single-Thread* Anwendungen führt die Synchronisation auf der Systemaufrufe Ebene auch zu einer synchronen Programmausführung. Der Programmablaufpfad wird ausschließlich durch

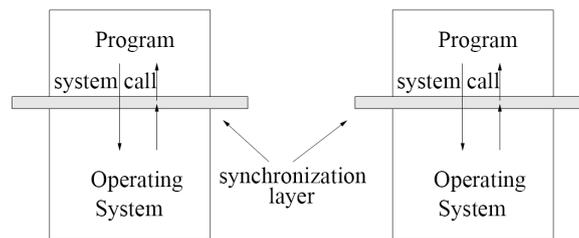


Abb. 3.11: Synchronisationsebene bei der System-level programm synchronization (Chen u. a., 2001b)

externe Ereignisse beeinflusst, die wiederum über die Systemaufrufsschnittstelle und die Systemsignalisierung des Betriebssystems transportiert werden. Die Reihenfolge der externen Ereignisse und damit die Zustandsübergänge der Anwendung sind auf allen parallel laufenden Instanzen gleich.

Problematisch wird dieses Verfahren jedoch bei Anwendungen, die *Multi-Threading* benötigen. Hier lässt sich nicht sicherstellen, dass die Threads auf allen parallel laufenden Instanzen die externen Ereignisse in derselben Reihenfolge generieren. Für solche Anwendungen reicht die *System-level program synchronization* daher nicht aus. Ein weiterer Nachteil ist, dass eine Trennung in *Szenenverwaltung* und *Szenenberechnung* bei diesem Verfahren nicht möglich ist, da die Anwendung „vollständig oberhalb“ der Synchronisationsgrenze liegt. Die Anwendung wird damit stets vollständig repliziert, während eine partielle Replikation nicht möglich ist (vgl. Abschnitt 3.3.2).

### Application-level program synchronization

Um die Probleme des vorherigen Verfahren zu umgehen, lässt sich die Synchronisationsgrenze auch direkt in der Anwendung platzieren. Diese Verfahren funktioniert analog zur *System-level program synchronization*. Anstelle von Systemaufrufen werden hier jedoch Funktionsaufrufe zwischen den Instanzen synchronisiert.

Die *Application-level program synchronization* ermöglicht damit sowohl die vollständige als auch die partielle Replikation. Insbesondere durch die partielle Replikation ist dieses Verfahren sehr gut skalierbar, da jeder Knoten (bei entsprechender Konfiguration) nicht alle Objekte der Szene berücksichtigen muss. Offensichtlicher Nachteil ist jedoch, dass man auf eine für die Anwendung transparente Lösung zur Synchronisation verzichtet. Anwendungen für dieses Verfahren müssen speziell angepasst bzw. neu entwickelt werden.

### 3.3.3 Verfahren für die Szenensynchronisation

Bei der Szenensynchronisation wird die Beschreibung der 3D-Szene in Form von grafischen Primitiven oder Objekten eines Szenengraphen zwischen den Instanzen synchronisiert. Die eigentliche Berechnung des Bildes einer Szene soll damit auf mehrere Workstations und deren GPUs aufgeteilt werden. Die Verfahren die dafür geeignet sind werden häufig auch unter den *Verfahren für paralleles Rendering* zusammengefasst.

Allgemein ist für das parallele Rendering zunächst eine *Dekomposition* notwendig, die die zu bewältigende Renderaufgabe in mehrere Teilaufgaben unterteilt und an die vorhandenen Ressourcen verteilt. Nach dem eigentlichen Renderprozess erhält man eine Reihe von Teilresultaten, die dann durch eine *Komposition* zum Endresultat zusammengeführt werden müssen (Abb. 3.12).

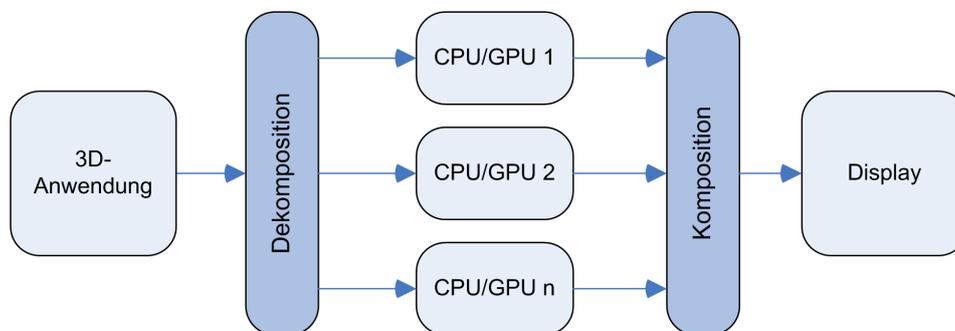


Abb. 3.12: Dekomposition und Komposition beim parallelen Rendering (in Anlehnung an [Robert \(2006\)](#))

Es existieren viele konkrete Ansätze, das Rendering durch den Einsatz von Dekomposition und Komposition zu parallelisieren. Um die Vielzahl der Ansätze einordnen und vergleichen zu können, wurde der Prozess des Renderns von [Molnar u. a. \(1994\)](#) untersucht und als Sortierproblem betrachtet. Die Ergebnisse dieser Untersuchung gelten noch immer als wichtige Grundlage in aktuellen Realisierungen.

Vor der Bildberechnung liegt die Geometrie nach [Molnar u. a. \(1994\)](#) in einem ungeordneten Zustand vor. Das Ziel des Renderprozesses ist es, die Geometrie in eine Ordnung zu bringen, so dass als Ergebnis ein korrektes Bild entsteht. Das Rendering kann dafür in zwei aufeinanderfolgende Teilaufgaben zerlegt werden: die *Geometriebearbeitung* und die *Rasterisierung*. Während der Geometriebearbeitung werden Objektpositionen und der Einfluss von Oberflächenmaterialien sowie Lichtquellen berechnet. Außerhalb des sichtbaren Bereichs liegende grafische Primitive werden verworfen. Bei der Rasterisierung werden die Sichtbarkeiten und Farben von den Bildschirmpunkten bestimmt, die aus den Ergebnissen der Geometriebearbeitung hervorgehen.

Die Sortierung in den Verfahren für paralleles Rendering kann während der Geometriebearbeitung, zwischen Geometriebearbeitung und Rasterisierung sowie während der Rasterisierung

erfolgen. In Abhängigkeit von dem so gewählten Zeitpunkt beschreibt man diese Verfahren als *Sort-First*, *Sort-Middle* oder *Sort-Last*.

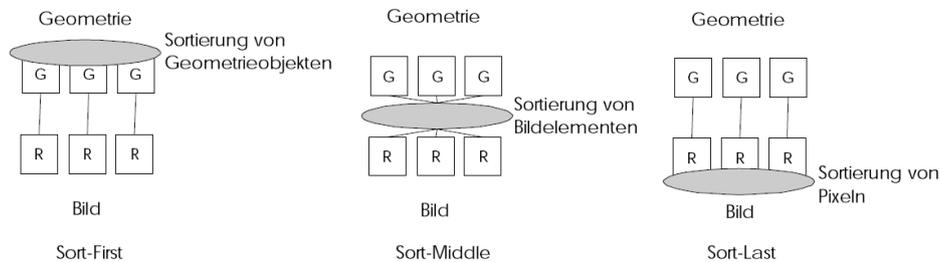


Abb. 3.13: Klassifizierung paralleler Rendering-Verfahren (Roth, 2002)

Für die Konzeption des Systems in dieser Arbeit kommen unterschiedliche Verfahren für das parallele Rendering in Frage. Bei der Auswahl eines geeigneten Verfahrens für das System dieser Arbeit ist hier insbesondere von Bedeutung, wie sich diese bezüglich der Aspekte *Performanz* und *Skalierbarkeit* verhalten. Im Folgenden werden daher die gängigsten Verfahrensklassen vorgestellt und bewertet.

### Sort-First/ Tiled Rendering

Für verteilte Visualisierungssysteme bedeutet ein Sort-First Verfahren, dass zu Beginn der Geometriebearbeitung jedes graphische Objekt einem festgelegten Bildbereich zugeordnet wird. Jeder Rechner im Cluster ist für einen Bildbereich zuständig. Es werden nur die Objekte berechnet, die innerhalb des Zuständigkeitsbereiches liegen. Ein Beispiel für die Realisierung eines Sort-First-Verfahrens ist WireGL (WireGL, 2006). Hier werden primitive OpenGL-Instruktionen auf mehrere Rechner verteilt. Der größte Vorteil dieses Ansatzes ist, dass die Bildberechnung bestehender OpenGL-Anwendungen ohne Modifikationen parallelisiert werden kann.

Da die Sortierung der grafischen Primitiven sehr früh in dem Bildberechnungsprozess stattfindet, skaliert das Verfahren nach Molnar u. a. (1994) in Bezug auf den insgesamt notwendigen Kommunikationsaufwand. Mit der statischen Zuordnung in festgelegte Bildbereiche gibt es jedoch Probleme, wenn die Objekte in bestimmten Bildbereichen dauerhaft erhöhte „Aktivitätsgrade“ aufweisen. Die so entstehende Rechenlast wird dann nur unregelmäßig auf die Ressourcen verteilt (load imbalancing), was zu einer Einschränkung der Skalierbarkeit führt.

### Sort-Middle Rendering

Dieses Verfahren ist mittlerweile nicht mehr zeitgemäß. Sowohl die Geometriebearbeitung als auch die Rasterisierung werden bei modernen Grafikkarten als eine Einheit durchgeführt. Die

Zwischenergebnisse stehen damit für eine Sortierung nicht mehr zur Verfügung. Sort-Middle-Verfahren lassen sich daher in verteilten Visualisierungssystemen auf Software-Ebene nur schwer realisieren. Auf Hardware-Ebene erweist sich dieses Verfahren jedoch als äußerst performant, wie z.B. in (Eldridge u. a., 2000) gezeigt wird.

### Sort-Last Rendering

Bei dem Sort-Last-Verfahren werden die Geometriedaten in Tiefenbereiche unterteilt und parallel gerendert. Auf diese Weise entstehen Teilbilder mit den Ausmaßen des endgültigen Bildes, jedoch nur mit einem Teil der Bildinformationen. Nach der Rasterisierung werden die Teilbilder daher übereinandergelegt und zum gewünschten Endbild zusammengefügt. Zur Bestimmung der Sichtbarkeit von einzelnen Pixeln sind dafür zusätzlich Tiefeninformationen für die Bildschirmpunkte erforderlich.

Der Kommunikationsaufwand ist bei diesem Verfahren deutlich höher als bei dem Sort-First-Verfahren. Auch die Komposition des endgültigen Bildes ist aufwendiger. Dennoch stellen Molnar u. a. (1994) fest, dass das Sort-Last-Verfahren sehr gut skaliert und die Rechenlast auch bei unterschiedlich starken Aktivitätsbereichen im Bild in der Regel gleichmäßig auf die Ressourcen aufteilt.

### 3.3.4 Realisierungsansätze: Invasiv vs. Nicht-Invasiv

Die zuvor dargestellten Strategien und Verfahren für die Skalierbarkeit von verteilten Visualisierungssystemen lassen sich grundsätzlich in zwei verschiedenen Ausprägungen realisieren (Robert, 2006):

**Invasiv:** Bei diesem Ansatz steht die maximale Effizienz in Bezug auf Skalierbarkeit und Flexibilität des Visualisierungssystems im Vordergrund. Um dies garantieren zu können, bieten die konkreten Realisierungen des invasiven Ansatzes im Allgemeinen eine eigene optimierte API<sup>5</sup> zur Nutzung des Visualisierungssystems an. Der Nachteil ist dabei, dass keine existierenden Anwendungen in solchen Umgebungen eingesetzt werden können. Anwendungen müssen speziell entwickelt bzw. für die API adaptiert werden. Stellvertreter dieses Ansatzes sind beispielsweise OpenGL Multipipe SDK (SGI, 2002), CAVELib (Mechdyne, 2007a) oder Performer (SGI, 2008a).

**Nicht-Invasiv:** Im Gegensatz zum invasiven Ansatz werden hier Abstriche in der Performanz akzeptiert, um die spezielle Adaption bzw. Neuentwicklung von Anwendungen zu verhindern. Der Einsatz in einer speziellen Visualisierungsumgebung soll für bestehende Applikationen transparent bleiben. Dementsprechend werden hierbei nicht eigene APIs eingesetzt, sondern Abstraktionsschichten zu Standardschnittstellen, wie z.B. einem virtuellen Bildschirmtreiber. Nach diesem Prinzip sind u.a. WireGL (WireGL, 2006), Chromium (Chromium, 2007) und OpenGL Vizserver (SGI, 2008b) gestaltet.

---

<sup>5</sup>Application Programming Interface

Für die Auswahl eines geeigneten Ansatzes bzw. für die mögliche Wiederverwendung von bestehenden Lösungen muss demzufolge geklärt werden, welche Art von Anwendungen mit dem skalierbaren Visualisierungssystem betrieben werden sollen.

## 3.4 Auswertung und Zusammenführung

Das praktische Ziel dieser Arbeit ist die Entwicklung einer spielbasierten Lern- und Trainingsumgebung für Gruppen, die sich zur gleichen Zeit am gleichen Ort aufhalten, sowie eines exemplarischen Serious Game, welches die Funktionalität dieser Umgebung veranschaulichen soll.

In den vorherigen Abschnitten wurde dazu analysiert, welche Designimplikationen und Lernmodelle bei der Entwicklung von kollaborativen Serious Games verwendet werden können (Unterkapitel 3.1), welche Vorteile sich bei der Nutzung von hochauflösenden Powerwalls ergeben (Unterkapitel 3.2) und durch welche Strategien und Verfahren sich Visualisierungen skalieren lassen (Unterkapitel 3.3).

Basierend auf der Analyse wird in den folgenden Abschnitten ein Beispielszenario für den Einsatz einer solchen spielbasierten Trainingsumgebung vorgestellt, bei der sich die Eigenschaften eines CCW zu nutze gemacht und Serious Games in Form von spielbasierten Simulationen zur Wissensvermittlung eingesetzt werden.

Für eine differenzierte Begrifflichkeit wird diese Trainingsumgebung in den weiteren Ausführungen dieser Arbeit aufgrund ihrer unmittelbaren Verwandtschaft zum CCW und zur Kennzeichnung der speziellen Nutzform als *Co-located Collaborative Training Environment* (CCTE) bezeichnet.

### 3.4.1 Beispielszenario für Gruppentrainings

Das UbiComp-Projekt der HAW Hamburg befässt sich in einem seiner aktuellen Forschungsthemen mit der funktionalen und technologischen Entwicklung von Geräten und Arbeitsumgebungen für Rettungseinsätze der Feuerwehr bei Großbränden. In dem sogenannten *Rescue-Szenario* werden sowohl unterstützende Technologien für die unmittelbare Brandbekämpfung (z.B. tragbare Computersysteme für die Einsatzkräfte (Hink, 2007b)) als auch für die Einsatzkoordination (z.B. die Entwicklung eines Leitstandes (Piening, 2007)) untersucht. Das Beispielszenario in dieser Arbeit knüpft an die Arbeiten des Rescue-Szenarios an, beschäftigt sich jedoch mit den vorbereitenden Trainingsmaßnahmen für solche Rettungseinsätze im Katastrophenfall.

### **Beschreibung des Szenarios**

Ein Einsatzteam der Feuerwehr besteht in der Regel aus den *Einsatzkräften* und der *Einsatzleitung*. Die Einsatzkräfte sind diejenigen, die unmittelbar mit der Brandbekämpfung bzw. der Personenrettung beschäftigt sind. Die Einsatzleitung koordiniert die Aktionen der Einsatzkräfte. Damit ein realer Einsatz bei Großbränden reibungslos verlaufen kann, müssen sowohl die einzelnen Aktionen aller beteiligten Mitglieder eines Einsatzteams als auch deren Kooperation im Vorfeld einstudiert werden. Für diese Vorbereitung werden Trainingsmaßnahmen durchgeführt, um verschiedene Szenarien zu simulieren, mit denen das Einsatzteam in einem realen Einsatz konfrontiert werden kann.

Der Fokus der Trainingsmaßnahmen liegt in diesem Beispielszenario auf Situationen, in denen die Rettung von Personen aus brennenden Gebäuden erforderlich ist. Insbesondere bei Großbränden ist eine genaue Planung und Koordination des Einsatzes notwendig, da das Einsatzteam hier oftmals in unbekanntem und großen Gebäudekomplexen, wie z.B. Flughafenterminals oder Hochhäusern, agieren muss.

Das Training des Einsatzteams findet in diesem Beispielszenario allerdings nicht auf einem Übungsgelände, sondern in einer *Co-located Collaborative Training Environment (CCTE)* statt. Unter Verwendung eines 3D-basierten Computerspiels wird ein interaktives System bereitgestellt, in dem verschiedene Trainingsszenarien definiert und simuliert werden. Zu der allgemeinen Definition eines Trainingsszenarios gehören hier:

- eine virtuelle Trainingsumgebung für das Szenario (z.B. ein dreidimensionales Gebäude- oder Geländemodell)
- Objekte in der virtuellen Trainingsumgebung, mit denen interagiert werden kann
- die Formulierung von Aufgaben, die im Trainingsszenario zu bewältigen sind
- die Festlegung von Fähigkeiten und Werkzeugen, die für die Erfüllung der Aufgaben verwendet werden können

Die Mitglieder des Einsatzteams können sich mit Avataren durch die virtuelle Trainingsumgebung des Computerspiels bewegen und zur Erfüllung der Trainingsziele mit den Objekten in der Umgebung interagieren.

### **Trainingsziele im Szenario**

Brandherde breiten sich schnell aus, was zu einer verstärkten Rauchentwicklung und oftmals auch zu einer akuten Einsturzgefahr des Gebäudes führt. Bei realen Rettungseinsätzen ist daher die „benötigte Zeit“ ein entscheidender Faktor darüber, ob der Einsatz gelingt und ob es zu Personenschäden kommt. Insbesondere in großen Gebäudekomplexen ist eine schnelle Durchführung von Evakuierungsmaßnahmen umso entscheidender, da hier oftmals weite und verschachtelte Streckenabschnitte zurückgelegt werden müssen.

Ein gut eingespieltes Einsatzteam muss in der Lage sein, Rettungseinsätze „so schnell wie möglich“ und „so gut wie möglich“ durchführen zu können. Durch das spielbasierte Gruppentraining des Einsatzteams im CCTE sollen daher insbesondere die folgenden Aspekte geschult werden:

**Planung des Einsatzes:** Bevor der eigentliche Einsatz beginnen kann, muss sich das Einsatzteam mit den bekannten Parametern des Einsatzortes vertraut machen. Zu diesen Parametern zählen z.B. eine Übersicht über das zu evakuierende Gebäude (z.B. ein Grundriss), bekannte Brandherde im Gebäude sowie die Anzahl und mögliche Aufenthaltsorte der zu evakuierenden Personen. Das Ziel ist hier, dass das Einsatzteam unter Zeitdruck mit den zur Verfügung stehenden und unter Umständen eingeschränkten Informationen in der Lage ist, eine möglichst optimale initiale Vorgehensweise zu planen.

**Koordination des laufenden Einsatzes:** Während der Einsatz läuft, d.h. die Evakuierung des Gebäudes durchgeführt wird, müssen die einzelnen Aktivitäten aller Teammitglieder koordiniert werden. Hier steht die Identifizierung und Delegation von Aufgaben bzw. Teilaufgaben im Trainingsszenario im Vordergrund, so dass der gesamte Zeitaufwand minimiert wird.

**Kommunikation und Kooperation der Teammitglieder:** Während eines Einsatzes sind gute Kommunikations- und Kooperationsfähigkeiten gefragt, um das gemeinsame Ziel (in diesem Fall die Evakuierung aller Personen) zu erreichen. Ein wiederholtes Durchlaufen der Trainingsszenarien soll diesbezüglich dazu beitragen, beim Einsatz auftauchende Situationen bzw. Probleme präzise schildern und gemeinsam bewältigen zu können.

**Räumliche Orientierung:** Für die Einsatzkräfte ist es von Vorteil, wenn sie eine gewisse „Ortskenntnis“ besitzen. In der virtuellen Trainingsumgebung ist es möglich dreidimensionale Modelle real existierender Gebäude nachzubilden. Damit können sich die Einsatzkräfte durch wiederholtes Training im CCTE mit dem Aufbau des Gebäudes vertraut machen und ihre räumliche Orientierung in diesem Gebäude erhöhen, ohne das real existierende Gebäude vorher tatsächlich betreten zu haben.

**Reaktion auf dynamische Situationsänderungen:** Während eines Einsatzes sind die Parameter des Einsatzortes, die in der Einsatzplanung zugrunde gelegt wurden, einem ständigen Wandel unterlegen. So entstehen beispielsweise neue Brandherde, vermeintlich begehbare Streckenabschnitte sind blockiert oder es befinden sich mehr Personen im Gebäude, als zunächst angenommen wurde. Im Rahmen der Ausbildung im CCTE soll das gesamte Einsatzteam (d.h. Einsatzleitung und Einsatzkräfte) mit solchen dynamischen Situationsänderungen konfrontiert werden, um eine gewisse Routine für die Behandlung von solchen Ereignissen zu entwickeln.

Die hier dargestellten Aspekte beziehen sich zunächst nur auf das geschilderte Beispielszenario und charakterisieren keine allgemeingültigen Trainingsziele im CCTE.

## Aufgaben und Rollen

Die Durchführung der Teamtrainings erfolgt auf Basis der zuvor erwähnten Trainingsszenarien, die in der spielbasierten Simulation laufen. Bei dieser Ausbildung im CCTE werden drei verschiedene Rollen unterstützt (siehe Abb. 3.14):

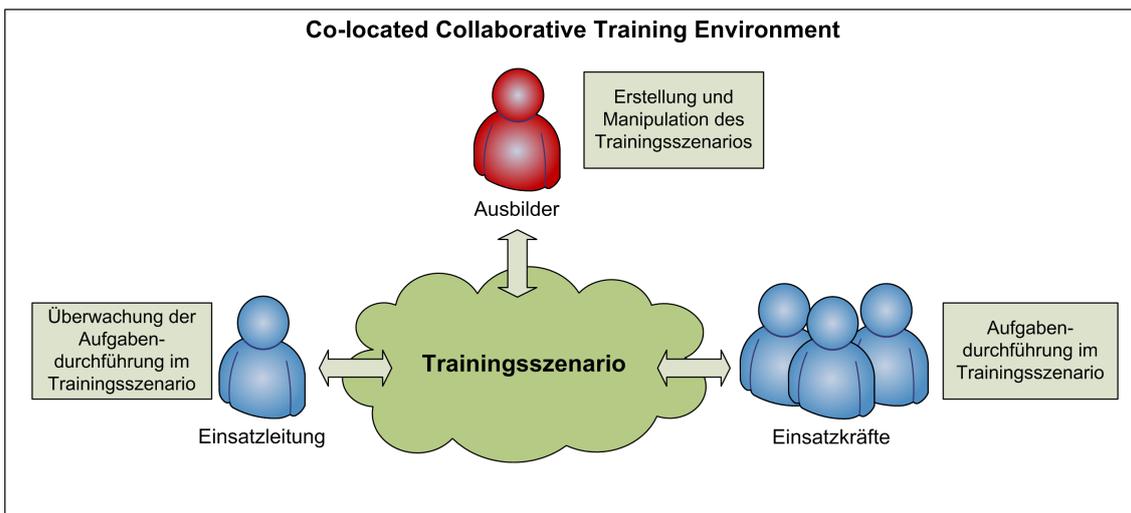


Abb. 3.14: Aufgaben und Rollen in dem Beispielszenario

**Einsatzkraft:** Die Einsatzkräfte sind die ausführenden Elemente einer Gruppe. Sie können zu Beginn eines Trainingsszenarios bei der Planung zur initialen Vorgehensweise involviert sein. Während der Ausbildung üben sie größtenteils praktische Tätigkeiten aus, d.h. sie interagieren direkt mit Objekten in der virtuellen Trainingsumgebung, um bestimmte Aufgaben bzw. Teilaufgaben zu erfüllen. Zur Erfüllung der Aufgaben müssen sich die Gruppenmitglieder direkt in der Umgebung bewegen können. Um die Bedingungen eines realen Einsatzes abzubilden, ist die Sicht der Einsatzkräfte eingeschränkt, d.h. sie können die virtuelle Trainingsumgebung nur aus einer Ego-Perspektive durchlaufen.

**Einsatzleitung:** Die Einsatzleitung plant, überwacht und koordiniert die Aufgaben, die in einem Trainingsszenario zu bewältigen sind. Sie delegiert die Einsatzkräfte und weist ihnen Aufgaben bzw. Teilaufgaben zu, die für eine erfolgreiche Beendigung des Trainingsszenarios notwendig sind. Die Rolle der Einsatzleitung erfordert eine große Übersicht über alle laufenden Aktivitäten und Zusammenhänge im Trainingsszenario, ohne dabei die Sicht für detaillierte Informationen zu verlieren.

**Ausbilder:** Der Ausbilder führt die Gesamtaufsicht über den Verlauf der Ausbildung. Er erstellt das Trainingsszenario, die darin zu lösenden Aufgaben und startet bzw. beendet die Ausbildung. Während der Ausbildung kann er Hilfestellungen geben oder ggfs. den Schwierigkeitsgrad des Trainingsszenarios verändern, indem er z.B. neue Aufgaben stellt oder neue Objekte als Hindernisse hinzufügt bzw. entfernt. Für seine Aufgabe muss der Ausbilder die gesamte Gruppe sowie deren Kommunikation und Kooperation beobachten

können. Zur Manipulation des Trainings szenarios muss er zudem die Möglichkeit haben, sich wie die Einsatzkräfte in der virtuellen Trainingsumgebung frei bewegen zu können.

Auf Basis der Aufgaben, die die drei Rollen in dem Gruppentraining ausüben, lassen sich unterschiedliche Anforderungen an das System ableiten. In den nächsten beiden Abschnitten werden daher die funktionalen und nicht-funktionalen Anforderungen besprochen, die das System leisten muss, um die im Beispielszenario skizzierte Ausbildung vom Einsatzteam effektiv zu unterstützen.

### 3.4.2 Funktionale Anforderungen

Auf abstrakter Ebene lassen sich mit den drei involvierten Rollen beim Training im CCTE fünf verschiedene Anwendungsfälle ableiten (Abb. 3.4.2):

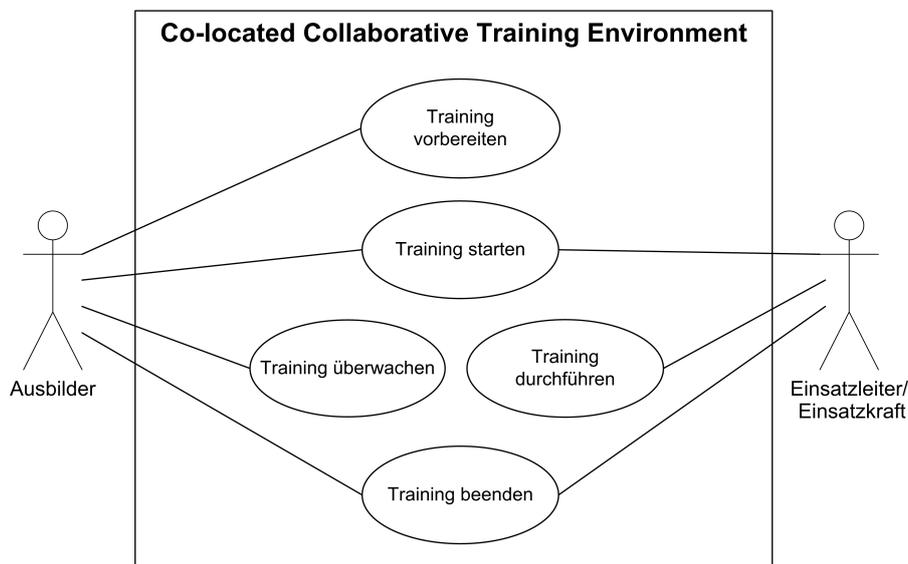


Abb. 3.15: Allgemeine Anwendungsfälle im CCTE

**Training vorbereiten:** Bevor eine Trainingsmaßnahme durchgeführt werden kann, muss der Ausbilder die gesamte Trainingsumgebung für die Auszubildenden vorbereiten.

**Training starten:** Nachdem alle Vorbereitungen getroffen und administrativen Angelegenheiten geklärt wurden, kann das eigentliche Training für alle Beteiligten gestartet werden.

**Training durchführen:** Die Einsatzleiter und Einsatzkräfte werden im Rahmen des Trainings mit einem Trainings szenario konfrontiert, in dem sie verschiedene Aufgaben bewältigen müssen, um die vom Ausbilder definierten Einsatzziele zu erreichen.

**Training überwachen:** Während der Ausbildung kommt dem Ausbilder die Aufgabe zuteil, das Training zu überwachen. Dabei muss er sowohl die Aktivitäten in der virtuellen als auch in der realen Trainingsumgebung beaufsichtigen.

**Training beenden:** Das Einsatzteam kann die Ausbildung beenden, wenn entweder alle Einsatzziele erreicht wurden oder der Einsatz fehlgeschlagen ist. Der Ausbilder hat zudem die Möglichkeit den Einsatz vorzeitig abubrechen.

Die Trainingsmaßnahmen sollen im CCTE auf Basis einer spielbasierten Simulation erfolgen. Diese muss so gestaltet sein, dass die oben genannten allgemeinen Anwendungsfälle für alle Rollen ermöglicht werden. In den nachfolgenden Unterabschnitten werden daher die allgemeinen Anwendungsfälle konkretisiert und darauf basierend die funktionalen Anforderungen an das System erhoben. Für eine erhöhte Übersichtlichkeit werden die Anwendungsfälle für alle drei Rollen getrennt betrachtet.

### **Funktionale Systemanforderungen aus der Sicht des Ausbilders**

Im Folgenden werden die Anwendungsfälle beschrieben, die die funktionalen Anforderungen aus der Sicht des Ausbilders verdeutlichen:

- **Trainingsszenario einrichten:** Der Ausbilder muss für das Training ein Trainingsszenario einrichten können. Im Rahmen des Beispielszenarios (Abschnitt 3.4.1) wurde skizziert, mit welchen Elementen ein Trainingsszenario definiert wird. Darauf basierend muss das System die folgenden Funktionalitäten bieten:
  - **Virtuelle Trainingsumgebung auswählen:** Es muss die Möglichkeit bestehen, eine virtuelle Trainingsumgebung für das Szenario auswählen zu können. Bei einer solchen Umgebung kann es sich beispielsweise um Gebäude- oder Geländemodelle, die in einem 3D-Format vorliegen. Diese Modelle können entweder aus einer Liste bestehender Elemente ausgewählt oder neu definiert bzw. modelliert werden. Dies impliziert, dass das System das vorliegende 3D-Format interpretieren und grafisch darstellen kann.
  - **Objekte definieren:** Innerhalb der virtuellen Trainingsumgebung kann der Ausbilder Objekte und deren Eigenschaften definieren, die für das Trainingsszenario benötigt werden. In dem Beispielszenario dieser Arbeit kann es sich dabei z.B. um Brandherde oder Löschwerkzeuge handeln. Wie auch bei den Modellen der virtuellen Trainingsumgebung werden für diese Objekte entweder bestehende Elemente verwendet oder neue Elemente definiert bzw. modelliert.
  - **Aufgaben und Ziele definieren:** Das System muss die Definition von Aufgaben und Zielen ermöglichen, die von dem Einsatzteam im Rahmen des Trainingsszenarios erledigt bzw. erreicht werden müssen.
  - **Rahmenbedingungen definieren:** Darüberhinaus muss es möglich sein, Rahmenbedingungen für den Einsatz definieren zu können. Dazu zählen neben den verfügbaren Fähigkeiten und Werkzeugen der verschiedenen Rollen auch Parameter des Einsatzes, wie z.B. eine (maximale) Zeitvorgabe oder die Anzahl der zu evakuierenden Personen in einem Gebäude.

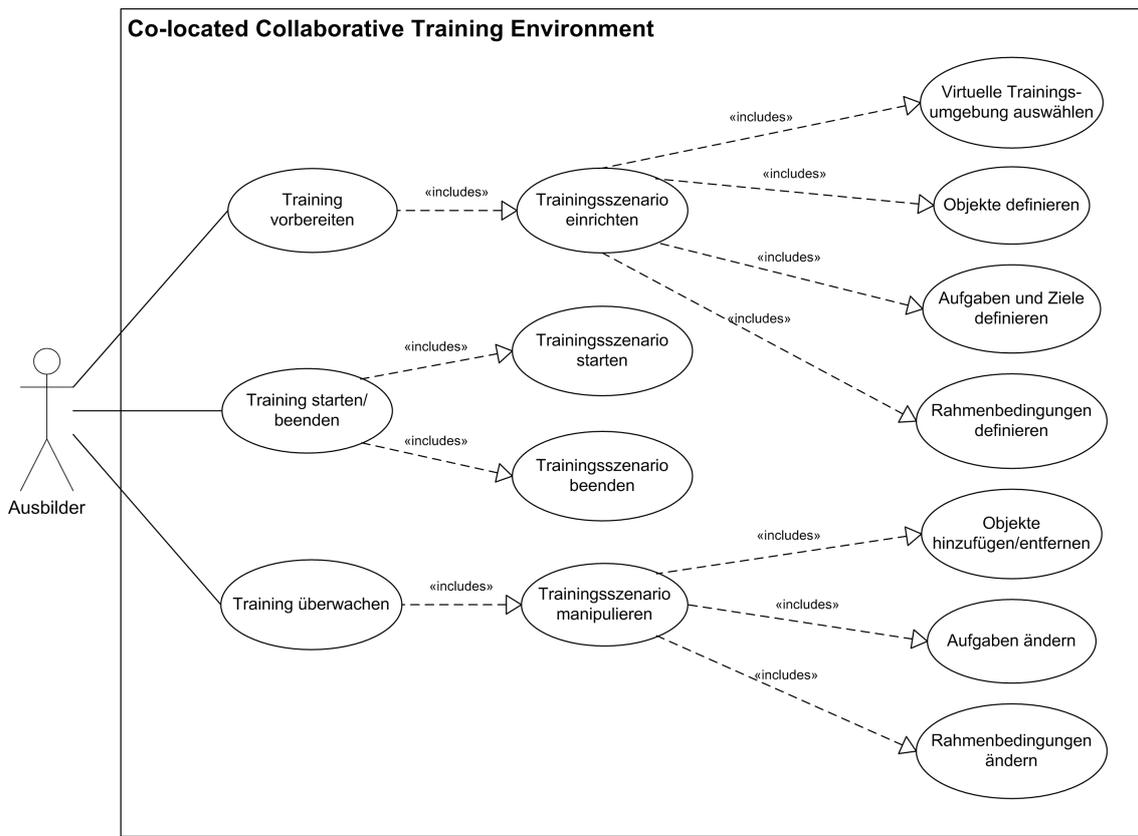


Abb. 3.16: Anwendungsfalldiagramm für die Rolle des Ausbilders

- **Trainingszenario starten:** Nach der Einrichtung des Trainingszenarios kann es gestartet werden. Das System startet die eigentliche spielbasierte Simulation, in der alle Parameter berücksichtigt werden, die durch die Definition des Trainingszenarios vorgegeben wurden. Das Einsatzteam kann nun dem Trainingszenario beitreten, um mit dem Training zu beginnen.
- **Trainingszenario beenden:** Wenn alle Einsatzziele erreicht wurden oder das Training durch den Ausbilder abgebrochen wird, wird die spielbasierte Simulation beendet. Das Trainingszenario kann nun entweder wiederholt oder ein neues Szenario vorbereitet werden.
- **Trainingszenario manipulieren:** Während des Einsatzes muss das System die Möglichkeit bieten, die Parameter der spielebasierten Simulation zu beeinflussen. Damit kann der Ausbilder den Schwierigkeitsgrad des Trainings dynamisch steuern. Folgende Aktionen stehen dabei zur Verfügung:
  - **Objekte hinzufügen/entfernen:** Der Ausbilder kann der virtuellen Trainingsumgebung neue Objekte hinzufügen bzw. bestehende Objekte aus der Umgebung entfernen.
  - **Aufgaben oder Ziele ändern:** Alle bestehenden Aufgaben und Ziele können ge-

ändert werden. In diesem Rahmen ist es auch möglich neue Aufgaben oder Ziele zu definieren bzw. bestehende Aufgaben oder Ziele zu entfernen.

- **Rahmenbedingungen ändern:** Die Rahmenbedingungen können während des laufenden Einsatzes manipuliert werden. So kann z.B. die zur Verfügung stehende Zeit verkürzt bzw. verlängert werden.

In Abbildung 3.4.2 werden die Zusammenhänge aller beschriebenen Anwendungsfälle für den Ausbilder in einem UML-konformen Anwendungsfalldiagramm dargestellt.

### Funktionale Systemanforderungen aus der Sicht des Einsatzleiters

Im Folgenden werden die Anwendungsfälle beschrieben, die die funktionalen Anforderungen aus der Sicht des Einsatzleiters verdeutlichen:

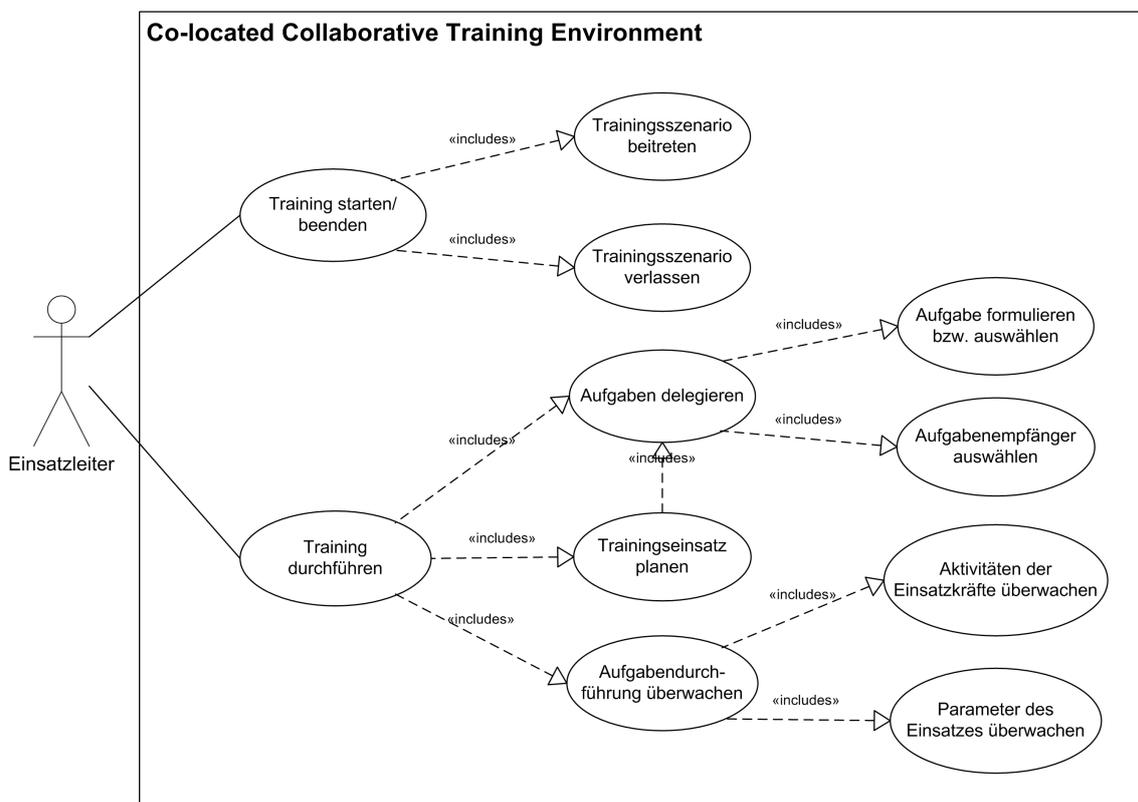


Abb. 3.17: Anwendungsfalldiagramm für die Rolle des Einsatzleiters

- **Trainingsszenario beitreten:** Wenn das Trainingsszenario von dem Ausbilder erstellt und gestartet wurde, kann der Einsatzleiter der spielbasierten Simulation als Benutzer beitreten. Der Einsatzleiter erhält eine Visualisierung der virtuellen Trainingsumgebung sowie Interaktionsmöglichkeiten innerhalb dieser Umgebung, die ihm aufgrund seiner Rolle vom Ausbilder zugesprochen wurden.

- **Trainingseinsatz planen:** Jedes Training beginnt mit einer Einsatzplanung, in der alle bekannten Informationen des Trainingseinsatzes gesichtet werden. Das System muss hierzu Möglichkeiten bieten, die gesamte virtuelle Trainingsumgebung zu überblicken, alle zu erledigenden Aufgaben darzustellen und diese an die Einsatzkräfte verteilen zu können. Der Anwendungsfall *Aufgaben delegieren* wird während des gesamten Trainings benötigt und wird gesondert dargestellt.
- **Aufgabendurchführung überwachen:** Die Hauptaufgabe des Einsatzleiters ist die Koordination des Trainingseinsatzes. Dazu muss er alle Zusammenhänge im Trainingsszenario überwachen können. Diese Überwachung lässt sich durch zwei Anforderungen konkretisieren:
  - **Aktivitäten der Einsatzkräfte überwachen:** Für eine Koordination müssen die Aktivitäten aller Einsatzkräfte für den Einsatzleiter transparent sein. Dabei müssen sowohl detaillierte Informationen zu den Aktivitäten einzelner Einsatzkräfte als auch deren Zusammenhänge zur Verfügung stehen.
  - **Einsatzparameter überwachen:** Neben den Aktivitäten der Einsatzkräfte ist die Überwachung aller weiteren Parameter des Einsatzortes notwendig, um dynamische Situationsänderungen zu identifizieren und auf diese entsprechend reagieren zu können. Beispiele für solche Situationsänderungen sind z.B. das Aufkeimen neuer Brandherde oder das Auftauchen weiterer Personen im Gebäude.
- **Aufgaben delegieren:** Neben der reinen Überwachung der Aufgabendurchführung muss der Einsatzleiter in der Lage sein, während des Trainings Aufgaben bzw. Teilaufgaben an die Einsatzkräfte zu delegieren. Das System muss dazu im Detail zwei Funktionalitäten bereitstellen:
  - **Aufgabe formulieren bzw. auswählen:** Aufgaben oder Teilaufgaben müssen vor einer Delegation zunächst formuliert bzw. ausgewählt werden können.
  - **Aufgabenempfänger auswählen:** Eine formulierte bzw. ausgewählte Aufgabe wird einer Einsatzkraft zugeteilt und übermittelt, indem diese als Aufgabenempfänger ausgewählt werden kann.
- **Trainingsszenario verlassen:** Nach einem erfolgreichen, fehlgeschlagenen oder durch den Ausbilder abgebrochenen Einsatz kann der Einsatzleiter das Trainingsszenario verlassen. Die Visualisierung der virtuellen Trainingsumgebung wird damit beendet.

In Abbildung 3.4.2 werden die Zusammenhänge aller beschriebenen Anwendungsfälle für den Einsatzleiter in einem UML-konformen Anwendungsfalldiagramm dargestellt.

### Funktionale Systemanforderungen aus der Sicht der Einsatzkraft

Im Folgenden werden die Anwendungsfälle beschrieben, die die funktionalen Anforderungen aus der Sicht der Einsatzkraft verdeutlichen:

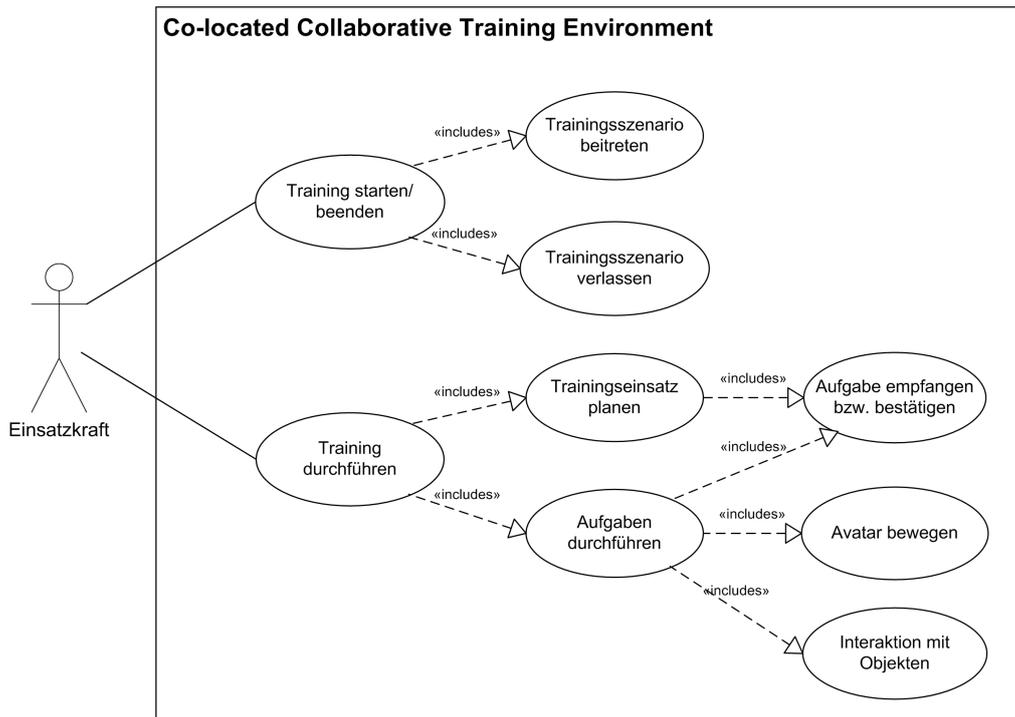


Abb. 3.18: Anwendungsfalldiagramm für die Rolle der Einsatzkraft

- **Trainingszenario beitreten:** Dieser Anwendungsfall gestaltet sich analog zu dem entsprechenden Anwendungsfall des Einsatzleiters.
- **Trainingseinsatz planen:** Auch die Einsatzkräfte nehmen an der Einsatzplanung teil und müssen von dem System dementsprechend ebenfalls mit allen bekannten Informationen über das Trainingszenario versorgt werden. Der Empfang der Aufgaben, die im Rahmen der Planung verteilt werden, müssen durch die Einsatzkräfte bestätigt werden. Näheres zur Aufgabenbestätigung wird unter *Aufgaben durchführen* erläutert.
- **Aufgaben durchführen:** Als ausführende Elemente des Einsatzteams müssen die Einsatzkräfte verschiedene Aufgaben im Trainingszenario erledigen. Für die Aufgabendurchführung müssen folgende Funktionalitäten gegeben sein:
  - **Avatar bewegen:** Für die Ausführung der Aufgaben müssen sich die Einsatzkräfte direkt in der virtuellen Trainingsumgebung bewegen können. Das System muss hierzu mit einem *Avatar* eine Repräsentation der Einsatzkräfte in der virtuellen Umgebung bereitstellen. Dieser Avatar kann von den Einsatzkräften direkt gesteuert werden.
  - **Aufgabe empfangen bzw. bestätigen:** Während des Einsatzes müssen die Einsatzkräfte Aufgaben vom Einsatzleiter empfangen und bestätigen können.
  - **Interaktion mit Objekten:** Die Einsatzkräfte bewegen sich mit ihrem Avatar durch die virtuelle Trainingsumgebung und können dabei mit den Objekten interagieren, die sich in dem Trainingszenario befinden. Die Art der möglichen Interaktionen mit

diesen Objekten ist abhängig von den Objekteigenschaften sowie den Fähigkeiten und Werkzeugen der Einsatzkräfte, die bei der Definition des Trainingsszenarios von dem Ausbilder festgelegt wurden.

- **Trainingsszenario verlassen:** Dieser Anwendungsfall gestaltet sich analog zu dem entsprechenden Anwendungsfall des Einsatzleiters.

In Abbildung 3.4.2 werden die Zusammenhänge aller beschriebenen Anwendungsfälle für die Einsatzkraft in einem UML-konformen Anwendungsfalldiagramm dargestellt.

### Weitere funktionale Systemanforderungen

Neben den funktionalen Anforderungen, die sich aus den rollenspezifischen Anwendungsfällen ergeben, muss das System einige weitere Eigenschaften aufweisen, die aus dem Beispielszenario hervorgehen:

- **Spielbasiertes Training:** Das Training soll auf Basis einer spielbasierten Simulation bzw. eines kollaborativen Serious Game erfolgen. Die Konzeption des konkreten Spiels soll sich an den Designimplikationen orientieren, die im Rahmen der Analyse ermittelt wurden (vgl. Abschnitt 3.1.4).
- **Rollenunterstützung durch geeignete Visualisierungsformen:** In Bezug auf die Visualisierung haben die Rollen im CCTE unterschiedliche Anforderungen:
  - Die Einsatzkräfte müssen direkt mit den Objekten in der virtuellen Trainingsumgebung interagieren können. Sie bewegen sich demnach „in der Welt“ der Spielumgebung. Sie benötigen eine flexible und steuerbare Visualisierung der Umgebung. Zur Nachbildung des Sichtfeldes in einem realen Einsatz ist diese Visualisierung einzuschränken.
  - Dagegen muss der Einsatzleiter stets die gesamte Situation und Abläufe im Trainingsszenario überwachen können. Er benötigt viele Informationen zu allen Objekten in der Welt und zu den Aktionen der Einsatzkräfte, die sich in der virtuellen Trainingsumgebung bewegen. Dazu benötigt er eine sehr detaillierte und informationsreiche Visualisierung „auf die Welt“ der Spielumgebung.
  - Der Ausbilder benötigt einerseits eine freie Beweglichkeit in der virtuellen Trainingsumgebung, um die Objekte im Trainingsszenario manipulieren zu können. Andererseits muss er die gesamte reale Trainingssituation überwachen können. Dazu benötigt er vor allem einen Überblick über die Kollaboration der Gruppe „in der realen Umgebung“.

Bei der Konzeption des Systems müssen daher zur Unterstützung der drei Rollen unterschiedliche Visualisierungsformen der virtuellen Trainingsumgebung im CCTE berücksichtigt werden.

- **Synchronität der Visualisierung:** Alle beteiligten Personen im CCTE verwenden eine Visualisierung auf dieselbe virtuelle Trainingsumgebung. Damit die Kooperation und Kollaboration zur Erreichung der Trainingsziele reibungslos verlaufen kann, ist es erforderlich, dass alle Visualisierungen synchron laufen. Jede Veränderung in der virtuellen Trainingsumgebung, wie z.B. das Hinzufügen von neuen Objekten oder die Bewegung der virtuellen Repräsentationen der Einsatzkräfte (Avatare), muss demnach zu einer konsistenten und gleichzeitigen Veränderung aller Visualisierungen führen.

Diese Aufstellung beschreibt keine konkreten Anwendungsfälle, sondern allgemeine Eigenschaften der Systemumgebung. Sie werden insbesondere als Ansätze für die Konzeption der Systeminfrastruktur des CCTE herangezogen, die zu Beginn des nächsten Kapitels in Abschnitt 4.1 spezifischer erläutert wird.

### 3.4.3 Nicht-funktionale Anforderungen

Neben den funktionalen Anforderungen, die sich aus den Anwendungsfällen der drei Rollen im CCTE ergeben, werden an das zu konzipierende System auch eine Reihe von nicht-funktionalen Anforderungen erhoben. Die nicht-funktionalen Anforderungen beschreiben Rahmenbedingungen, unter denen die funktionalen Anforderungen erbracht werden müssen, Anforderungen die unabhängig von einem speziellen Anwendungsfall sind, oder die aus anderen Gründen nicht als Funktion beschrieben werden können (Kahlbrandt, 1998).

#### Portierbarkeit und Übertragbarkeit

Das für die Anforderungsanalyse gewählte Szenario ist nur ein Beispiel für Gruppentrainings. Das System muss so gestaltet sein, dass es auch für andere Anwendungsbereiche einsetzbar ist. Insbesondere die zugrundeliegende spielbasierte Simulation soll es ermöglichen, verschiedenartige Spielkonzepte für kollaborative Serious Games im CCTE integrieren zu können.

#### Performanz und Skalierbarkeit

Bei einer Verwendung des Systems können der Umfang der Trainingsszenarien und die Anzahl der Personen, die im CCTE ausgebildet werden sollen stark variieren. Bei sehr umfangreichen Trainingsszenarien kann die virtuelle Trainingsumgebung sehr weitläufig ausfallen. Dabei werden in der Regel auch sehr viele Objekte in dieser Umgebung benötigt. Große Gruppen erfordern zudem eine entsprechende Anzahl synchron laufender Visualisierungen der virtuellen Trainingsumgebung. Damit das System auch unter diesen Umständen eine einwandfreie Funktionalität liefern kann, muss die Repräsentation und Visualisierung der virtuellen Trainingsumgebung performant und skalierbar sein.

## 4 Design und Realisierung

In diesem Kapitel werden das Design und die Realisierung des *Co-located Collaborative Training Environment* und eines exemplarischen *Serious Game*, das für das Training des Einsatzteams in dieser Umgebung verwendet wird, besprochen.

In Unterkapitel 4.1 werden zunächst einige Konzeptionsansätze für den Trainingsraum vorgestellt, um das Gesamtbild des Systems zu veranschaulichen. Anschließend werden in Unterkapitel 4.2 einige grundlegende Aspekte der Systemarchitektur vermittelt. Im darauffolgenden Unterkapitel 4.3 erfolgt schließlich die Darstellung der eigentlichen Systemarchitektur. Nach der Beschreibung des kollaborativen Spielkonzepts (Unterkapitel 4.4), das die rollenspezifischen Anforderungen integriert und für die Durchführung des Gruppentrainings eingesetzt wird, werden abschließend in Unterkapitel 4.5 die wesentlichen Aspekte der prototypischen Realisierung erläutert.

### 4.1 Konzeptionsansätze für den Trainingsraum

Die zu Beginn der Analyse geschilderte Ausgangssituation ging davon aus, dass gruppenbasierte Trainingsmaßnahmen in einem speziell dafür konzipierten Trainingsraum mit den Eigenschaften eines *Co-located Collaborative Workplace* (CCW) stattfinden (vgl. Abschnitt 2.1.3). Dazu zählen die Bereitstellung eines gemeinsamen Anwendungsbereiches und eines gemeinsamen Anwendungskontextes.

Zum Ende der Analyse wurde in Unterkapitel 3.4 für diesen Trainingsraum der Begriff *Co-Located Collaborative Training Environment* (CCTE) eingeführt. In dem folgenden Abschnitt wird genauer dargelegt, wie sich die Eigenschaften eines CCW in dem CCTE wiederfinden (Abschnitt 4.1.1).

#### 4.1.1 CCW als Co-located Collaborative Training Environment (CCTE)

Die Darstellung eines gemeinsamen Arbeitsbereiches wird in dem CCTE durch eine hochauflösende Bildschirmwand (Powerwall) übernommen. Mit diesem Anzeigemedium existiert im CCTE ein typischer Vertreter aus dem Bereich der *skalierbaren Visualisierungssysteme* (2.2). Das klassische Verständnis von einem skalierbaren Visualisierungssystem wird in dieser Arbeit jedoch erweitert.

Der Forderung nach einem gemeinsamen Anwendungskontext wird mit dem Ansatz entgegengekommen, die reale Infrastruktur des CCTE mit einer virtuellen Systemstruktur zu unterlegen (vgl. z.B. [Kindberg und Fox \(2002\)](#)). Dazu werden alle Anzeigemedien innerhalb des CCTE als Fenster zu einem virtuellen Modell - einer „3D-Welt“ - betrachtet. In dieser werden alle notwendigen Informationen bzw. Daten der Gruppe als dreidimensionale Objekte dargestellt. Die Perspektiven auf diese Welt, d.h. die Blickrichtung der Fenster, kann sich beliebig unterscheiden und von den Benutzern frei verändert werden. So können unter anderem sehr große Objekte mit vielen Detailinformationen, wie z.B. Fahrzeug- oder Gebäudemodelle, aus unterschiedlichsten Blickwinkeln betrachtet werden. Die Objekte in dem virtuellen Modell können durch die zur Verfügung stehenden Eingabegeräte von allen Benutzern manipuliert werden. Die Benutzer im CCTE sollen auf diese Weise, neben der persönlichen Kommunikation, auch über die verfügbaren Objekte im virtuellen Modell interagieren können. Die Änderungen der Objekte sollen sich dabei für die gesamte Infrastruktur konsistent verhalten, d.h. alle Visualisierungen innerhalb des CCTE zeigen stets die aktuellsten Inhalte.

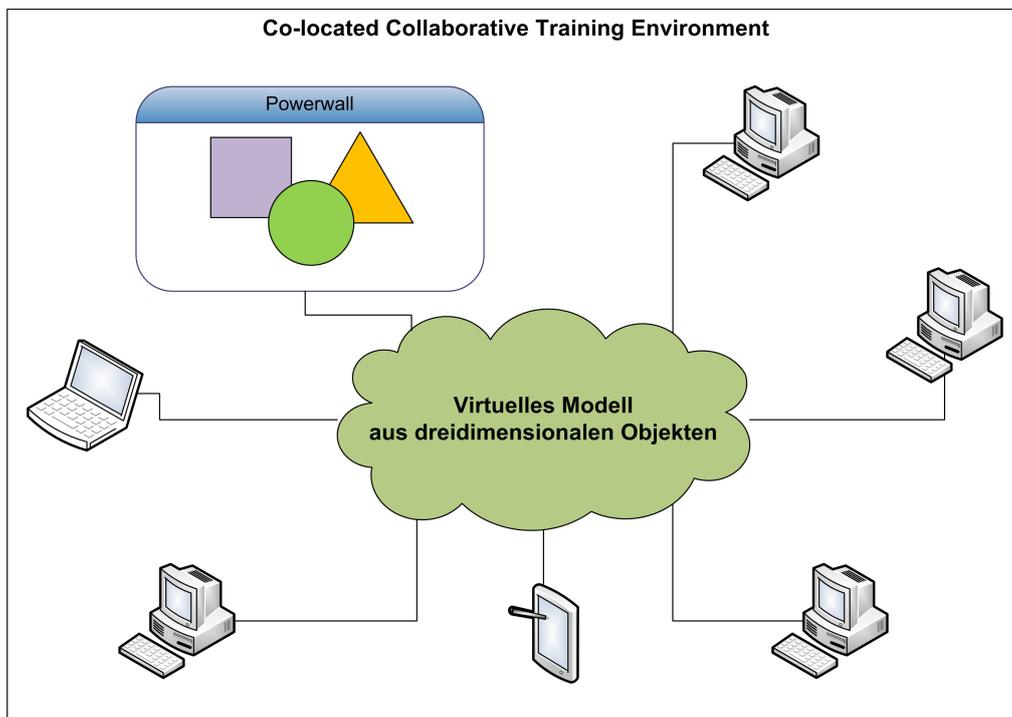


Abb. 4.1: CCTE als ein System mit skalierbarer Visualisierung: Anzeigemedien als Fenster zu einem gemeinsamen virtuellen 3D-Modell

Dieser Ansatz erfordert die parallele Ausführung von 3D-Anwendungen, die gleichzeitige Verwendung von verschiedenen Ausgabemedien und die effiziente Handhabung großer Datenmengen bzw. die Darstellung von sehr vielen Bildpunkten (sämtliche Daten werden als 3D-Objekte behandelt). Damit sind die Eigenschaften gegeben, die nach [Robert \(2006\)](#) ein skalierbares Visualisierungssystem charakterisieren (vgl. Unterkapitel 2.2). Im Rahmen dieser Arbeit wird daher der CCTE selbst als ein *System mit skalierbarer Visualisierung* begriffen, in dem mit

mehreren Anzeigemedien unterschiedliche Perspektiven auf dieselbe 3D-Welt möglich sind. Die Visualisierung mit der Powerwall, also die Verwendung von mehreren Anzeigemedien als ein einheitliches Anzeigemedium, bildet hierbei ein eigenständiges *Teilsystem* innerhalb des CCTE.

#### 4.1.2 Globale und lokale Sichten im CCTE

Die globale Sicht ist für die Einsatzleitung notwendig. Hierfür ist ein Visualisierungssystem notwendig, das in der Lage ist, sehr viele und detaillierte Informationen darzustellen. Wie in der Analyse gezeigt wurde, kann eine Powerwall diese Anforderung erfüllen. Darüberhinaus bringt diese zusätzlich einige kognitive Vorteile mit sich (Abschnitt 3.2.2), aus denen der Einsatzleiter bei der Planung und Überwachung der Trainingsszenarios profitieren kann.

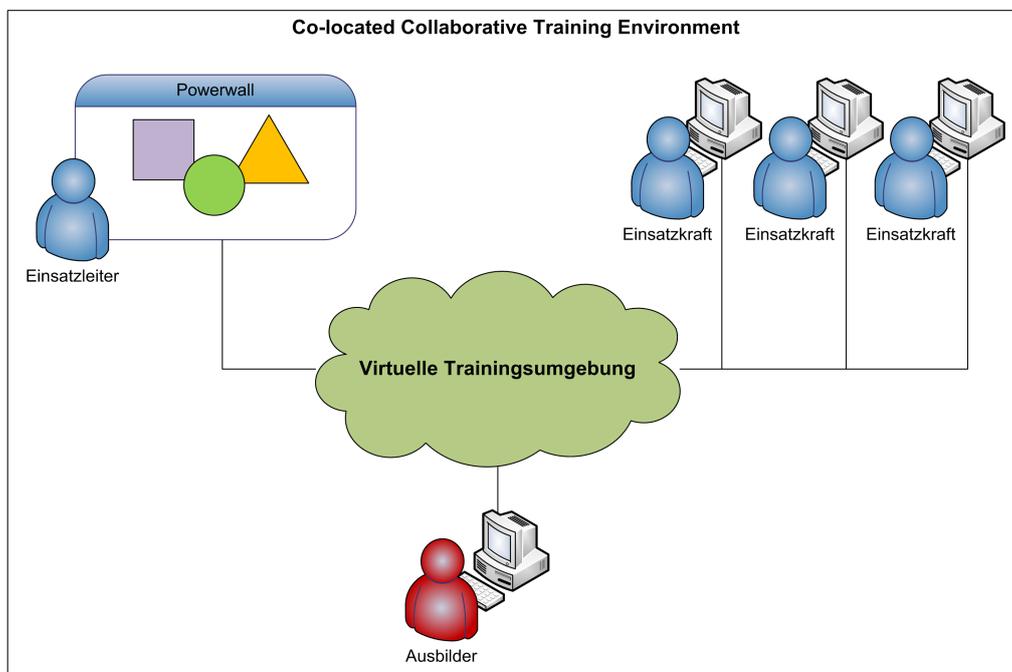


Abb. 4.2: Globale und lokale Sichten auf die virtuelle Trainingsumgebung im CCTE

Die Einsatzkräfte benötigen dagegen lokale Sichten, um sich in der virtuellen Trainingsumgebung bewegen und mit den darin enthaltenen Objekten interagieren zu können. Für eine Übertragbarkeit auf die Realität, müssen diese lokalen Sichten sowohl eine Ego-Perspektive<sup>1</sup> als auch ein eingeschränktes Sichtfeld bieten. Für letzteres eignen sich „klassische“ Anzeigemedien, bei denen nur die zentrierte Sicht eingesetzt wird (vgl. Abschnitt 3.2.2).

<sup>1</sup>Die Ego-Perspektive bezeichnet eine typische Kameraperspektive in einem Computerspiel, bei der die Darstellung der Spielwelt durch die Augen der Spielfigur erfolgt.

Die Hauptaufgabe des Ausbilders ist die Überwachung der realen Trainingsssituation, d.h. die Beobachtung der beteiligten Personen im CCTE. Der Ausbilder benötigt keine spezielle Visualisierungsform und kann daher ebenfalls eine lokale Sicht benutzen, um die Objekte in der virtuellen Trainingsumgebung zu manipulieren. Es empfiehlt sich hier jedoch, diese lokale Sicht von der Einschränkung der Ego-Perspektive zu lösen, und dem Ausbilder damit eine freie Beweglichkeit in der virtuellen Trainingsumgebung zu ermöglichen.

Eine schematische Darstellung der Visualisierung im CCTE bezüglich globaler und lokaler Sichten findet sich in Abbildung 4.2.

### 4.1.3 Aufbau der Infrastruktur des CCTE

In Abschnitt 3.2.3 wurden Interaktionsbereiche bei der Verwendung von großen Anzeigemedien vorgestellt. Unter Einbeziehung dieser Bereiche, der für die globalen bzw. lokalen Sichten notwendigen „Gerätelandschaft“ und den Aufgaben aller Rollen, die im CCTE unterstützt werden sollen (Abschnitt 3.4.1), wurde ein Konzept für den infrastrukturellen Aufbau des CCTE erstellt (Abb. 4.3).

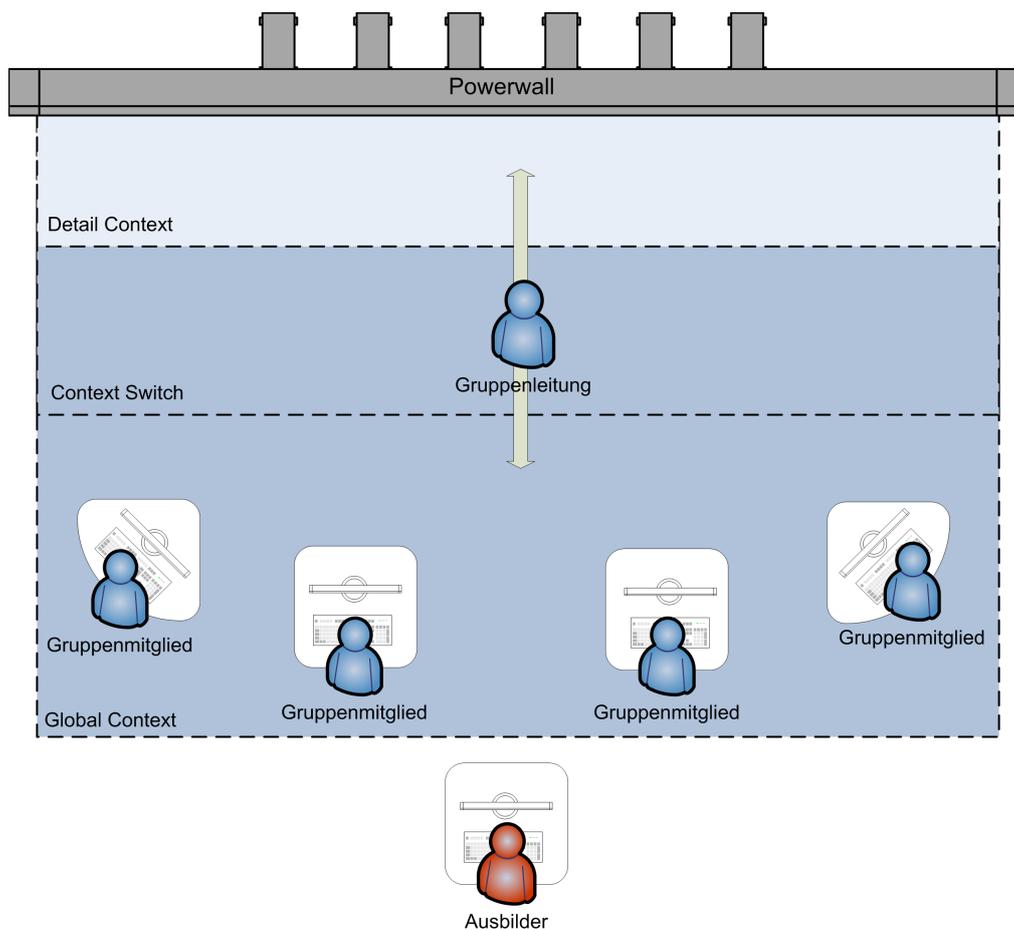


Abb. 4.3: Nutzung der Interaktionsbereiche im CCTE (vgl. Abb. 3.7)

Die Grundlage dafür bildet das in Abschnitt 3.2.3 eingeführte generalisierte Modell für Interaktionsebenen und Nutzungsbereiche (Abb. 3.7). In dem Modell wird der physikalische Raum vor der Powerwall in die Bereiche *Detail Context*, *Context Switch* und *Global Context* eingeteilt. Für die Trainingsmaßnahmen im CCTE werden die drei beteiligten Rollen wie folgt in dieses Modell eingeordnet:

**Einsatzleiter:** Der Einsatzleiter ist der *Hauptnutzer* der hochauflösenden Bildschirmwand. Um seinen Aufgaben im Rahmen des Trainings nachgehen zu können, muss er die vorteilhaften Eigenschaften der Powerwall (Abschnitt 3.2.4) uneingeschränkt nutzen können. Der Einsatzleiter kann sich daher frei in den drei Interaktionsbereichen bewegen.

**Einsatzkräfte:** Die Einsatzkräfte sind in dieser infrastrukturellen Konzeption *Nebennutzer* der Powerwall. Für die Durchführung ihrer Aufgaben werden ihnen feste Arbeitsplätze zugewiesen, die für die lokalen Sichten und Interaktionen verwendet werden. Bei der Kommunikation, Kooperation und Koordination im Rahmen des spielbasierten Gruppentrainings kann es jedoch erforderlich sein, dass das gesamte Einsatzteam z.B. bestimmte Strategien abstimmen muss. Die Powerwall dient dafür im CCTE als gemeinsamer Arbeitsbereich, der als Diskussionsplattform genutzt werden kann. Die statischen Arbeitsplätze der Einsatzkräfte werden daher in dem *Global Context* platziert, so dass neben den lokalen Sichten gleichzeitig auch eine globale Übersicht der Situation möglich ist.

**Ausbilder:** Die drei Nutzungsbereiche sind für die Aufgabendurchführung des Ausbilders irrelevant. Vielmehr muss er die gesamte „reale“ Trainingssituation, d.h. beispielsweise die Kommunikation zwischen den Gruppenmitgliedern, überblicken können. Der Ausbilder mitsamt seinem Arbeitsplatz befindet sich daher beim Training im Hintergrund des eigentlichen Geschehens.

Der hier vorgestellte Aufbau der Infrastruktur wird zunächst für das weitere Design und die Realisierung zugrunde gelegt. In der Evaluation in Kapitel 5 wird diskutiert, welche alternativen Aufbaumöglichkeiten denkbar und bezüglich der Trainingseffektivität möglicherweise günstiger sind.

## 4.2 Grundlagen der Systemarchitektur

In diesem Unterkapitel werden einige grundlegende Aspekte der konkreten Systemarchitektur diskutiert, die im nachfolgenden Unterkapitel 4.3 vorgestellt wird. Zunächst wird in Abschnitt 4.2.1 das Entwurfsmuster Model-View-Controller und dessen Anwendbarkeit in der Spieleentwicklung erläutert. Im darauffolgenden Abschnitt 4.2.2 wird die allgemeine Funktionsweise des Client-Server-Modells skizziert und dessen Relevanz bei der Entwicklung von Mehrspielersystemen veranschaulicht.

### 4.2.1 MVC-Entwurfsmuster in der Spieleentwicklung

*Model-View-Controller* (MVC) ist ein Entwurfsmuster, das zum Ziel hat, die Präsentation eines Programms von seiner Logik und den Daten zu trennen. So soll vor allem die Wartbarkeit verbessert werden. Außerdem werden Projekte strukturierter und flexibler, da die Logik und die grafische Benutzerschnittstelle einfacher austauschbar sind ([Buschmann u. a., 1996](#)).

#### MVC in „klassischen“ Anwendungen

Das MVC-Entwurfsmuster organisiert ein System in drei Komponenten (siehe [Abb. 4.4](#), links):

- **Model:** Die Model-Komponente verwaltet die Daten der Anwendung und liefert diese größtenteils an die View-Komponente. Ferner nimmt sie aufgrund von Instruktionen, die von der Controller-Komponente gesendet werden, Änderungen an den Anwendungsdaten vor.
- **View:** Die View-Komponente ist für die grafische Präsentation der Anwendungsdaten zuständig, die von der Model-Komponente verwaltet werden.
- **Controller:** Die Controller-Komponente interpretiert Interaktionen des Anwenders, die mit Hilfe von Eingabegeräten, wie z.B. Maus oder Tastatur, erzeugt werden. Wenn Anwendungsdaten oder deren Darstellung sich aufgrund von Benutzereingaben ändern sollen, informiert der Controller die Model- und View-Komponente.

Üblicherweise spielen eine Control- und eine View-Komponente paarweise zusammen. Sie „kennen“ sich gegenseitig und beziehen sich auf dasselbe Model. Ein Model kann von mehreren solcher View-Control-Paare gesteuert werden.

#### MVC in Spielanwendungen

In Anlehnung an [Smed und Hakonen \(2003\)](#) lassen sich bei Computerspielen drei verschiedene Aufgabenmodule unterscheiden:

- **Spielzustand:** die Verwaltung aller Informationen in der Spielumgebung, z.B. Objekte oder Punktestände
- **Spielvisualisierung:** die grafische Darstellung der aktuellen Spielsituation
- **Spielmechanik:** die Manipulation von Informationen in der Spielumgebung unter definierten Spielregeln

Diese Aufgabenmodule lassen sich auf das MVC-Entwurfsmuster übertragen bzw. können durch dieses abgebildet werden (siehe [Abb. 4.4](#), rechts).

Der Spielzustand beschreibt die Anwendungsdaten des Spiels und entspricht damit dem Model. Die Spielvisualisierung korreliert mit dem View. Die Spielmechanik interpretiert Eingaben

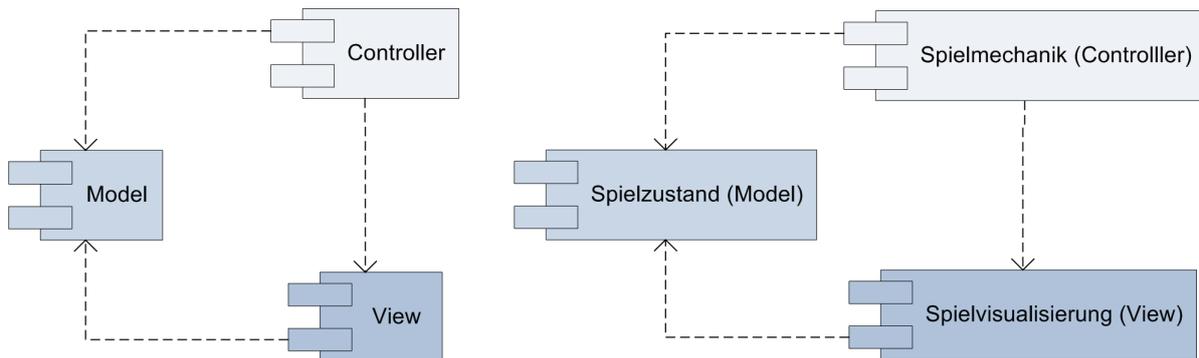


Abb. 4.4: MVC in der Spieleentwicklung: links die allgemeinen MVC-Komponenten sowie deren Zusammenspiel und rechts die Übertragung des Entwurfsmusters auf die Komponenten einer Spielanwendung

vom Benutzer bzw. Spieler, evaluiert diese mit den gegebenen Spielregeln und veranlasst darauf basierend Änderungen am Spielzustand. Damit entspricht dieses Aufgabenmodul dem Controller.

## 4.2.2 Client-Server-Modell in der Spieleentwicklung

Das Client-Server-Modell beschreibt eine Möglichkeit, Aufgaben und Dienstleistungen innerhalb eines verteilten Systems zu verteilen. Die Aufgaben werden von Systemkomponenten erledigt, die in *Clients* und *Server* unterteilt werden. Der Client kann auf Wunsch eine Aufgabe vom Server anfordern, wie z.B. ein Betriebsmittel. Der Server, der sich auf einem beliebigen anderen Rechner im verteilten System befindet, beantwortet die Anforderung, d.h. er stellt z.B. das gewünschte Betriebsmittel bereit.

### Allgemeines Client-Server-Modell

Das Client-Server-Modell ist das Standardkonzept für die Verteilung von Aufgaben innerhalb eines Netzwerkes. Aufgaben werden mittels Server auf verschiedenen Knoten in einem Netzwerk verteilt und können bei Bedarf von mehreren Clients zur Lösung ihrer eigenen Aufgaben angefordert werden. Bei den Aufgaben kann es sich um Standardaufgaben (z.B. E-Mail- oder Webseiten-Zugriffe) oder um spezifische Aufgaben eines Softwaresystems handeln. Eine Aufgabe wird im Client-Server-Modell als Dienst bezeichnet.

Die Abbildung 4.5 zeigt die Aufteilung der Verantwortlichkeiten in dem Client-Server-Modell. Ein Server ist ein System, das einen Dienst (Service) anbietet. Im Rahmen des Client-Server-Konzepts kann ein anderes System, der Client, diesen Dienst nutzen. Die Kommunikation zwischen Client und Server ist abhängig vom Dienst. Er bestimmt, welche Daten zwischen beiden ausgetauscht werden. Der Server ist in Bereitschaft, um jederzeit auf die Kontaktaufnahme eines Clients reagieren zu können. Im Unterschied zum Client, der aktiv einen Dienst anfordert,

verhält sich der Server *passiv* und wartet auf Anforderungen. Die Regeln der Kommunikation für einen Dienst (Format, Aufruf des Servers, und die Bedeutung der zwischen Server und Client ausgetauschten Daten), werden durch ein Protokoll festgelegt. Das Protokoll ist spezifisch für den jeweiligen Dienst.



Abb. 4.5: Allgemeine Rollenverteilung beim Client-Server-Modell

Clients und Server können als Programme auf verschiedenen Rechnern oder auf demselben Rechner ablaufen. Allgemein kann das Konzept zu einer Gruppe von Servern, die eine Gruppe von Diensten anbietet, ausgebaut werden (Mail-Server, Web-Server, Applikations-Server oder Datenbank-Server).

Da in der Praxis diese Server meist gesammelt auf bestimmten Rechnern laufen, hat es sich eingebürgert, diese Rechner selber als Server zu bezeichnen.

### Client-Server-Modell in Mehrspielersystemen

In der Spieleentwicklung wird das Client-Server-Modell üblicherweise zur Unterstützung von Mehrspielersystemen eingesetzt. Die Verteilung der Aufgaben und Verantwortlichkeiten lässt sich dabei in der Regel wie folgt aufschlüsseln:

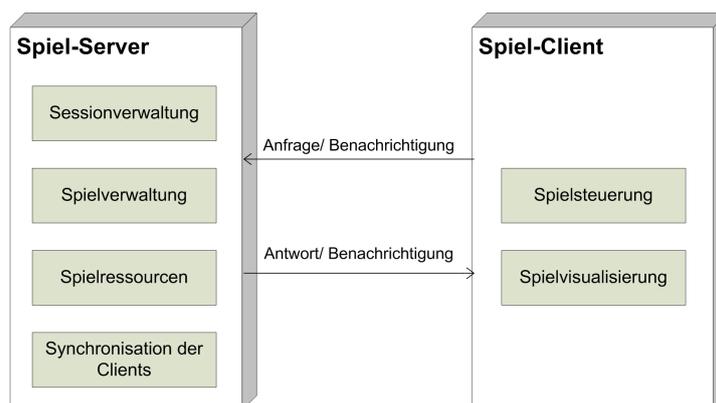


Abb. 4.6: Aufgaben von Client und Server im Kontext von mehrbenutzerfähigen Spielen

**Spieserver:** Der Spieserver stellt die zentrale Steuerungsinstanz des Spiels dar. Die Clients müssen sich zur Teilnahme an dem Spiel zunächst am Server anmelden. Der Server registriert diese Anmeldungen und überwacht alle Verbindungen zu den Clients in der *Sessionverwaltung*. Der eigentliche Spielablauf wird durch die *Spielverwaltung* des Servers kontrolliert. Der Server kann damit z.B. ein neues Spiel starten bzw. ein laufendes

beenden. Zudem liegt hier die eigentliche Anwendungslogik, d.h. die Regeln des Spiels, und das Zustandsmodell der Spielumgebung. Eine weitere Aufgabe des Servers ist die Verwaltung der *Spielressourcen*, d.h. alle Informationen und Daten, die für das Spiel notwendig sind. Hierbei kann es sich z.B. um Grafiken oder 3D-Modelle handeln, die für die Visualisierung des Spiels erforderlich sind. Als zentrale Anlaufstelle für alle Clients muss der Spielserver zudem dafür sorgen, dass alle Clients stets eine synchrone und konsistente Sicht auf den aktuellen Spielzustand haben.

**Spielclient:** Die Clients sind in der Regel für die *Spielsteuerung* und die *Spielvisualisierung* zuständig. Die Spielsteuerung umfasst alle Interaktionsmöglichkeiten, mit denen die Inhalte des Spiels bzw. der Spielzustand beeinflusst werden können. Die Spielvisualisierung entspricht einer grafischen Aufbereitung und Präsentation des aktuellen Spielzustands, der vom Server verwaltet wird. Zur Durchführung dieser Aufgaben werden die Dienste genutzt, die der Spielserver für diese Zwecke zur Verfügung stellt.

Die Abbildung 4.6 zeigt eine schematische Darstellung der typischen Aufgaben und der Kommunikation von Client und Server im Kontext von mehrbenutzerfähigen Spielen.

### Kommunikationsmodelle in Client-Server-Systemen

Für die Kommunikation zwischen Client und Server bzw. zwischen *Sender* und *Empfänger* stehen grundsätzlich das synchrone und das asynchrone Kommunikationsmodell zur Verfügung.

**Synchrone Kommunikation:** Bei der synchronen Kommunikation übermittelt der Sender eine Nachricht an den Empfänger und wartet anschließend auf eine Antwort von diesem. Während dieser Wartezeit blockiert der Sender. Synchrone Kommunikation wird in der Regel bei verteilten Systemen mit hoher Interaktionsrate und enger Kopplung genutzt.

**Asynchrone Kommunikation:** Bei der asynchronen Kommunikation wartet der Sender nicht auf eine Antwort des Empfängers und blockiert damit auch nicht. Sobald ihn die Antwort erreicht, wird er benachrichtigt. Sender und Empfänger lassen sich auf diese Weise entkoppeln. Dieses Kommunikationsmodell wird oft bei schlechter Netzwerkverbindung und hoher Ausfallwahrscheinlichkeit eingesetzt.

Synchrone Kommunikation lässt sich mittels asynchroner Kommunikation umzusetzen und umgekehrt. Im ersten Fall wartet der Sender explizit, bis der Empfänger eine Antwort auf die Nachricht sendet. Diese wird ebenfalls explizit übermittelt. Beim umgekehrten Fall werden die Nachrichten zwischen Sender und Empfänger durch einen zusätzlichen Prozess zwischenspeichert.

## 4.3 Systemarchitektur

In diesem Unterkapitel wird die Systemarchitektur der im Rahmen dieser Arbeit entwickelten *Co-located Collaborative Training Environment* (CCTE) besprochen. Zunächst wird dazu in Abschnitt 4.3.1 eine Übersicht des gesamten Systems geliefert. Dabei werden alle wesentlichen Subsysteme erläutert, die für die erwünschte Funktionalität der gesamten Umgebung erforderlich sind. Den Kern des CCTE bilden die spielbasierten Simulationen, auf deren Basis das eigentliche Training erfolgt. In Abschnitt 4.3.2 wird daher der Entwurf eines Basissystems für solche spielbasierten Simulationen vorgestellt. In den darauffolgenden Abschnitten (Abschnitte 4.3.3 - 4.3.5) werden die einzelnen Komponenten des Basissystems im Detail erläutert.

### 4.3.1 Systemübersicht

Die Abbildung 4.7 zeigt eine Übersicht aller erforderlichen Subsysteme im CCTE. Die Verbindungen zwischen den Subsystemen in Form der gestrichelten Pfeile sind nach UML-konformer Notation als „Quelle benutzt Ziel“ zu interpretieren. Die einzelnen Aufgaben und Verantwortlichkeiten dieser Subsysteme werden im Folgenden kurz vorgestellt:

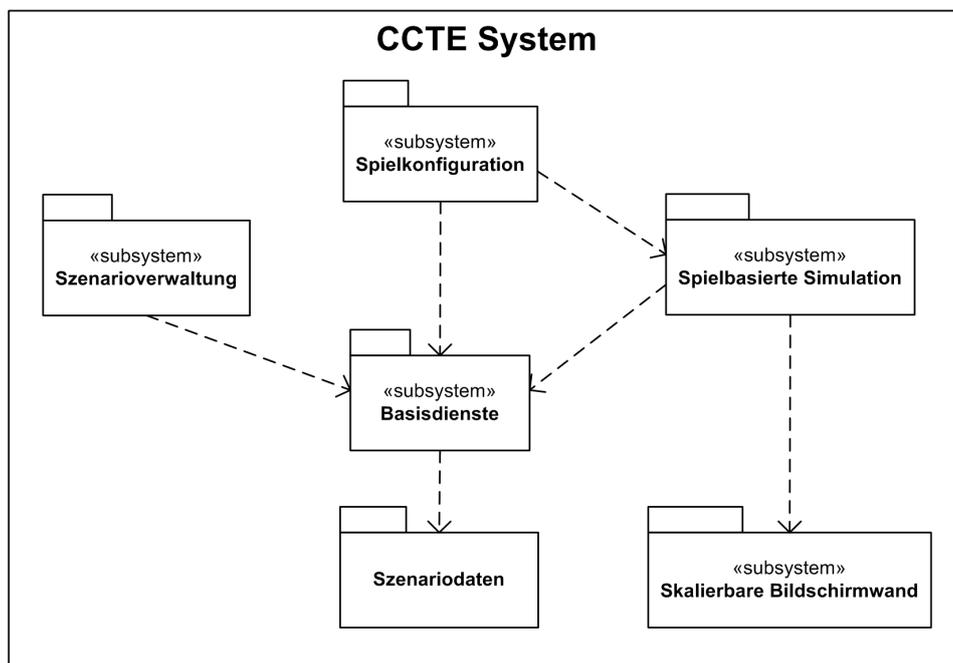


Abb. 4.7: Eine grobe Übersicht des CCTE-Systems

**Szenariodaten:** Diese Komponente enthält alle Informationen, die für die Initialisierung und Darstellung der Trainingsszenarien notwendig sind. Dazu zählen beispielsweise die verwendeten grafischen Modelle (z.B. Spieler- oder Gebäudemodelle) oder die Beschreibungen der Zusammensetzung von Trainingsszenarien.

**Basisdienste:** In diesem Subsystem werden zwei Dienste zusammengefasst:

- **Kommunikation:** Das CCTE ist ein verteiltes System und benötigt daher geeignete Methoden zur Kommunikation zwischen den einzelnen Instanzen des Systems. Dafür bietet dieser Basisdienst synchrone und asynchrone Kommunikationsmöglichkeiten an, wobei von den konkret verwendeten Technologien abstrahiert wird.
- **Szenariodatenzugriff:** Ein weiterer Bestandteil der Basisdienste ist der Zugriff auf die Szenariodaten. Auch hier wird eine Schnittstelle angeboten, die von der konkreten Zugriffstechnik auf die notwendigen Daten abstrahiert. Im Zusammenspiel mit dem Basisdienst *Kommunikation* wird damit auch für mehrere, im gesamten System verteilte Szenariodaten ein einheitlicher Zugriffspunkt für alle Subsysteme bereitgestellt.

**Szenarioverwaltung:** Hier werden verschiedene Werkzeuge für die Rolle des Ausbilders zur Verfügung gestellt, damit dieser neue Trainingsszenarien und die darin enthaltenen Objekte für die spielbasierte Simulation erstellen kann.

**Spielkonfiguration:** Dieses Subsystem stellt verschiedene Dienste bereit, um die eigentliche spielbasierte Simulation konfigurieren und starten zu können. Aus der Sicht der beteiligten Rollen Ausbilder, Einsatzleiter und Einsatzkraft werden hier die funktionalen Anforderungen „Training vorbereiten“ sowie „Training starten/beenden“ abgedeckt. Dazu müssen Kommunikationsverbindungen zwischen den einzelnen Instanzen des verteilten Systems aufgebaut und die spielbasierte Simulation mit den notwendigen Szenariodaten initialisiert werden. Die dafür erforderlichen Funktionalitäten werden über das Subsystem „Basisdienste“ genutzt.

**Spielbasierte Simulation:** Dieses Subsystem bildet den Kern des CCTE. Es beinhaltet die aktuelle Zustandsbeschreibung der virtuellen Trainingsumgebung, die Aktionsmöglichkeiten der beteiligten Akteure sowie die Visualisierung der Spielumgebung.

**Skalierbare Bildschirmwand:** Die skalierbare Bildschirmwand bildet ein eigenständiges Subsystem innerhalb des CCTE. Es verwaltet einen Rechnercluster mit den daran angeschlossenen Anzeigemedien und schließt diese zu einem einheitlichen Visualisierungskontext zusammen. Die spielbasierte Simulation nutzt die Dienste dieses Subsystems für die globale Sicht auf das virtuelle Modell im CCTE.

Die Subsysteme *Szenariodaten*, *Basisdienste*, *Szenarioverwaltung* und *Spielkonfiguration* werden zwar für die Gesamtfunktionalität des Systems benötigt, spielen jedoch für den eigentlichen Schwerpunkt in dieser Arbeit nur eine untergeordnete Rolle. Der Fokus des weiteren Systementwurfs liegt daher auf den beiden zuletzt aufgeführten Subsystemen *Spielbasierte Simulation* und *Skalierbare Bildschirmwand*.

### 4.3.2 Basissystem für spielbasierte Simulationen im CCTE

Als Grundlage für das CCTE wurde ein Basissystem konzipiert, in dem verschiedene spielbasierte Simulationen eingesetzt werden können. Das Basissystem wurde in Anlehnung an das zuvor dargestellte Model-View-Controller Entwurfsmuster gestaltet und besteht aus den Komponenten *Spielsimulation* (Model), *Spielvisualisierung* (View) und *Spielmechanik* (Controller) (Abb. 4.8).

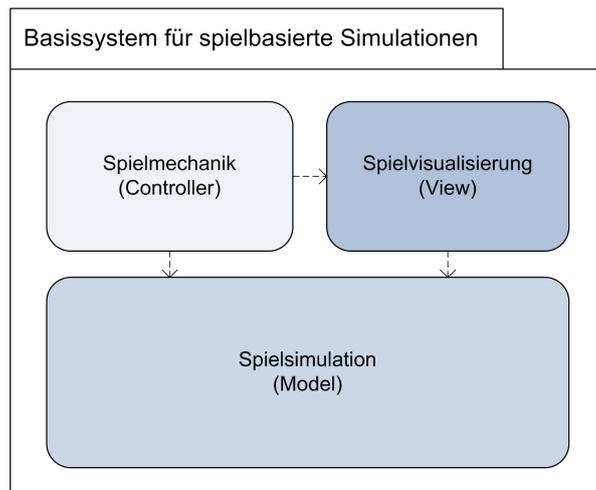


Abb. 4.8: Die Komponenten des Basissystems nach dem MVC-Entwurfsmuster

Im Folgenden wird kurz dargestellt, welche Aufgaben die drei Komponenten im Umfeld des CCTE übernehmen:

**Spielsimulation (Model):** Diese Komponente stellt ein Simulationssystem zur Verfügung, welches das Modell der virtuellen Trainingsumgebung des CCTE verwaltet. Sie enthält alle Informationen bzw. Objekte, die sich in dieser Umgebung befinden und zur Beschreibung des aktuellen Trainingsszenarios notwendig sind. Die Manipulation der Inhalte des Trainingsszenarios und damit die Veränderung des Spielzustands kann auf zwei Weisen erfolgen:

- **externe Manipulation:** Die Spielsimulation bietet nach aussen eine Schnittstelle an, um die enthaltenen Informationen und Objekte verändern zu können. So können z.B. Objekte hinzugefügt, bewegt oder entfernt werden. Auf diese Weise kann die Spielsimulation zu Beginn eines Trainings mit einem bestimmten Trainingsszenario konfiguriert werden. Weiterhin besteht damit während des Trainings die Möglichkeit die Spielumgebung interaktiv zu ändern, was für die Aufgaben der verschiedenen Rollen im CCTE erforderlich ist.
- **interne Manipulation:** Neben der Manipulation von außen kann die Spielsimulation zum Teil autonom über Änderungen des Spielzustands entscheiden, indem in dieser eine Reihe von logischen Zusammenhängen zwischen den Objekten und

physikalischen Gesetzmäßigkeiten, die in der virtuellen Trainingsumgebung Gültigkeit haben, definiert werden. Dazu zählen u.a. das sich Objektkörper mit fester Materie und Masse (z.B. Avatare) im dreidimensionalen Raum nicht „überschneiden“ können und dem Gesetz der Schwerkraft unterliegen. Interne Manipulation werden stets durch externe Manipulationen ausgelöst.

Die Spielsimulation muss zu jeder Zeit die Gültigkeit des Modells gewährleisten. Alle Manipulationen des Modells werden daher nach dem Transaktionskonzept für verteilte Systeme behandelt und müssen den sogenannten ACID-Eigenschaften genügen:

- **Atomicity:** Von einer atomaren Handlung spricht man, wenn eine Transaktion entweder ganz oder gar nicht ausgeführt wird. Eine Transaktion kann sich aus mehreren Manipulationsanweisungen zusammensetzen, wird jedoch als eine einzelne elementare Operation betrachtet, die nicht von anderen Operationen unterbrochen werden kann.
- **Consistency:** Konsistenz bedeutet, dass eine Transaktion nach Beendigung einen konsistenten Modellzustand hinterlässt, falls der Modellzustand vor der Transaktion auch konsistent war.
- **Isolation:** Durch das Prinzip der Isolation wird verhindert, dass sich in Ausführung befindliche Transaktionen gegenseitig beeinflussen. Realisiert wird dies beispielsweise durch spezielle Sperrprotokolle oder Zeitstempelverfahren.
- **Durability:** Das Ergebnis einer Transaktion ist dauerhaft. Die Auswirkungen einer erfolgreich abgeschlossenen Transaktion bleiben für die verbleibende Dauer des Spiels in dem Modell erhalten.

Dadurch kann sichergestellt werden, dass das Modell der Spielsimulation bei allen Manipulationen von einem konsistenten in einen anderen konsistenten Zustand überführt wird.

**Spielvisualisierung (View):** Diese Komponente umfasst sämtliche Visualisierungsformen, die für die Unterstützung der drei verschiedenen Rollen erforderlich sind. Dabei werden lokale und globale Sichtweisen unterstützt. Neben der Darstellung der dreidimensionalen Szene, also den „Fenstern“ zur virtuellen Trainingsumgebung, werden hier zusätzlich weitere grafische Elemente bereitgestellt, um den Spielern alle notwendigen Zustandsinformationen präsentieren zu können.

**Spielmechanik (Controller):** Die Spielmechanik stellt den Controller in der MVC-basierten Konzeption dieses Basissystems dar. In dieser werden alle konkreten Spielregeln der spielbasierten Simulation und die Interaktionsmöglichkeiten der beteiligten Rollen definiert. Externe Benutzereingaben werden im Rahmen der Spielmechanik interpretiert und gegen die Spielregeln evaluiert. Die auf diese Evaluation basierenden Änderungen des Spielzustands werden an die Spielsimulation als eine oder mehrere externe Manipulationen weitergereicht. Die Spielsimulation verarbeitet diese externen Manipulationen wie zuvor beschrieben nach dem Transaktionskonzept und passt den von ihr verwalteten Spielzustand entsprechend an.

Die drei Komponenten bilden abgeschlossene Module mit klar definierten Schnittstellen. Die hier gewählte Komponenteneinteilung verhält sich gegenüber der Änderbarkeit und Wartbarkeit des Systems sehr robust. Um beispielsweise ein neues Spielkonzept bzw. neue Spielregeln zu verwenden, bedarf es lediglich einer Änderung der Controller-Komponente. Für eine alternative Repräsentation der virtuellen Trainingsumgebung (z.B. als zweidimensionale Übersichtskarte) muss nur eine entsprechende neue View-Komponente bereitgestellt werden. Entsprechendes gilt für die Komponente *Model*: wie der Spielzustand konkret verwaltet wird, bleibt für die Spielmechanik und die Spielvisualisierung verborgen. Die internen Vorgänge der Spielsimulation lassen sich damit bei Bedarf ändern bzw. die Spielsimulation kann vollständig ausgetauscht werden, wenn die definierten Schnittstellen eingehalten werden.

### Unterstützung von rollenspezifischen Interaktionen und Visualisierungen

Wie zuvor erwähnt, spielen Controller und View in einem MVC-System üblicherweise paarweise zusammen und bilden gemeinsam die Benutzerschnittstelle zum Model. Auch in der Konzeption des Basissystems findet sich diese Struktur zur Unterstützung der drei beteiligten Rollen Ausbilder, Einsatzkraft und Einsatzleitung wieder.

Jede Rolle hat eine individuelle Sicht auf die virtuelle Trainingsumgebung und verschiedene Interaktionsmöglichkeiten, um auf den Spielzustand Einfluss zu nehmen. Daraus ergibt sich, dass alle drei Rollen zwar auf demselben Model (d.h. der Spielsimulation) arbeiten, dafür jedoch unterschiedliche Benutzungsschnittstellen benötigen. In dem hier dargestellten MVC-basierten Basissystem für spielbasierte Simulationen kann dies dadurch erreicht werden, dass für jede Rolle ein individuelles Control/View-Paar konzipiert wird, welches die erforderlichen Funktionalitäten zur Visualisierung und Interaktion mit der Spielsimulation bietet (siehe Abb. 4.9).

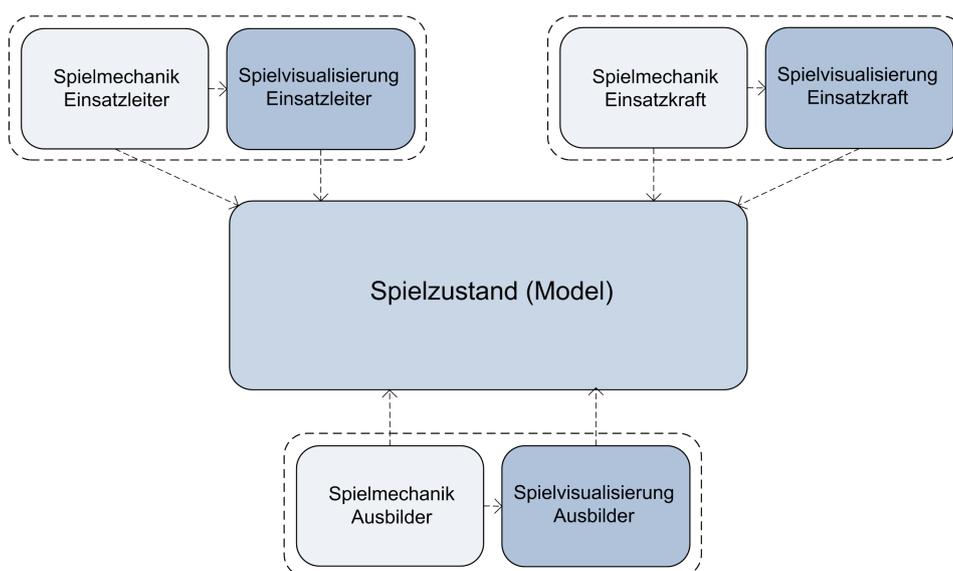


Abb. 4.9: Control/View-Paare bilden die rollenspezifischen Benutzerschnittstellen zum System

Die Summe aller Control/View-Paare charakterisiert demnach die konkrete spielbasierte Simulation, die zum Training der Gruppe eingesetzt wird.

### Verteilungsmöglichkeiten der MVC-Komponenten

Die zuvor dargestellten Control/View-Paare zur Unterstützung der Rollen sind im CCTE verteilt und greifen auf dasselbe Model zu. Um die Allgemeingültigkeit der Spielsimulation und damit eine konsistente Sichtweise aller Benutzerschnittstellen zu bewahren, muss die Konsistenz des Model durch eine zentrale Instanz kontrolliert werden.

In der Spieleentwicklung ist es diesbezüglich üblich, den Spielzustand in einen Serverknoten auszulagern (vgl. Abschnitt 4.2.2). Die Abbildung 4.10 zeigt eine solche Verteilung für die

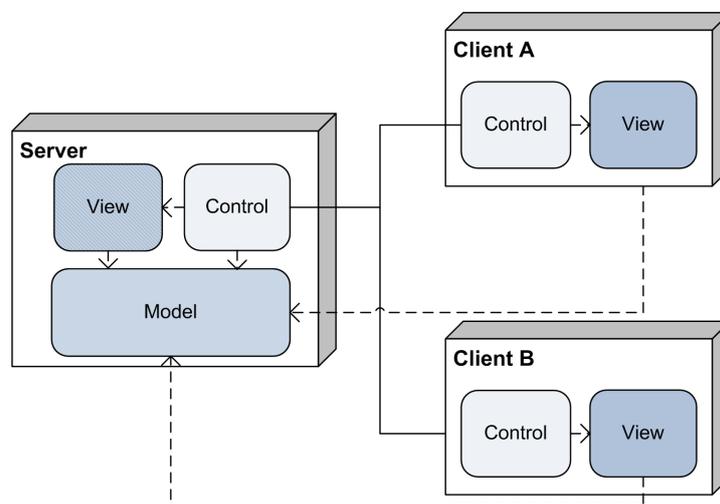


Abb. 4.10: Verteilungsmöglichkeit der MVC-Komponenten: eine zentrale Model-Komponente für alle Clients

MVC-Komponenten des Basissystems. Die Client-Knoten verfügen lediglich über Benutzerschnittstellen (Control/View-Paare) zum System, während das Model mit dem Spielzustand auf einen zentralen Server ausgelagert wird. Auch der Server verfügt über mindestens eine Controller-Komponente und (optional) über einen eigenen View. Einer dieser serverseitigen Controller fungiert als Vermittlungsstelle zwischen allen im System verteilten Controllern und dem Model. Hier laufen die Eingaben aller Benutzer zusammen und werden an das Model weitergeleitet. Die Kommunikation zwischen Controller (Client) und Controller (Server) sowie zwischen Model und View erfolgt dabei vollständig über das Netzwerk.

In der Regel verfügt der Server bei spielbasierten Simulationen über weitere Controller-Instanzen, da mit diesem z.B. computergesteuerte Mitspieler bzw. Gegner in das Spiel integriert werden können. Ein sogenannter KI-Controller funktioniert analog zu den „menschlichen“ Controller-Komponenten. Es werden Interaktionsmöglichkeiten definiert, die Änderungen des Spielzustands bewirken können. Zusätzlich werden bestimmte Verhaltensmuster integriert, wie und wann diese Interaktionsmöglichkeiten zu nutzen sind. Im Server können sodann eine oder

mehrere dieser KI-Kontroller instanziiert und gestartet werden. Die Zustandsänderungen, die aus diesen computergesteuerten Aktionen hervorgehen, werden aus Sicht des Model genauso verarbeitet wie die Aktionen der „menschlichen“ Benutzer.

Der Nachteil der Verteilung mit einem zentralen Model ist die Netzwerkbelastung bei der Kommunikation zwischen den Komponenten. Der Kommunikationsaufwand zwischen Controller (Client) und Controller (Server) ist unkritisch. Hier werden keine komplexen Datenstrukturen ausgetauscht, sondern im Wesentlichen nur entfernte Funktionsaufrufe getätigt<sup>2</sup>.

Bei der Kommunikation zwischen dem Model und dem View gestaltet sich die Situation etwas anders: das Model enthält die komplette Szenenbeschreibung aller Objekte der virtuellen Umgebung, die für die Berechnung der 3D-Szenen im View benötigt wird. Da es sich bei diesen nicht um eine statische, sondern um eine bewegte Szenerie handelt, müssen nach [Rumbke \(2006\)](#) mindestens 25-30 Bilder pro Sekunde berechnet und dargestellt werden, um dem Benutzer ein flüssiges Bild zu bieten. Damit die Steuerung nicht beeinträchtigt wird, empfiehlt sich sogar ein Wert von 60 Bildern pro Sekunde. Die Anzahl der erforderlichen Bilder pro Sekunde gibt einen unmittelbaren Rückschluss darauf, wie oft der View die aktuelle Szenenbeschreibung vom Model anfordern muss. Für größere Szenarien mit vielen Objekten müssen hier bei jeder Anfrage umfangreiche Datenstrukturen übermittelt werden, womit das Netzwerk kontinuierlich sehr stark belastet wird und damit keine Skalierbarkeit des Systems gewährleistet werden kann.

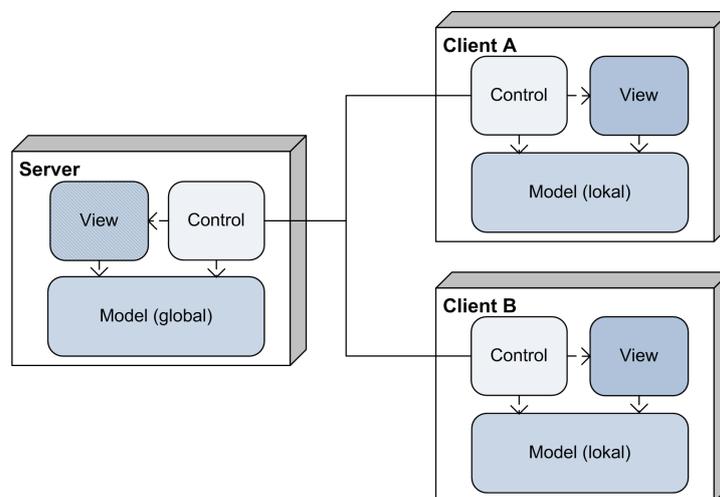


Abb. 4.11: Verteilungsmöglichkeit der MVC-Komponenten: alle Clients verfügen über eine eigene Model-Komponente

Eine Lösung zu diesem Skalierungsproblem besteht darin, die Netzwerkkommunikation zwischen Model und View zu eliminieren. Dazu verfügt jeder Knoten im verteilten System, d.h. der Server und alle Clients, über eine eigene Version des Model (siehe Abb. 4.11). Die Synchronität und Konsistenz aller verteilten Model wird über eine Eingabesynchronisation (Abschnitt 3.3.2)

<sup>2</sup>Details zu der Controller-Controller-Kommunikation werden in Abschnitt 4.3.5 diskutiert

gewährleistet. Alle Eingaben laufen im Server zusammen, werden gegebenenfalls überarbeitet (vgl. Abschnitt 4.3.5) und an alle involvierten Clients propagiert. Der Server sorgt dafür, dass alle Clients die Eingaben in derselben Reihenfolge erhalten. Um dies sicherstellen zu können, ist ein zuverlässiger Kommunikationsdienst erforderlich. Die Clients wenden alle auf diese Weise erhaltenen Eingaben auf ihr jeweiliges Model und haben damit den aktuellen Spielzustand jederzeit lokal verfügbar.

Der Server besitzt in diesem Rahmen ein Model von *globaler Gültigkeit*, d.h. es dient als zentrale Referenz für den gesamten Spielfortschritt. Das Servermodel wird z.B. zur Feststellung von Gewinn bzw. Niederlage oder zur Initialisierung des Model eines neu hinzugekommenen Clients verwendet. Dahingegen verfügen die Clients über Models mit *lokaler Gültigkeit*, d.h. sie dienen lediglich dazu, den aktuellen Spielzustand für den lokalen View bereitzuhalten, haben jedoch keinen Einfluss auf den globalen Spielfortschritt.

Eine wesentliche Voraussetzung für diese alternative Verteilungsmöglichkeit ist, dass sich das Model, also die Spielsimulation, deterministisch verhält. Gleiche Eingaben müssen die gleichen Zustandsänderungen in allen verteilten Spielsimulationen hervorrufen. Dies impliziert auch, dass aus allen externen Manipulationen die gleichen internen Manipulationen resultieren (vgl. Abschnitt 4.3.2).

### Detaillierte Teilkomponenten des Basissystems

Jede Komponente besteht aus weiteren Teilkomponenten, die konkrete Aufgaben und Verantwortlichkeiten innerhalb des Systems übernehmen. Die Abbildung 4.12 zeigt eine Übersicht aller wesentlichen Teilkomponenten und deren Abhängigkeiten. Die Unterstützung mehrerer Spieler durch eine synchrone Visualisierung der virtuellen Trainingsumgebung, und der damit verbundenen Eingabesynchronisation, wurde in dieser Darstellung aus Übersichtsgründen noch nicht explizit berücksichtigt.

Auf die Verarbeitung und Ausgabe von Soundeffekten wurde in dieser Konzeption vollständig verzichtet, da diese für den schwerpunktmäßigen Untersuchungsgegenstand dieser Arbeit irrelevant sind. Die Integration von Soundeffekten ist jedoch nicht generell auszuschließen. So können z.B. Umgebungsgeräusche dazu beitragen, die realistische Wirkung einer Simulation zu erhöhen.

In den folgenden Abschnitten werden die Systemkomponenten Spielzustand (Model), Spielvisualisierung (View) und Spielmechanik (Controller) ausführlicher dargestellt. Dabei werden alle Teilkomponenten aus Abbildung 4.12 sowie deren Zusammenhänge erläutert.

#### 4.3.3 Komponente Spielsimulation (Model)

Die Komponente Spielsimulation des Basissystems setzt sich aus folgenden Teilkomponenten zusammen:

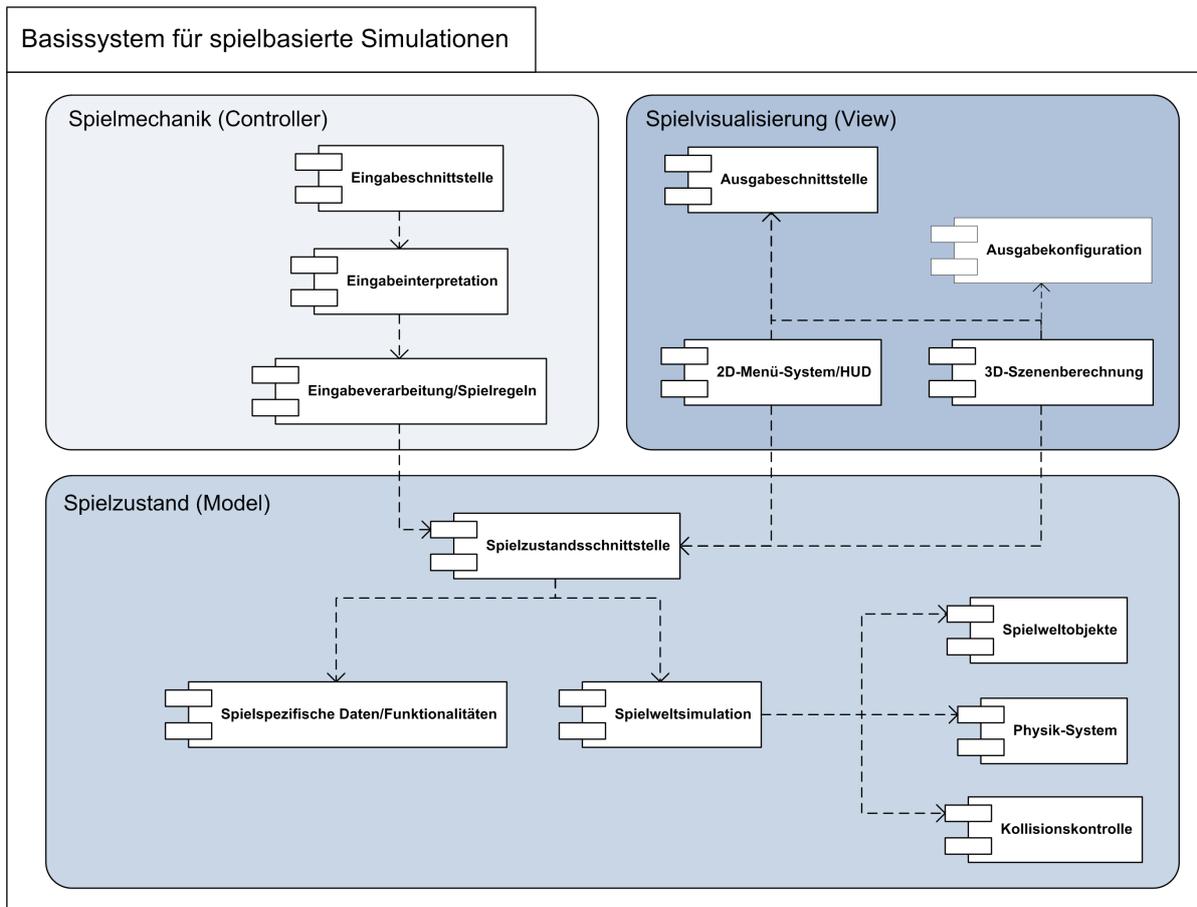


Abb. 4.12: Wesentliche Architekturkomponenten des Basissystems für spielbasierte Simulationen

**Spielzustandsschnittstelle:** Diese Komponente dient als einheitliche Schnittstelle zum Modell für Controller und View. Sie verbirgt damit die „darunterliegende“ Komplexität der Teilsysteme. Die konzeptionelle Vorlage dieser Komponente ist das Entwurfsmuster *Fassade* (Gamma u. a., 1995). Die Erweiterbarkeit des Systems und insbesondere des Modells um weitere, als nur die in der Spielzustandsschnittstelle definierten, Funktionalitäten lassen sich durch das objekt-orientierte Vererbungsprinzip gewährleisten. Dieser Aspekt wird im Anschluss an die Erläuterung der Teilkomponenten expliziter diskutiert.

**Spielspezifische Daten/ Funktionalitäten:** Das Basissystem wurde so konzipiert, dass eine größtmögliche Wiederverwendbarkeit für verschiedenartige spielbasierte Simulationen besteht. Das Modell bietet dahingehend Möglichkeiten eine virtuelle Trainingsumgebung sehr allgemein zu beschreiben (z.B. Objekt hinzufügen, Objekt entfernen oder Objekt bewegen). Dennoch bringt jedes konkrete Spielprinzip in der Regel zusätzliche Daten und Funktionalitäten mit sich, die vom Modell verwaltet bzw. bereitgestellt werden müssen, um den vollständigen Spielzustand beschreiben zu können. Die beiden Teilkomponenten *spielspezifische Daten* und *spielspezifische Funktionalitäten* dienen daher in dieser Übersicht als „Platzhalter“ für konkrete Komponenten, die von „Spiel zu Spiel“

unterschiedlich ausfallen können. Im Zusammenhang mit der Erweiterbarkeit des Systems wird dieser Punkt ebenfalls im Anschluss an die Erläuterung der Teilkomponenten konkreter ausgeführt.

**Spielweltobjekte:** Die Spielweltobjekte stellen die konkreten Inhalte eines Szenarios dar. Dabei kann es sich beispielsweise um Spieler-, Kamera-, Gebäude- oder Interaktionsobjekte handeln. Zu der allgemeinen Beschreibung eines Objekts gehören u.a. die Koordinaten und Ausrichtung in einem dreidimensionalen Raum sowie grundlegende physikalische Eigenschaften. Dazu zählen z.B. die Größe bzw. die Ausmaße und die Masse des Objekts. Die konkrete geometrische Beschaffenheit, d.h. das 3D-Modell für die Darstellung des Objekts, wird in dem Model noch nicht definiert. Die Zuordnung von Spielweltobjekten zu ihren visuellen Repräsentationen erfolgt in der Komponente *View* des Basissystems. Über die allgemeine Beschreibung hinaus kann jedes Spielweltobjekt über weitere spezifische Eigenschaften verfügen. Ein Spielerobjekt kann beispielsweise eine Eigenschaft „Gesundheitszustand“ mit sich führen, während für ein Kameraobjekt die aktuelle „Brennweite“ relevant ist.

**Physik-System:** Die Spielweltobjekte für sich allein genommen, stellen lediglich Objekte in einem dreidimensionalen Raum dar. Damit die virtuelle Trainingsumgebung ein für den Menschen realistisch wirkendes Umfeld darstellt, müssen vertraute physikalische Gesetzmäßigkeiten berücksichtigt werden. Diese Aufgabe wird von dem Physik-System übernommen. Die Komplexität eines solchen Systems kann stark variieren. Eine grundlegende Funktionalität bildet jedoch die Berücksichtigung der Zusammenhänge zwischen einer „Erdschwerkraft“ und der physischen Masse von Objekten. Damit kann beispielsweise gewährleistet werden, dass Spielerfiguren auf den horizontalen Oberflächen der virtuellen Umgebung „laufen“ können. Darüberhinaus lassen sich weitere physikalische Einflüsse integrieren, wie z.B. Reibungseffekte oder Wiedereinflüsse (Gegenwind, nasse Oberflächen,...). In dieser Konzeption wird nur die grundlegende Funktionalität berücksichtigt. Für zukünftige Arbeiten, die ggfs. realistischere physikalische Simulationsbedingungen erfordern, besteht hier jedoch die Möglichkeit sogenannte *Physik-Engines* einzubinden. Dabei handelt es sich um Systeme bzw. Softwarebibliotheken, die sich auf realistische physikalische Effekte spezialisiert haben. Eine exemplarische Anwendung solcher Systeme findet sich in [Roßberger \(2008\)](#).

**Kollisionskontrolle:** Ein wesentliches Element in einer interaktiven, virtuellen Umgebung ist die Möglichkeit, die Berührung bzw. Überschneidung von Spielweltobjekten feststellen zu können. Aus einer solchen Kollision können dann entsprechende Interaktionsmöglichkeiten oder Verhaltensweisen abgeleitet werden, die auf die Spielweltobjekte zu übertragen sind. Wenn sich beispielsweise ein Objekt „Spielerfigur“ orthogonal auf ein Objekt „Wand“ zubewegt und die beiden Objekte kollidieren, bedeutet das in der Regel, dass sich das Objekt „Spielerfigur“ nicht weiter vorwärts bewegen kann, auch wenn über den Controller entsprechende Eingaben gemeldet werden. Zur Feststellung einer Kollision werden die Ausmaße zugrundegelegt, die für jedes Spielweltobjekt definiert wurden. Diese Ausmaße werden oftmals auch als *Hüllkörper* (bounding volumes) bezeichnet. Anzumerken ist,

dass ein Hüllkörper keine Vorgaben für das in der View verwendete 3D-Modell eines Objekt macht. Für eine, dem Benutzer realistisch erscheinende, Visualisierung ist allerdings empfehlenswert, wenn sich das gewählte 3D-Modell exakt in den vorgegebenen Hüllkörper einfügt.

**Spielweltsimulation:** Die Spielweltsimulation ist für den Aufbau und die Organisation aller Objekte in der virtuellen Umgebung zuständig. Sie manipuliert die Objekte nicht nur auf Eingaben von außen, sondern kann unter Nutzung des Physiksystems und der Kollisionskontrolle „eigenständig“ entscheiden, welche (zusätzlichen) Veränderungen an den Objekten vorzunehmen sind, um ein *plausibles Verhalten* der virtuellen Umgebung zu gewährleisten. Wird beispielsweise ein Objekt „Kugel“ durch den Spieler mit einer bestimmten Kraft angestossen, sorgt die Spielweltsimulation dafür, dass die Kugel solange vorwärts rollt, bis der einwirkende Kraftimpuls durch die Reibungseffekte „aufgebraucht“ wurde bzw. auf ein Hindernis trifft.

Um manipulative Operationen zu beschleunigen und eine effiziente Szenenberechnung im View zu ermöglichen, werden alle Spielweltobjekte in einem sogenannten Szenengraphen verwaltet. Ein Szenengraph ist eine objekt-orientierte Datenstruktur, mit der die logische und räumliche Anordnung der darzustellenden dreidimensionalen Szene beschrieben werden kann. Aus graphentheoretischer Sicht ist ein Szenengraph ein zusammenhängender, einseitig gerichteter, azyklischer Graph, dessen Wurzelknoten die Gesamtszene, also hier die gesamte virtuelle Trainingsumgebung enthält. Dieser Wurzel untergeordnet sind die Kindknoten, die die einzelnen Spielweltobjekte der Szene enthalten. Dieser Ansatz ermöglicht die hierarchische Modellierung der Objekte in der Szene.

Jeder Knoten des Szenengraphen hat üblicherweise eine Transformationsmatrix. Bei Manipulation dieser Matrix wird das zugehörige Objekt selbst, aber auch die Objekte aller untergeordneten Knoten transformiert. Man unterscheidet in diesem Fall zwischen Objektkoordinaten (Koordinaten eines Objektes bezüglich des übergeordneten Objektes) und Weltkoordinaten (Koordinaten eines Objektes bezüglich des Ursprungs der Gesamtszene). Durch diese hierarchische Sicht wird der Aufbau und die Manipulation der Szene deutlich vereinfacht. Man muss nicht jedes Einzelteil eines Objektes einzeln transformieren, sondern transformiert einfach die Gesamtheit aller Einzelteile.

Die Komponente *Spielsimulation* ist für alle Knoten, d.h. Server und Clients, im verteilten System des CCTE gleich. Die Initialisierung der Spielsimulation erfolgt durch die Systemkomponente *Spielkonfiguration* (Abschnitt 4.3.1), wenn der Server gestartet wird bzw. wenn ein Client der spielbasierten Trainingsumgebung beitrifft. Die Konfiguration erfolgt ebenfalls über die Spielzustandsschnittstelle, indem alle notwendigen Informationen, z.B. Objekte in der Umgebung oder initiale Punktstände, als eine Folge von externen Manipulationen an das Model delegiert werden. Die gesamte Initialisierung der Spielsimulation wird dabei als eine atomare Handlung betrachtet (vgl. Abschnitt 4.3.2).

## Erweiterbarkeit und Änderbarkeit der Spielsimulation

Die Spielzustandsschnittstelle stellt wie zuvor dargelegt den exklusiven Zugriffspunkt zur Spielsimulation dar und maskiert damit die zugrundeliegende Funktionalität des Modells. Sie definiert eine Schnittstelle für alle minimal notwendigen Dienste, die die Spielsimulation für den Controller und den View im Basissystem leisten muss.

In der Konzeption ist dafür eine Basisimplementierung vorgesehen, die eine konkrete Realisierung der erforderlichen Dienste darstellt. Wenn das Model für ein Spiel zusätzliche Daten verwalten oder zusätzliche Funktionalitäten anbieten muss, kann die Basisimplementierung bei Bedarf nach dem Vererbungsprinzip spezialisiert werden. Sofern die interne Funktionalität des Modells geändert werden muss, z.B. weil eine alternative Datenstruktur für Szenenbeschreibung verwendet werden soll, lässt sich auch die Basisimplementierung austauschen. Hierbei muss lediglich die bestehende Schnittstellendefinition eingehalten werden. Auf diese Weise bietet die Spielsimulation ein hohes Maß an Erweiterbarkeit und Änderbarkeit, was hinsichtlich der Adaption für unterschiedliche Spielkonzepte und einer möglichen Weiterentwicklung von Vorteil ist.

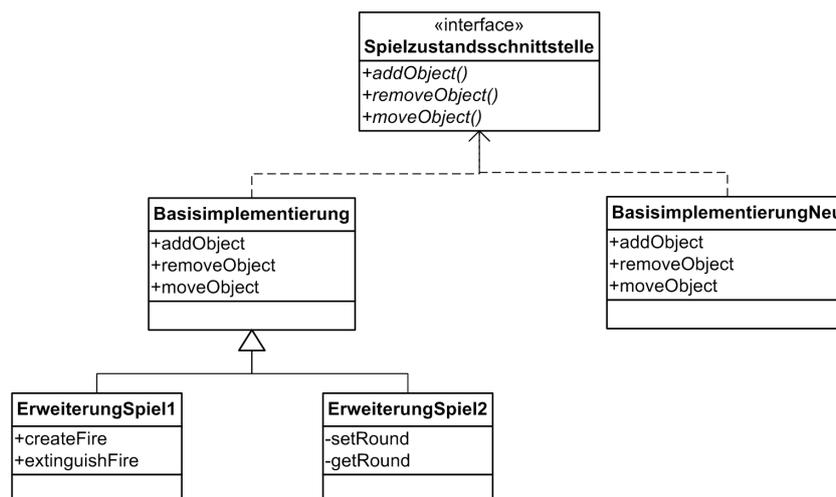


Abb. 4.13: Spielzustandsschnittstelle als Fassade zum Model: Erweiterbarkeit und Änderbarkeit der Spielsimulation durch Spezialisierung bzw. Austausch der Basisimplementierung

Die Abbildung 4.13 zeigt eine exemplarische statische Struktur dieser Konzeption. Die Schnittstelle definiert einige Basisdienste (`addObject`, `removeObject`, `moveObject`), die von der Klasse *Basisimplementierung* konkret realisiert werden. Die beiden exemplarischen Klassen *ErweiterungSpiel1* und *ErweiterungSpiel2* erweitern diese Klasse um zusätzliche Funktionalitäten, ohne die Schnittstellendefinition zu verletzen. Die Klasse *BasisimplementierungNeu* steht stellvertretend für den Fall, dass die Basisimplementierung ausgetauscht werden muss, weil sich die zugrundeliegende Funktionalität geändert hat. Auch hier bleibt die Schnittstelle davon unbeeinträchtigt.

### 4.3.4 Komponente Spielvisualisierung (View)

Die Komponente Spielvisualisierung des Basissystems setzt sich aus folgenden Teilkomponenten zusammen:

**2D-Menü-System/HUD:** Neben der Visualisierung der dreidimensionalen Szene sind für die Benutzer zur Steuerung der spielbasierten Simulation in der Regel die Darstellung weiterer spielspezifischer Informationen (z.B. Punktstände) und Bedienelemente notwendig (z.B. ein Chat-Fenster). Diese Komponente stellt daher eine Reihe von (zweidimensionalen) grafischen Elementen bereit, mit denen sogenannte Head-Up Displays (HUD) erstellt werden können, die über das berechnete Bild der 3D-Szene „gezeichnet“ werden können. Die grafische Benutzerschnittstelle kann auf diese Weise mit Informationen und Interaktionsmöglichkeiten angereichert werden, ohne dabei die Sicht auf die eigentliche Szene einzuschränken.

**Ausgabekonfiguration:** Die Ausgabekonfiguration wird in der Startphase eines Spiels bzw. eines Trainings durch die Komponente *Spielkonfiguration* initialisiert. Sie enthält Informationen zu der Repräsentation der Spielweltobjekte, d.h. welche Geometriedaten und Oberflächentexturen für die Spielweltobjekte zu verwenden sind. Die Repräsentationen werden durch 3D-Modelle beschrieben, welche zuvor mit einer entsprechenden 3D-Modellierungs-Software (z.B. Autodesk 3ds Max) erstellt wurden. Die 3D-Modelle werden extern in der Komponente *Szenariodaten* gelagert und bei der Initialisierung eines Spiels mit den Spielweltobjekten des Model verknüpft (vgl. Abschnitt 4.3.1). Die Trennung der Spielweltobjekte von ihren Repräsentationen ermöglicht z.B. den Austausch der geometrischen Beschreibung einer Spielerfigur oder die Verwendung von unterschiedlichen 3D-Modellen für zwei verschiedene Spielerfiguren.

**3D-Szenenberechnung:** Diese Komponente ist für kontinuierliche Berechnung der aktuellen Szene zuständig. Dazu werden zunächst die Positionen und Ausrichtungen der Spielweltobjekte und der zu verwendenden Kamera, die einem Spieler zugeordnet ist, vom Model angefordert. Aufgrund der zugrundeliegenden Szenengraph-Datenstruktur können Spielweltobjekte die außerhalb des Sichtbarkeitsbereiches der Kamera liegen, schon durch das Model herausgefiltert werden. Die Positions- und Rotationsinformationen der zu berechnenden Spielweltobjekte werden anschließend um ihre, aus der Ausgabekonfiguration bekannten, Repräsentationen ergänzt. Zur Berechnung des aktuellen Bildes werden die grafischen Primitiven, aus denen sich die 3D-Modelle zusammensetzen, mit der jeweiligen Kameraeinstellung an die Ausgabeschnittstelle des Systems weitergeleitet.

**Ausgabeschnittstelle:** Die Ausgabeschnittstelle wird zur Ansteuerung des Grafik-Subsystems (z.B. Grafikkarte und Monitor) verwendet, dass für die eigentliche Ausgabe der Visualisierung zuständig ist. Dafür werden bestehende Standard-Spezifikationen verwendet, die von den konkreten Hardwareschnittstellen abstrahieren und damit die Portabilität des Systems verbessern. Die beiden gängigsten Grafik-APIs für diese Zwecke sind *OpenGL*

und *DirectX*. OpenGL ist plattformunabhängig, während die Nutzung von DirectX ausschließlich auf Microsoft-basierten Plattformen möglich ist.

Obwohl alle drei Rollen im CCTE unterschiedliche Sichten auf die virtuelle Trainingsumgebung haben (vgl. Abschnitt 4.1.2), ist der hier dargestellte Aufbau der View-Komponente auf dieser Ebene des Systementwurfs in allen Fällen grundsätzlich gleich. Die Unterschiede ergeben sich auf Software-Ebene erst bei der eigentlichen Gestaltung der Benutzeroberfläche in der Systemrealisierung (d.h. der Komposition der darstellbaren Bildelemente aus den Komponenten 3D-Szenenberechnung und 2D-Menü-System/HUD) und bei der Konfiguration der Kameraperspektive zur Laufzeit des Systems (z.B. Vogelperspektive beim Einsatzleiter und Ego-Perspektive bei den Einsatzkräften).

Die tatsächlichen Unterschiede bei der Spielvisualisierung sind im CCTE auf Hardware-Ebene bei den verwendeten Anzeigemedien anzufinden. Wie in Abschnitt 4.1.2 dargestellt, nutzen die Einsatzkräfte und der Ausbilder „klassische“ Monitore. Der Einsatzleiter hingegen benötigt ein sehr großes und hochauflösendes Anzeigemedium, um seinen Aufgaben nachgehen zu können. Im Folgenden wird daher ein Systementwurf für eine skalierbare Bildschirmwand (Powerwall) dargelegt und in die zuvor dargelegte Konzeption des Basissystems integriert.

### **Grundlagen der Konzeption einer skalierbaren Bildschirmwand (Powerwall)**

In der Analyse dieser Arbeit wurden Strategien und Verfahren zur skalierbaren Visualisierung vorgestellt. Dabei wurden unter anderem invasive und nicht-invasive Realisierungskonzepte diskutiert (Abschnitt 3.3.4). Als Ansatz für den Systementwurf der skalierbaren Bildschirmwand in dieser Arbeit wird ein nicht-invasives Realisierungskonzept verfolgt. Dieser Entschluss ergab sich insbesondere aus dem Bestreben, das Visualisierungssystem im Ambient Labor der HAW Hamburg auch für weiterführende Arbeiten verfügbar zu machen. Hierfür wird die Strategie der Szenensynchronisation eingesetzt, wobei konkret das Verfahren des Sort-First/Tiled Rendering zum Tragen kommt (Abschnitt 3.3.3).

Im Folgenden wird zunächst gezeigt, wie sich dieses Verfahren allgemein in einem nicht-invasiven Realisierungskonzept eingliedern lässt. Im Anschluss daran wird ein konkretes Systemkonzept vorgestellt, das für die Nutzung im CCTE eingesetzt werden kann.

### **Sort-First/Tiled Rendering in einem nicht-invasiven Realisierungskonzept**

Die Abbildung 4.14 zeigt eine schematische Darstellung der typischen Schritte in der Grafikpipeline, die bei der Berechnung einer Szene mit dreidimensionalen Objekten notwendig sind:

1. Anwendung: Die Anwendung (z.B. die spielbasierte Simulation im CCTE) verwaltet alle Objekte in der Szene. Hier werden Änderungen an der Szene vorgenommen, wie sie zum Beispiel aufgrund der externen Manipulationen durch die Benutzer oder durch interne Manipulationen, die autonom vom System veranlasst werden, notwendig sind. Die

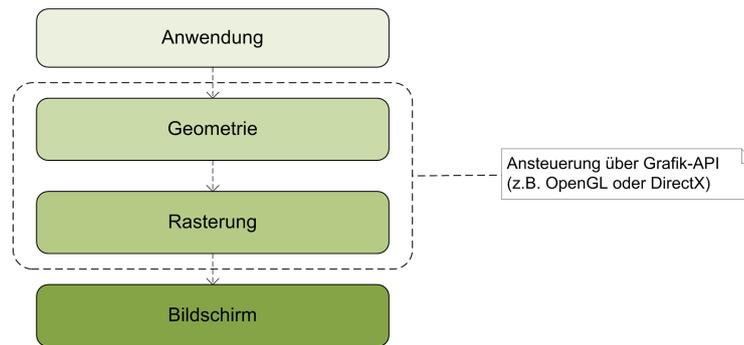


Abb. 4.14: Schematische Darstellung der Schritte in einer Grafikpipeline

- Anwendung veranlasst ein Rendering der aktuellen Szene, indem eine Szenenbeschreibung mit allen vorhandenen Grafikprimitiven an die Geometriebearbeitung weitergeleitet wird. Bei diesen Grafikprimitiven handelt es sich in der Regel um Dreiecke, Linien und Punkte.
2. Geometrie: Die Geometriebearbeitung erhält von der Anwendung eine Szenenbeschreibung in Form von Grafikprimitiven, die wiederum durch eine Reihe von Koordinaten in einem dreidimensionalen Raum beschrieben werden. Neben den Grafikprimitiven definiert die Szenenbeschreibung auch eine virtuelle Kamera oder einen Betrachter, der die Position und Blickrichtung angibt, aus der die Szene gerendert werden soll. Während der Geometriebearbeitung werden auf Basis der Grafikprimitiven und der virtuellen Kamera die Objektpositionen und der Einfluß von Oberflächenmaterialien sowie Lichtquellen berechnet. Außerhalb des sichtbaren Bereichs liegende grafische Primitive werden verworfen.
  3. Rasterisierung: Bei der Rasterisierung werden die Sichtbarkeiten und Farben von Bildschirmpunkten bestimmt, die auf den Ergebnissen der Geometriebearbeitung basieren. Dazu ist bei überlappenden Polygonen die Ermittlung der jeweiligen Sichtbarkeit notwendig. Die Farbe eines Pixels hängt von der Beleuchtung, Textur und anderen Materialeigenschaften des sichtbaren Polygons ab und wird oft anhand der Dreieckseckpunkte interpoliert.
  4. Bildschirm: Am Ende der Grafikpipeline steht ein vollständig berechnetes Bild der Szene, das auf dem Bildschirm ausgegeben werden kann.

Wie in Abschnitt 3.3.3 erläutert wurde, bedeutet Sort-First bei verteilten Visualisierungssystemen, dass zu Beginn der Geometriebearbeitung jedes graphische Objekt einem festgelegten Bildbereich zugeordnet wird. Jeder Rechner im Cluster ist für einen Bildbereich zuständig und berechnet nur die Objekte, die innerhalb seines Zuständigkeitsbereiches liegen.

In der in Abbildung 4.14 dargestellten Grafikpipeline ist dieses Verfahren demnach zwischen *Anwendung* und *Geometrie* zu platzieren. Die Ansteuerung der Grafikpipeline aus der Anwendung heraus, erfolgt üblicherweise über eine Grafik-API, die eine plattformübergreifende Standardschnittstelle anbietet (in der Regel OpenGL oder DirectX).

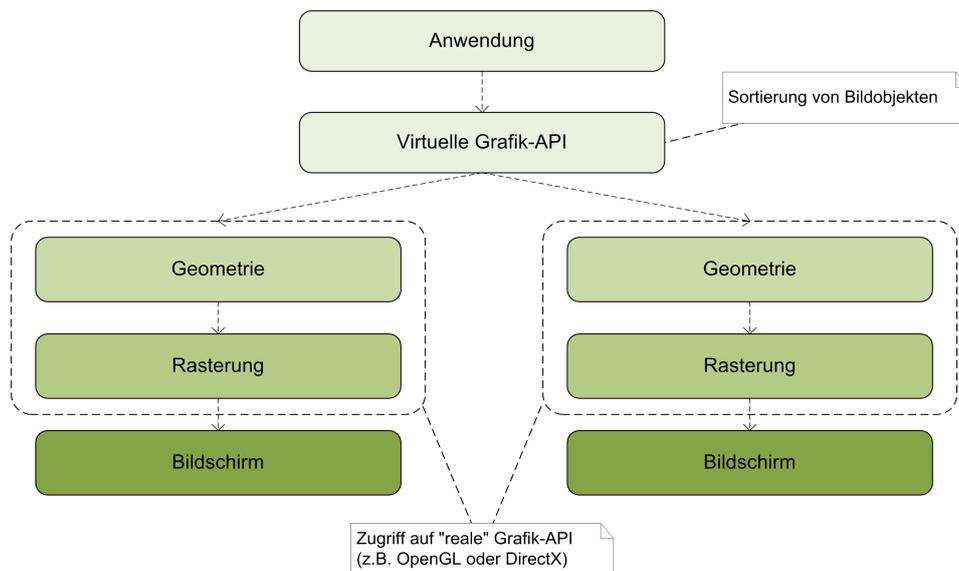


Abb. 4.15: Virtuelle Grafik-API zur Aufteilung der Grafikkette nach dem Sort-First-Verfahren

Eine für die Anwendung transparente Integration des Sort-First-Verfahrens, und damit ein nicht-invasives Konzept, ist möglich, wenn die von der Anwendung genutzte Schnittstelle der Grafik-API erhalten bleibt. Die Abbildung 4.15 zeigt eine schematische Übersicht dieser Vorgehensweise. Dort wird in die Grafikkette mit der virtuellen Grafik-API eine zusätzliche Komponente eingefügt. Diese Komponente bietet nach außen, also zur Anwendung, die gleiche Schnittstellenspezifikation wie die eigentlich eingesetzte Grafik-API (z.B. OpenGL). Die virtuelle Grafik-API beinhaltet die Logik des Sort-First-Verfahrens. Hier werden die Bildobjekte sortiert und auf die verschiedenen Knoten des Systems verteilt. Auf den jeweils zuständigen Knoten erfolgt anschließend die Berechnung der Teilszenen, die zusammengenommen das Bild der von der Anwendung beschriebenen Gesamtszene ergeben.

### Systementwurf für skalierbare Bildschirmwände

Nach der zuvor erläuterten grundlegenden Vorgehensweise zur verteilten Szenenberechnung mit dem Sort-First-Verfahren, wurde ein Systementwurf für skalierbare Bildschirmwände erstellt (Abb. 4.16). In Anlehnung an die Synchronisationsstrategien für die Datenverteilung in skalierbaren Visualisierungssystemen nach Roth (2005) (vgl. Abschnitt 3.3.1) wurde die Systemarchitektur nach dem Client-Server Modell konzipiert. Es besteht aus folgenden Komponenten:

- **Virtuelle Ausgabeschnittstelle:** Diese Komponente dient als Substitution der Grafik-API, die von der Anwendung eigentlich zur Ansteuerung der Grafikkette verwendet wird. Die Schnittstelle der Komponente entspricht daher nach außen genau dieser Grafik-API. Sie nimmt damit die Szenenbeschreibung, für die ein Bild errechnet werden soll, von der Anwendung entgegen. Die Szenenbeschreibung wird jedoch nicht zur Berechnung

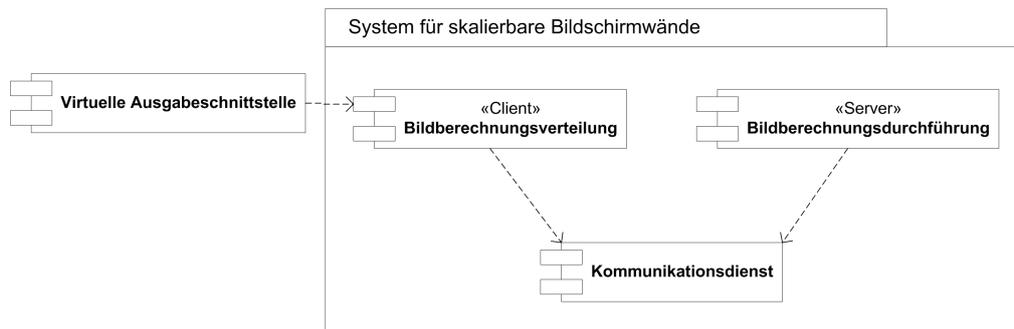


Abb. 4.16: Übersicht des Systems für hochauflösende Bildschirmwände

an die Grafikpipeline, sondern an die Komponente Bildberechnungsverteilung weitergeleitet.

- **Bildberechnungsverteilung:** Diese Komponente stellt den Client-Anteil der Systemarchitektur und bekommt über die virtuelle Ausgabeschnittstelle die Beschreibung der zu berechnenden Szene. Eine Szenenbeschreibung besteht aus einer Menge von Bildobjekten bzw. Grafikprimitiven und einer virtuellen Kamera. In der Bildberechnungsverteilung findet die Dekomposition der Szene statt (vgl. Abschnitt 3.3.3), d.h. hier werden die Bildobjekte in der Szenenbeschreibung sortiert und auf die zuständigen Knoten im verteilten Visualisierungssystem verteilt. Dazu benötigt diese Komponente explizite Informationen zur Konfiguration des gesamten Systems:
  1. Für die Sortierung der Bildobjekte müssen die Auflösungen und die Anordnung der Anzeigemedien bekannt sein, auf denen letztendlich die Ausgabe der Teilbilder der Szene erfolgen soll.
  2. Für die Verteilung muss zusätzlich bekannt sein, wie die Ausgaberechner im verteilten System zu erreichen sind (z.B. eine Identifikation über die IP-Adresse).

Aufgrund der bekannten Informationen kann die ursprüngliche Szenenbeschreibung in mehrere Teile zerlegt werden. Jede dieser Teilbeschreibungen enthält eine neue virtuelle Kamera, die auf Basis der Auflösungen und der Anordnung der verteilten Anzeigemedien ermittelt wird, sowie die Menge der Bildobjekte, die in diesem Bereich liegen.

- **Bildberechnungsdurchführung:** Diese Komponente stellt den Server-Anteil der Systemarchitektur und beinhaltet die ursprüngliche Grafikpipeline. Sie bekommt demnach von der Bildberechnungsverteilung eine Szenenbeschreibung, die unter Nutzung der zugrundeliegenden Grafik-API in der Grafikpipeline berechnet und auf dem Anzeigemedium ausgegeben wird.
- **Kommunikationsdienst:** Die Bildberechnungsverteilung (Client) und die Bildberechnungsdurchführung (Server) nutzen zum Austausch der Szenenbeschreibungen einen Kommunikationsdienst, der einen zuverlässigen Transport der auszutauschenden Daten garantiert.

Bei dieser Umsetzung des Verfahrens ist keine explizite Komposition der Szene notwendig (vgl. Abschnitt 3.3.3). Die Komposition erfolgt implizit durch die statische Anordnung der Ausgabemedien und der Tatsache, dass auf jedem Ausgabemedium des Visualisierungssystems ein grundsätzlich festgelegter Teil der Gesamtszene ausgegeben wird.

### Verteilung der Systemkomponenten und Integration in das Basissystem

Die Abbildung 4.17 zeigt die Verteilung der Komponenten in dem Visualisierungssystem nach dem Client-Server-Modell. Die Komponente *Bildberechnungsverteilung* läuft zusammen mit der Anwendung und dem (clientseitigen) Kommunikationsdienst auf dem Anwendungsclient. Die Komponente *Bildberechnungsdurchführung* läuft zusammen mit dem (serverseitigen) Kommunikationsdienst auf den Visualisierungsservern, die für die direkte Ansteuerung der Bildschirme zuständig sind.

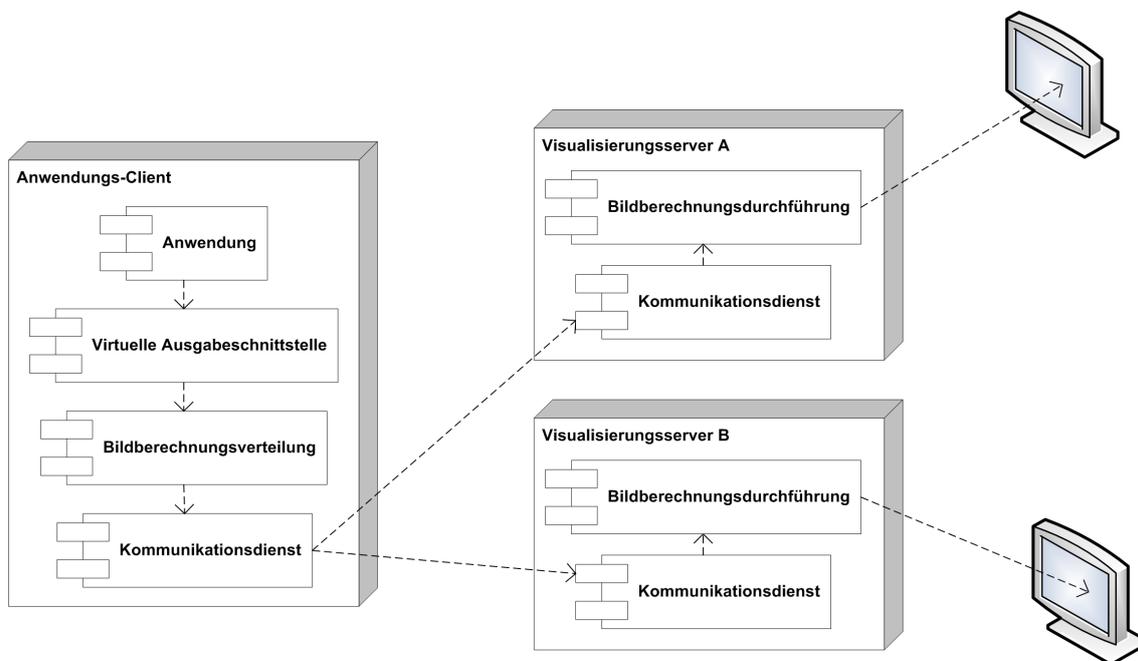


Abb. 4.17: Verteilung der Komponenten des Visualisierungssystems

Für jede skalierbare Bildschirmwand, die mit diesem System betrieben wird, gibt es demnach genau *einen* Anwendungsclient und *n* Visualisierungsserver (*n* in Abhängigkeit von der Anzahl der anzusteuernenden Monitore).

In Abbildung 4.18 ist dargestellt, wie sich das hier konzipierte Visualisierungssystem in das zuvor dargestellte Basissystem für spielbasierte Simulationen integrieren lässt. Die ursprünglich vorgesehene Ausgabeschnittstelle (z.B. OpenGL) wird dazu einfach durch die virtuelle Ausgabeschnittstelle ersetzt. Für alle abhängigen Komponenten des Basissystems bleibt diese Änderung aufgrund der identischen Schnittstelle verborgen.

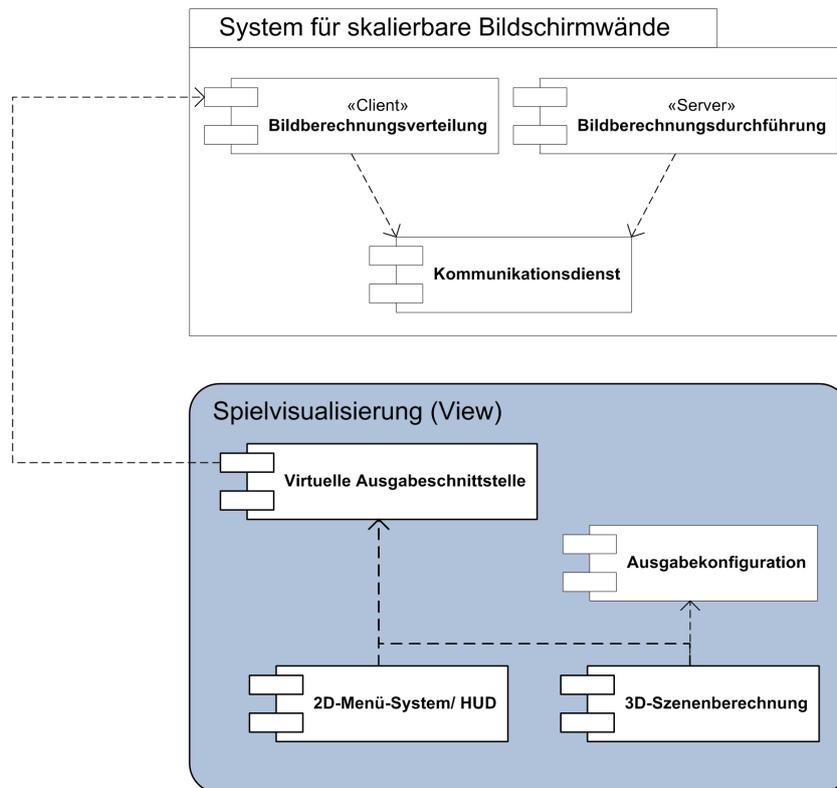


Abb. 4.18: Integration des Visualisierungssystems in das Basissystem

### 4.3.5 Komponente Spielmechanik (Control)

Die Komponente Spielmechanik des Basissystems setzt sich aus folgenden Teilkomponenten zusammen:

**Eingabeschnittstelle:** Die Eingabeschnittstelle dient zur Verwaltung der Eingabegeräte, wie z.B. Tastatur, Maus oder Joypad. Die Schnittstelle dient als Abstraktionsebene, um verschiedenartige Eingabegeräte zur Steuerung der spielbasierten Simulation einbinden zu können.

**Eingabeinterpretation:** Diese Teilkomponente ist für Vorverarbeitung der Benutzereingaben verantwortlich, die in der nachfolgenden Eingabeverarbeitung auf die Spielregeln angewendet bzw. gegen diese geprüft werden. Die Benutzereingaben aus der Eingabeschnittstelle werden hier interpretiert und in ein, für die Eingabeverarbeitung und das Model, „verständliches“ Format umgewandelt. So wird z.B. die Benutzereingabe „linke Maustaste“ in „Interaktion mit anvisiertem Objekt“ übertragen und an die Eingabeverarbeitung weitergereicht. Die Eingabeinterpretation beschreibt damit zugleich die Aktionsmöglichkeiten, die einem Benutzer im Rahmen einer konkreten spielbasierten Simulation zur Verfügung stehen.

**Eingabeverarbeitung/Spielregeln:** Mit dieser Teilkomponente wird der eigentliche Ablauf des Spiels definiert. Hier werden die Spielregeln festgelegt, die eine konkrete spielbasierte

Simulation charakterisieren. Die vorverarbeiteten Benutzereingaben aus der Eingabeinterpretation werden hier daraufhin geprüft, welche Änderungen am Model gemäß der Spielregeln vorgenommen werden müssen. Um an das zuvor genannte Beispiel anzuknüpfen, könnte die „Interaktion mit anvisiertem Objekt“ in den Spielregeln als „Objekt einsammeln“ definiert sein. Daraus lassen sich z.B. zwei Änderungen ableiten, die durch die Eingabeverarbeitung an das Model weitergegeben werden müssen: 1. „anvisiertes Objekt entfernen“ und 2. „anvisiertes Objekt dem Inventar des Spielers hinzufügen“.

Die Darstellung der Teilkomponenten beschreibt alle grundsätzlichen Aufgaben und Verantwortlichkeiten, die durch den Controller in der Architektur des Basissystems übernommen werden. Die für die synchrone Unterstützung mehrerer Spieler notwendige Verteilung und Synchronisation der Eingaben wurde dabei noch nicht explizit berücksichtigt. Diese Funktionalitäten des Systems wird in den folgenden Ausführungen dargelegt.

### **Eingabeverarbeitung als Verteilungs- und Synchronisationsschnittstelle**

Unabhängig davon welche Verteilungsform für die MVC-Komponenten des Basissystems gewählt wird, müssen alle Eingaben der Benutzer dem Server mitgeteilt werden (vgl. Abschnitt 4.3.2). Dazu wird die Teilkomponente *Eingabeverarbeitung* in dieser Konzeption in einen Client- und in einen Server-Teil aufgeteilt.

Auf der Seite der Clients bekommt die Eingabeverarbeitung die interpretierten Benutzereingaben aus der Eingabeinterpretation. Die Evaluierung der Spielregeln erfolgt jedoch nicht lokal. Stattdessen werden die Benutzereingaben über den Basisdienst *Kommunikation* an den Server weitergeleitet. In der serverseitigen Eingabeverarbeitung laufen die interpretierten Benutzereingaben aller Clients zusammen. Hier erfolgt eine zentrale Evaluierung der Spielregeln, die dann gegebenenfalls zu Änderungen des Models führen können.

Wie in Abschnitt 4.3.2 dargestellt wurde, ist die Eingabesynchronisation aus Gründen der Skalierbarkeit ein geeignetes Mittel, um allen Clients eine synchrone Interaktion und Visualisierung zu ermöglichen. Nachdem die interpretierten Benutzereingaben eines Clients evaluiert wurden, werden daher die Ergebnisse, d.h. die vorzunehmenden Änderungen am Model, an alle Clients des Systems propagiert. Jeder Client empfängt die Ergebnisse über die clientseitige Eingabeverarbeitung und wendet diese auf sein lokales Model an. Das Model bezieht seine Zustandsänderungen damit aus den Benutzereingaben aller beteiligten Benutzer. Die Gesamtheit aller verteilten Controller, d.h. vom Server und allen Clients, bildet demnach - aus der Sicht des Model - einen einheitlichen Controller. Die Abbildung 4.19 zeigt eine schematische Übersicht der gesamten Abhängigkeiten der verteilten Controller.

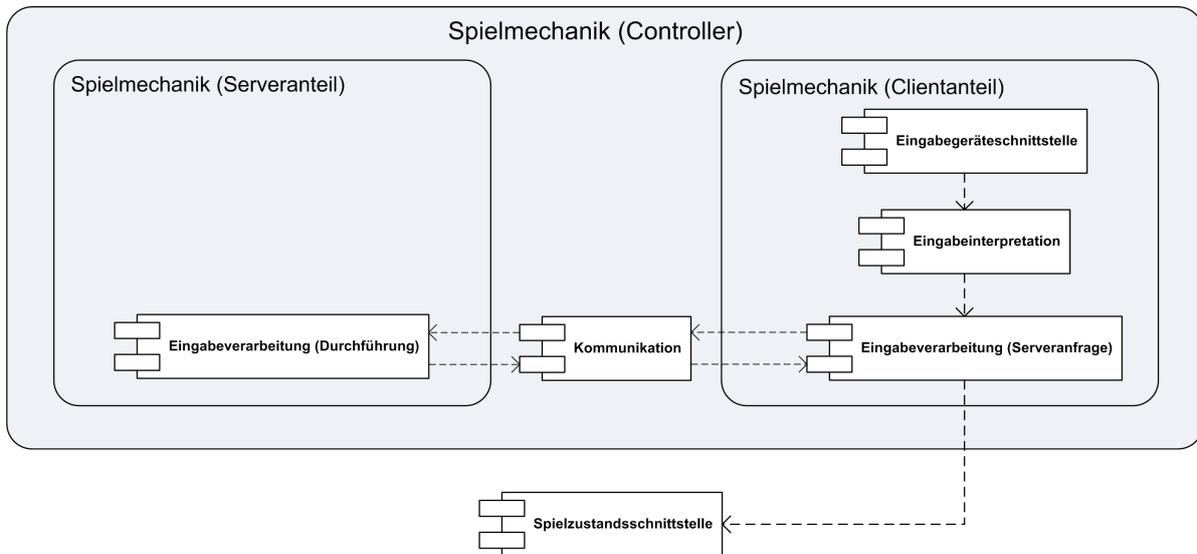


Abb. 4.19: Die Eingabeverarbeitung dient als Verteilungs- und Synchronisationsschnittstelle des Basissystems

## 4.4 Entwurf eines kollaborativen Spielkonzepts

Mit dem Basissystem als Grundlage kann das Spielkonzept integriert werden, das zum eigentlichen Teamtraining verwendet wird. Das Spielkonzept stellt eine Umsetzung bzw. Konkretisierung der funktionalen Anforderungen „Training überwachen“ und „Training durchführen“ dar, die im Abschnitt 3.4.2 ermittelt und für die verschiedenen Rollen durch spezifischere Anforderungen dargestellt wurden.

Im Folgenden wird zunächst in Abschnitt 4.4.1 das Spielkonzept beschrieben. Anschließend wird in Abschnitt 4.4.2 gezeigt, wie bei der Konzeption die Designimplikationen für die Entwicklung von kollaborativen Serious Games berücksichtigt wurden (vgl. Abschnitt 3.1.4).

### 4.4.1 Beschreibung des Spielkonzepts

Die Struktur der folgenden Beschreibung des Spielkonzepts wurde in Anlehnung an der von [Rollings und Morris \(2003\)](#) vorgeschlagenen Gestaltung eines *Game Design Documents* gewählt:

**Hintergrundgeschichte:** Aufgrund eines defekten Kabels ist ein mehrstöckiges Bürogebäude in Brand geraten. Die meisten Menschen, die an diesem Tag in dem Gebäude ihrer Arbeit nachgingen, konnten sich bereits über die bekannten Fluchtwege retten. Allerdings gibt es noch einige Nachzügler aus den oberen Geschossen. Sie versuchen über das Treppenhaus ins Erdgeschoss zu gelangen, um dort das Gebäude über den Notausgang zu verlassen. Im Erdgeschoss des Gebäudes angekommen, müssen sie feststellen, dass der Notausgang aufgrund eines Einsturzes unpassierbar ist und ihnen somit der rettende

Weg nach draußen versperrt bleibt. Der einzige Ausweg ist nun der Haupteingang des Gebäudes. Der Weg dorthin ist ein nahezu unmögliches Unterfangen, da bereits das ganze Erdgeschoss in Flammen steht. Mit der Aussichtslosigkeit vor Augen geraten die flüchtenden Menschen in Panik und rennen blindlings drauf los.

**Aufgabe der Spieler:** Die Spieler übernehmen die Rolle eines Einsatzteams der Feuerwehr, das versucht, die letzten flüchtenden Menschen aus dem Gebäude zu retten. Dazu müssen diese sicher und wohlbehalten zum Haupteingang des Gebäudes geleitet werden. Diese Aufgabe wird nicht ganz einfach, da die Menschen in Panik sind und sich dabei nicht mehr rational beeinflussen lassen. Sie rennen - ohne zu denken - immer geradeaus, und dabei mitunter direkt in den nächsten Brandherd hinein.

Das Einsatzteam der Feuerwehr besteht aus zwei verschiedenen Rollen:

1. Einsatzleitung (ca. 1-2 pro Team)
2. Einsatzkraft (ca. 3-5 pro Team)

Die Einsatzleitung überwacht und koordiniert den gesamten Einsatz aus der Vogelperspektive (Anwendungsfall „Aufgabendurchführung überwachen“). Zur Visualisierung und Interaktion wird hierbei auf die Powerwall zurückgegriffen. Die Einsatzkräfte sind die Männer „vor Ort“, die sich mit einem virtuellen Avatar aus der Ego-Perspektive unmittelbar durch das brennende Gebäude bewegen können (Anwendungsfall „Avatar bewegen“). Zur Visualisierung und Interaktion werden hier jeweils stationäre PCs oder Notebooks eingesetzt. Um den Fluchtweg der Menschen beeinflussen zu können, stehen den Einsatzkräften zwei Optionen zur Verfügung (Anwendungsfall „Interaktion mit Objekten“):

- Sie können einen Weg blockieren, um den in Panik geratenen Menschen eine andere Laufrichtung aufzuzwingen.
- Im gesamten Gebäude sind Feuerlöscher verteilt, mit denen die Spieler einen Brandherd löschen können, der einen Weg versperrt.

Die Einsatzleitung kann den Einsatzkräften dabei unterstützend zur Seite stehen, indem sie z.B. für jede Einsatzkraft Wegpunkte setzen kann, um die nächste zu blockierende Position zu markieren (Anwendungsfall „Aufgaben deligieren“). Des Weiteren kann zu jeder Einsatzkraft jederzeit dessen Status eingesehen oder auch sein Blickwinkel mitverfolgt werden (Anwendungsfall „Aktivitäten der Einsatzkräfte überwachen“).

Das Spiel wird von einem Ausbilder überwacht, der den Schwierigkeitsgrad steuert und für dynamische Situationsänderungen sorgt. Dazu kann er Brandherde, Feuerlöscher und flüchtende Menschen hinzufügen bzw. entfernen (Anwendungsfall „Training manipulieren“).

**Ziel des Spiels:** Das Ziel des Spiels ist, alle flüchtenden Menschen zum Haupteingang zu bringen, bevor das Gebäude einstürzt. Der Einsturz des Gebäudes wird durch einen

rückwärts laufenden Timer prognostiziert. Für jeden geretteten Menschen gibt es Punkte. Für jeden Menschen, der es nicht aus dem Gebäude geschafft, werden Punkte abgezogen. Ein zusätzlicher Punktebonus wird am Ende des Einsatzes auf Basis der verbleibenden Restzeit vergeben. Das Spiel verfügt über eine Punkterangliste, die zeigt welches Team sich in dem jeweiligen Einsatz am Besten geschlagen hat.

**Spielablauf:** Jeder Einsatz gliedert sich in zwei Phasen:

1. Planungsphase
2. Aktionsphase

In der Planungsphase steht dem gesamten Einsatzteam eine kurze Zeit zur Verfügung, um die initiale Vorgehensweise des Einsatzes zu planen (Anwendungsfall „Trainingseinsatz planen“). In diesem Zeitraum werden die jeweiligen Startpunkte jeder Einsatzkraft aus einer Reihe vorgegebener Möglichkeiten ausgewählt. Zudem können hier erste Aktionen und Aufgaben koordiniert und abgesprochen werden. Die verbleibende Zeit für die Planungsphase wird durch einen rückwärts laufenden Timer gekennzeichnet. Mit Ablauf des Timers startet der Einsatz automatisch, die Einsatzkräfte werden an die ausgewählten Startpunkte gesetzt und es erfolgt die Aktionsphase. Wenn zu diesem Zeitpunkt noch keine Startpunkte gewählt wurden, werden die Einsatzkräfte zufällig über die vorgegebenen Möglichkeiten verteilt. Es besteht auch die Möglichkeit die Planungsphase noch vor Ablauf des Timers zu beenden und den Einsatz (und damit die Aktionsphase) manuell zu starten. Die verbleibende Restzeit aus der Planungsphase wird in der Aktionsphase als Zeitbonus gutgeschrieben.

In der Aktionsphase können alle Spieler ihren rollenspezifischen Aufgaben nachgehen. Auch die Aktionsphase ist zeitlich begrenzt. Wenn die flüchtenden Menschen bis zum Ablauf der Zeit nicht gerettet wurden, gilt das Spiel als verloren. Während der Aktionsphase kommt es zu drastischen Abzügen auf die verbleibende Zeit, wenn ein flüchtender Mensch mit einem Brandherd in Berührung kommt. Das Einsatzteam sollte diese Situation durch kontrolliertes und koordiniertes Vorgehen vermeiden, um die Chancen auf eine erfolgreiche Beendigung des Trainingsszenarios zu bewahren.

Von einer genauen Spezifikation der im Spielkonzept erwähnten Timer bzw. Zeiten für bestimmte Spielabschnitte wird an dieser Stelle abgesehen. Dafür „brauchbare“ Werte sind von einer Reihe von Faktoren abhängig (z.B. Größe und Komplexität des Trainingsszenarios, Bewegungsgeschwindigkeiten der Spielfiguren) und können erst anhand einiger Testläufe ermittelt werden.

#### 4.4.2 Berücksichtigung der Designimplikationen

Die Gestaltung des Spielkonzepts orientiert sich an den in Unterkapitel 3.1 ermittelten Designimplikationen für die Entwicklung von Serious Games für Teamtrainings. Die folgende Aufstellung zeigt auf, welche Designimplikationen in welcher Art und Weise berücksichtigt wurden:

- **Interaktivität:** Alle Spiele besitzen Interaktionsmöglichkeiten in dem Trainingsszenario (*Aufgaben der Spieler* in Abschnitt 4.4.1).
- **Modellplausibilität:** Die Modellplausibilität wird durch das Basissystem gewährleistet (vgl. Abschnitt 4.3.2).
- **Unsicherheit:** Die flüchtenden Menschen agieren autonom in dem Trainingsszenario, womit keine konkrete Vorhersage des weiteren Spielverlaufs bzw. des Spielausgangs für die Spieler möglich ist.
- **Motivation:** Ein einfaches Punktesystem mit einer Rangliste dient zur Motivation für wiederholtes Spielen bzw. Trainieren.
- **Spielfluss:** Alle Spieler wissen welches Hauptziel zu erreichen ist. Die dafür zu erledigen Einzelaufgaben werden vom Einsatzleiter bestimmt und verteilt. Der Schwierigkeitsgrad kann durch den Ausbilder dynamisch angepasst werden.
- **Reflektion:** Die Artikulation von gemeinsamen Vorgehensweisen und Problemen ist sowohl in der Planungs- als auch in Aktionsphase ein wesentlicher Teil des Spielkonzepts.
- **Perspektiven:** Verschiedene Perspektiven werden sowohl durch die Einnahme verschiedener Rollen und deren Verantwortlichkeiten als auch durch unterschiedliche Sichtweisen auf das Trainingsszenario unterstützt.
- **Angemessenheit:** Diese Designimplikation wurde nicht explizit berücksichtigt.
- **Erkundung:** Alle beteiligten Spieler können die Umgebung des Trainingsszenario aus verschiedenen Blickwinkeln erkunden.
- **Inkrementalität:** Diese Designimplikation wurde nicht explizit berücksichtigt. Hier ist es aber denkbar, aufeinander aufbauende Trainingsszenarien zu entwickeln, deren Schwierigkeitsgrad stufenweise zunimmt.
- **Mehrdeutige Aufgabensituationen:** Es gibt zwar genau eine Siegbedingung, aber sehr viele Wege diese zu erreichen, da die genaue Laufrichtung der flüchtenden Menschen vorhersehbar ist.
- **Unterstützung konvergenter Gruppenprozesse:** Der gesamte Spielverlauf stellt einen konvergenten Gruppenprozess dar. In der Planungs- und Aktionsphase werden die Gesamtsituation überblickt, Probleme identifiziert, Strategien entwickelt und Aufgaben durchgeführt, um sich letztendlich auf einen gemeinsamen Lösungsweg zu verständigen.

Auf Basis der Analyse in Unterkapitel 3.1 erfüllt das hier dargestellt Spielkonzept damit aus rein formaler Sicht den Anspruch eines Serious Game.

## 4.5 Realisierung

Im Rahmen dieser Arbeit wurde ein Prototyp der spielbasierten, kollaborativen Trainingsumgebung realisiert. In diesem Prototyp wurde nicht das komplette ausgearbeitete Konzept umgesetzt, sondern lediglich die wesentlichen Funktionen, die die Tragfähigkeit des Systems veranschaulichen.

### 4.5.1 Allgemeines

In Hinblick auf die Möglichkeit zu einer Evaluation und Bewertung des Systems standen bei der Realisierung die folgenden Zielsetzungen im Vordergrund:

- Realisierung der globalen und lokalen Sichten für Einsatzleiter und Einsatzkräfte
- Bereitstellung eines rudimentären Basissystems für spielbasierte Simulation
- Realisierung der wesentlichen Aspekte des Spielkonzepts für das Teamtraining
- Erstellung eines Trainingsszenarios für die spielbasierte Simulation

Im Folgenden werden zunächst alle wesentlichen Technologien und Entwicklungswerkzeuge dargestellt, die bei der Realisierung eingesetzt wurden. Im Anschluss daran erfolgt eine konkrete Erläuterung, wie die Systemkomponenten aus der Übersicht in Abschnitt 4.3.1 in diesem Prototyp realisiert wurden. Ergänzt wird diese Darstellung durch einige Screenshots des Systems, die die wichtigsten Funktionalitäten veranschaulichen.

### Verwendung der Torque Game Engine (TGE)

Bei der Realisierung wurde deutlich, dass eine eigenständige Umsetzung des in der Konzeption dargestellten Basissystems im zeitlich begrenzten Rahmen dieser Arbeit nicht möglich ist. Da dieses System jedoch die Grundvoraussetzung für die Integration einer spielbasierten Simulation ist, und damit für eine Evaluation und Bewertung dieser Arbeit unabdinglich ist, wurde nach existierenden Realisierungen gesucht.

Dafür wurden verschiedene Technologien aus der Spieleentwicklung auf deren Übertragbarkeit auf das Systemkonzept geprüft. Zunächst wurden einige Grafik-Engines (z.B. [Ogre3D \(2007\)](#) und [Irrlicht \(2008\)](#)) betrachtet, die für die Berechnung und Darstellung einer dreidimensionalen Szene geeignet sind. Der Funktionalitätsumfang dieser Systeme entspricht der Komponente *Spielvisualisierung* des Basissystems. Die gesamte Szenenverwaltung, also sämtliche Spielweltobjekte sowie deren physikalischen und logischen Zusammenhänge, hätte damit auch „manuell“ realisiert werden müssen.

Im weiteren Verlauf der Technologieprüfung wurde daher nach umfassenderen Systemen gesucht, die nicht nur die Funktionalitäten der View-Komponente, sondern auch Funktionalitäten der Model-Komponente des Basissystems bereitstellen. In diesem Zusammenhang wurden

einige *Game-Engines* näher betrachtet. Bei solchen Systemen handelt es sich um Software-Pakete bzw. auch Softwarebibliotheken, mit denen häufig benutzte Werkzeuge für die Entwicklung von Computerspielen zur Verfügung gestellt werden. Einige Beispiele für solche Systeme sind die *C4 Engine*<sup>3</sup>, *Truevision3D*<sup>4</sup> und die *Torque Game Engine*<sup>5</sup>.

Nach eingehender Prüfung der verschiedenen Systeme auf die Eigenschaften *Funktionsumfang*, *Portabilität*, *Erweiterbarkeit* und *Performanz* fiel die Wahl auf die *Torque Game Engine* (TGE). TGE enthält neben einer Grafik-Engine z.B. auch eine Physik-Engine, Kollisionskontrolle, Soundunterstützung, Künstliche Intelligenz, Netzwerkprotokolle und eine Szenenverwaltung. Durch die Verfügbarkeit des kompletten Quelltextes von TGE besteht die Möglichkeit, das System beliebig anzupassen und für unterschiedliche Plattformen zu kompilieren.

Die Kernbibliotheken der *Torque Game Engine* sind vollständig in C++ implementiert. Die Programmierung der eigentlichen Spielmechanik (Gameplay) erfolgt bei TGE über *Torque-Script*. Dabei handelt es sich um eine C++-ähnliche Programmiersprache, die speziell zur Anwendung mit den Funktionsbibliotheken von TGE erstellt wurde.

Zur Kompilierung und Ausführung solcher *Torque-Scripts* wird eine interne *Scripting-Engine* verwendet. Diese fungiert als virtuelle Maschine, die eine Schnittstelle zu den Kernbibliotheken von TGE bildet. Ähnlich wie bei der Verwendung der Programmiersprache Java werden *Torque-Scripts* vor Ausführung in Byte-Code kompiliert, der dann durch die *Scripting-Engine* zur Laufzeit interpretiert wird und dabei die entsprechenden Kernfunktionalitäten der *Torque Game Engine* nutzt.

## Entwicklungsumgebung

Zur Verwaltung, Modifikation und Erweiterung der C++-basierten Kernbibliotheken von TGE wurde *Microsoft Visual Studio 2005* eingesetzt.

Für die *Torque-Scripts*, die zur Realisierung des Spielkonzepts erstellt wurden, ist *Torsion IDE*<sup>6</sup> zum Einsatz gekommen. Dabei handelt es sich um einen Editor, der speziell für die Syntax und die Semantik von *Torque-Script* konzipiert wurde. Insbesondere die umfassenden Funktionen zur allgemeinen Projektverwaltung und zum Debugging der erstellten Skripte, haben sich günstig auf die benötigte Entwicklungszeit ausgewirkt.

Für die Erstellung der 3D-Modelle des Trainingsszenarios, wurden die Modellierwerkzeuge *Torque Constructor* (Erstellung von Gebäuden) und *Autodesk 3ds Max* (Erstellung von kleineren Interaktionsobjekten) gewählt. Der Grund für die Verwendung von zwei Modellierwerkzeugen liegt in den benötigten 3D-Formaten, die bei der TGE verwendet werden. Die Details dazu werden in den Abschnitten 4.5.4 und 4.5.5 erläutert.

---

<sup>3</sup><http://www.terathon.com/c4engine>

<sup>4</sup><http://www.truevision3d.com>

<sup>5</sup><http://www.garagegames.com>

<sup>6</sup><http://www.garagegames.com/products/106>

## 4.5.2 Skalierbare Bildschirmwand (Powerwall)

Bei der Umsetzung des Systemkonzepts für skalierbare Bildschirmwände (Abschnitt 4.3.4) wurde auf bestehende Realisierungen gesetzt. Aufgrund der Ähnlichkeiten zum Systemkonzept dieser Arbeit wurde für diese Zwecke das Chromium-System (Humphreys u. a., 2002) eingesetzt und entsprechend angepasst.

### Grundlegende Funktionsweise und Analogien zum Systementwurf

Chromium ist ein erweiterbares System zum interaktiven, verteilten Rendering von Applikationen, die auf OpenGL basieren. Wie in der Konzeption vorgesehen, unterstützt auch Chromium das Sort-first/Tiled Rendering. Darüberhinaus lässt sich mit dem System auch das Sort-Last-Verfahren und ein hybrides Sort-First/Sort-Last -verfahren realisieren. Für die Umsetzung in dieser Arbeit wurde lediglich auf das Sort-First-Verfahren gesetzt. Chromium besteht im Wesentlichen aus drei Dienstprogrammen: *crappfaker*, *crserver* und *crmothership*.

Die Funktionalität des Dienstprogramms *crappfaker* entspricht den Komponenten *Virtuelle Ausgabeschnittstelle* und *Bildberechnungsverteilung* in der Konzeption dieser Arbeit (vgl. Abschnitt 4.3.4). Es läuft daher mit der Anwendung auf demselben Knoten im verteilten Visualisierungssystem. Die Anwendung wird dem Dienstprogramm *crappfaker* als Parameter übergeben, wenn es gestartet wird. Auf diese Weise wird die Anwendung „gekapselt“: alle von ihr ausgehenden OpenGL-Kommandos werden von dem Dienstprogramm *crappfaker* abgefangen und auf die entsprechenden Knoten verteilt.

Das Gegenstück zum *crappfaker* ist das Dienstprogramm *crserver*. Es entspricht damit der Komponente *Bildberechnungsdurchführung* in der Konzeption dieser Arbeit. Eine Instanz dieses Programms läuft auf jedem Rechner im verteilten System, der für die Steuerung eines Bildschirms zuständig ist. Das Dienstprogramm erhält die aufgeteilten OpenGL-Kommandos vom *crappfaker*, berechnet das Bild und gibt es auf dem von ihm verwalteten Bildschirm aus.

Die Kommunikation zwischen beiden Dienstprogrammen wird durch ein weiteres Dienstprogramm, das sogenannte *crmothership*, unterstützt. Dieses läuft ebenfalls auf einem Knoten mit der Anwendung. Dieses Dienstprogramm hält alle Informationen zur Konfiguration des Visualisierungssystem, wie z.B. die Anordnung der Bildschirme und entsprechende Netzwerkadressen.

### Aufbau der skalierbaren Bildschirmwand im Ambient Labor

Die Powerwall im Ambient Labor der HAW Hamburg besteht aus einer Matrix von insgesamt neun Bildschirmen (Anordnung: 3x3), die von einem Cluster aus insgesamt sechs Rechnern betrieben wird. Auf einem dieser Rechner läuft die Anwendung zusammen mit den Dienstprogrammen *crappfaker* und *crmothership*. Die übrigen Rechner werden zur Ansteuerung der



Abb. 4.20: Die skalierbare Bildschirmwand im Ambient Labor

Bildschirme verwendet und betreiben jeweils eine Instanz des Dienstprogramms *crserver*. Die Abbildung 4.20 zeigt die Powerwall im laufenden Betrieb<sup>7</sup>.

Eine ausführliche Auflistung der eingesetzten Hardware-Komponenten findet sich im Anhang A.1. Vertiefende Informationen zum Betrieb und zur Konfiguration von Chromium kann der offiziellen Dokumentation entnommen werden ([Chromium, 2007](#)).

### Interaktionsmöglichkeiten

Da in dieser Arbeit die technische Umsetzung einer skalierbaren Bildschirmwand und deren Integration in einer spielbasierten Trainingsumgebung im Vordergrund stand, konnten aus zeitlichen Aspekten keine speziellen Interaktionskonzepte für die *Close-range* oder *Distant Interaction* umgesetzt bzw. integriert werden (vgl. Abschnitt 3.2.3).

Dieser Aspekt wird für eine mögliche Weiterentwicklung des Systems offengelassen. Für die erste Evaluation in Kapitel 5 erfolgt die Interaktion daher mit klassischen Eingabegeräten (Maus und Tastatur) und auf Basis der *physischen Navigation*.

<sup>7</sup>Als exemplarische Anwendung dient hier Google Earth (<http://earth.google.de>)

### 4.5.3 Basisdienste

Die Basisdienste umfassen wie in Abschnitt beschrieben 4.3.1 die Kommunikation und den Szenariodatenzugriff.

Um die synchrone Visualisierung auf allen Anzeigemedien im CCTE zu garantieren, wurde ein zuverlässiger synchroner Kommunikationsdienst benötigt. Dafür wurde das Transmission Control Protocol (TCP) ausgewählt. Durch die Verwendung der TGE konnte *TorqueNet* als Programmierschnittstelle zu diesem Kommunikationsprotokoll eingesetzt werden. TorqueNet bietet einen abstrahierten Zugriff auf die beiden gängigsten Protokolle der Transportschicht (TCP und UDP). Zudem ermöglicht es eine vereinfachte Umsetzung des Client-/Server-Konzepts und stellt den sogenannten *Ghostmanager* bereit, der im weiteren Verlauf der Realisierung bei der Synchronisation der Clients notwendig ist.

Der Szenariodatenzugriff erfolgt bei dem vorliegenden Prototypen über das lokale Dateisystem des Rechners, der als Spielserver fungiert. Alle Szenariodaten, die auch von den Clients benötigt werden, werden zu Beginn bei der Spielkonfiguration an die Clients übertragen. Nach dieser Datensynchronisation erfolgt der Szenariodatenzugriff clientseitig ebenfalls über das lokale Dateisystem. Bei steigender Anzahl und/oder Komplexität der Trainingsszenarien empfiehlt sich für weiterführende Arbeiten eine zentrale Verwaltung der notwendigen Daten, z.B. auf einem dedizierten Datei- oder Datenbankserver.

### 4.5.4 Szenariodaten

Die für das Trainingsszenario notwendigen Daten bestehen im Wesentlichen aus drei Kategorien:

**3D-Modelle:** Unter diese Kategorien fallen alle Daten, die die geometrischen Beschaffenheiten aller in der virtuellen Trainingsumgebung repräsentieren. Bei der Verwendung der TGE werden zwei unterschiedliche Dateiformate unterstützt und im Prototyp eingesetzt:

- **Dynamix Treespace Shapes (\*.DTS):** Dieses Format wird zur Repräsentation von Objekten verwendet, die keine Strukturen in der virtuellen Umgebung darstellen. In der Terminologie von TGE werden solche Objekte als *Shapes* bezeichnet. Dazu zählen u.a. Spielerfiguren und alle Interaktionsobjekte, wie z.B. die im Spielkonzept erwähnten Feuerlöscher. Zur Gestaltung dieser Objekte kann eine beliebige 3D-Modellierungssoftware verwendet werden, die das DTS-Format unterstützt. In dieser Realisierung wurde dafür *Autodesk 3ds Max* in Kombination mit dem *Max2DTS Exporter*<sup>8</sup> eingesetzt, der die Exportierung vom 3DS-Formaten in das DTS-Format ermöglicht.

<sup>8</sup><http://www.garagegames.com/docs/tge/general/ch08s02.php>

- Dynamix Interior Formats (\*.DIF): Im Gegensatz zu DTS-basierten Modellen wird dieses Format dazu genutzt, um Strukturen, wie z.B. Gebäude, in der virtuellen Umgebung darzustellen. Im Rahmen der TGE werden solche Objekte als *Interiors* bezeichnet. Gebäudemodelle lassen sich zwar auch im DTS-Format erstellen, bringen jedoch bei entsprechender Größe, Vielfalt und Komplexität der zugrundeliegenden Polygone massive Performanzprobleme mit sich. Durch den Einsatz von Modellen im DIF-Format werden diese Probleme relativiert, in dem diese in der TGE „vereinfacht“ behandelt werden. Grundlage dafür bildet die Annahme, dass große Strukturen (z.B. Gebäude oder Brücken) in der virtuellen Umgebung statisch sind, d.h. nicht die Position wechseln oder rotieren. Diese Annahme kann z.B. dazu verwendet werden die Kollisionsberechnung mit dem sogenannten *Binary Space Partitioning* (BSP) effizienter zu gestalten. In dieser Realisierung wurde zur Gestaltung eines Übungsgebäudes für ein exemplarisches Trainingszenario der *Torque Constructor* verwendet (Abb. 4.21).

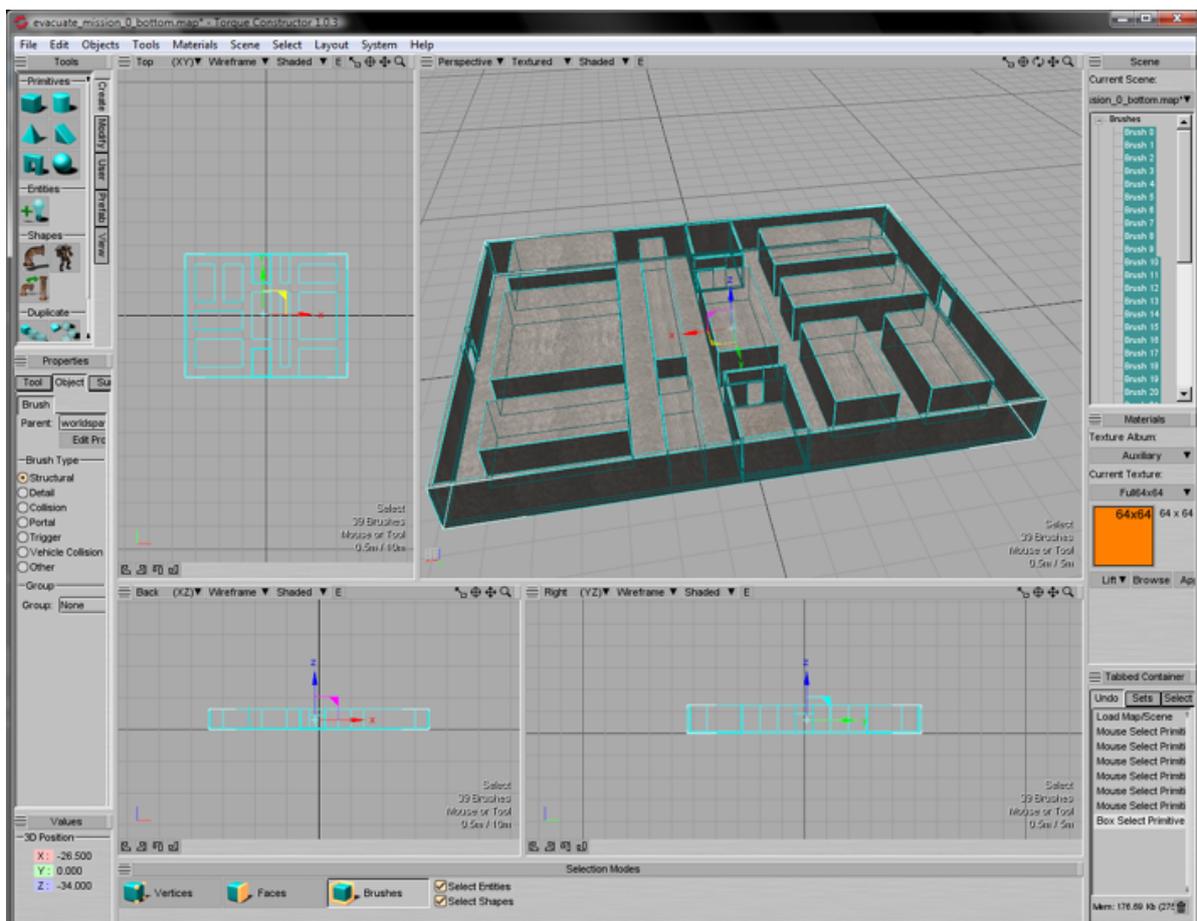


Abb. 4.21: Gestaltung eines Trainingsgebäudes für das Szenario im Torque Constructor

Für weitere Informationen zu diesen Dateiformaten wird an dieser Stelle auf [Maurina \(2006\)](#) verwiesen. Im Resümee dieser Arbeit wird diskutiert werden, welche alternativen

Verfahren und Formate hier denkbar sind, um z.B. bestehende CAD-basierte Gebäudemodelle einbinden zu können.

**2D-Grafiken:** Hierzu zählen in erster Linie die Texturen der oben dargestellten 3D-Modelle. Unter einer Textur versteht man in diesem Zusammenhang eine Grafik, die auf ein dreidimensionales Objekt projiziert wird. Die hierfür verwendeten Grafikformate sind typischerweise *Portable Network Graphics* (PNG), *Windows Bitmap* (BMP) und *Graphics Interchange Format* (GIF).

**Szenariobeschreibung:** In der Szenariobeschreibung wird der gesamte initiale Aufbau des Trainingsszenarios festgehalten. Sie enthält die Positionen und Rotationen aller darin enthaltenen Objekte sowie die initialen Werte von objektspezifischen Parametern (z.B. die Masse eines Objekts). Bei der Realisierung wurde für diese Szenariobeschreibung ein spezielles Torque-Script, das sogenannte *Mission-File*, verwendet. Es besteht aus einer Reihe von Objektinstanzierungen, die hierarchisch ineinander verschachtelt sind. Dadurch kann bei der Initialisierung der Spielsimulation ein logischer und räumlicher Zusammenhang zwischen den Objekten hergestellt werden, was sich in dem Aufbau des Szenengraphen wiederfindet. Für weiterführende Arbeiten, empfiehlt es sich, für das Szenario eine standardisierte Spezifikation zur Beschreibung der hierarchisch strukturierten Objekte zu verwenden. Hier empfiehlt sich der Einsatz von der *Extensible Markup Language* (XML), die speziell für solche Darstellungen konzipiert wurde.

Die für ein Trainingsszenario benötigten 3D-Modelle und 2D-Grafiken werden in der Initialisierungsphase eines Spiels auf die beteiligten Clients übertragen und sind anschliessend lokal auf jedem Knoten verfügbar. Die Szenariobeschreibungen werden nur auf dem Rechner benötigt, auf dem der Spielserver und die globale Spielsimulation gestartet wird. Für die Synchronisation der clientseitigen lokalen Spielsimulationen mit der globalen Spielsimulation müssen diese Initialisierungsskripte nicht übertragen werden.

### 4.5.5 Szenarioverwaltung

In der Konzeption ist die Szenarioverwaltung als eine Sammlung von Werkzeugen für die Rolle des Ausbilders vorgesehen, damit dieser neue Trainingsszenarios und die darin enthaltenen Objekte für die spielbasierte Simulation erstellen kann. In dieser Realisierung wurde darauf verzichtet, für diese Zwecke ein speziell erstelltes System bereitzustellen. Stattdessen wird für die Erstellung der Szenarios der sogenannte *Torque World Editor* verwendet, der Bestandteil der TGE ist. Mit diesem Werkzeug lassen sich die Objekte der virtuellen Umgebung platzieren und verschieben sowie deren objektspezifischen Eigenschaften konfigurieren. Eine auf diese Weise erstellte Szene kann anschließend als Szenariobeschreibung bzw. Mission-File exportiert werden. Die Abbildung 4.22 zeigt die Erstellung eines exemplarischen Trainingsszenarios mit dem Torque World Editor.

Eine Szenariobeschreibung enthält lediglich Informationen zu den Objekten und objektspezifischen Parametern in der virtuellen Umgebung. Wie in Abschnitt 3.4.1 erörtert wurde, gehören zu der vollständigen Definition eines Trainingsszenarios zusätzlich noch die Formulierung



Abb. 4.22: Erstellung eines Trainingszenarios über den Missionseditor (hier verwendet: Torque World Editor)

von Aufgaben, die zu einer erfolgreichen Beendigung des Trainingszenarios führen, und die Festlegung von Fähigkeiten und Werkzeugen, die für die Erfüllung der Aufgaben verwendet werden können. Für den Prototypen des Systems werden diese Informationen einfach in einer statischen Konfigurationsdatei abgelegt, die bei der Initialisierung des Spielservers ausgelesen wird.

#### 4.5.6 Spielkonfiguration

Die Spielkonfiguration dient dazu, die eigentliche spielbasierte Simulation zu konfigurieren und zu starten. In dem Prototyp wurde für alle beteiligten Rollen eine einheitliche Benutzeroberfläche gestaltet, die auf Basis der 2D-Bibliotheken von Torque realisiert wurde (Abb. 4.23). Die gesamte Logik der Spielkonfiguration wurde auf Basis von Torque-Script umgesetzt. Im Folgenden wird der Ablauf der in der Spielkonfiguration notwendigen Schritte und einige Realisierungsdetails dargelegt:

1. Trainingszenario auswählen: Die Auswahl des Trainings erfolgt durch den Ausbilder. Über eine grafische Benutzeroberfläche kann er aus einer Liste verfügbarer Szenariobeschreibungen in einem vorgegebenen lokalen Verzeichnis, das für das Training erwünschte Szenario auswählen (Abb. 4.24, links).



Abb. 4.23: Das Hauptmenü des Prototypen „Evacuate“

2. Spielserver starten: Nach der Auswahl des Trainingsszenarios kann der Ausbilder den Spielserver starten. Die Spielsimulation wird mit den Informationen der Szenariobeschreibung initialisiert. Zusätzlich dazu wird die Konfigurationsdatei ausgelesen, in der Aufgaben und Rahmenbedingungen definiert sind. In dieser Phase wird die Spielsimulation nur konfiguriert, jedoch noch nicht gestartet. Nach der Initialisierung wird über TorqueNet ein Kommunikationsserver gestartet (TCP-Serversocket), der auf eingehende Verbindungsanfragen wartet.
3. Spielserver auswählen: Das Einsatzteam kann anschließend über das Startmenü den Spielserver suchen und auswählen (Join Game). Um die genaue Kenntnis der Netzwerkadresse zu vermeiden, kann das Netzwerk nach einem bestehenden Spielserver durchsucht werden. Nach der Suche sind alle verfügbaren Spielserver in einer Liste mit auswählbaren Elementen verfügbar (Abb. 4.24, rechts).
4. Spielserver beitreten: Eine Bestätigung der Auswahl führt zu einer Beitrittsbenachrichtigung an den Spielserver über den Kommunikationsdienst. Die Clientanfrage wird von dem Kommunikationsdienst des Spielservers entgegengenommen und eine neue Client-Verbindung wird aufgebaut. Die Synchronisation der Spielsimulationen erfolgt bei Torque über den sogenannten Ghost Manager. Alle vorhandenen Objekte der virtuellen Umgebung werden durch den Ghost Manager an den neu beigetretenen Client verteilt. An dieser Stelle wird von der ursprünglichen Konzeption abgewichen. Der Ghost Manager ist ein Teil der Model-Komponente des Servers. Die zu verteilenden Objekte werden unmittelbar in das Model des Clients „kopiert“. Hier erfolgt also keine Kommunikation zwischen den Controller-Komponenten, sondern zwischen den Model-Komponenten. Damit wird hier keine Eingabesynchronisation verwendet, wie es ursprünglich vorgesehen war.

5. Training starten: Wenn alle Mitglieder des Einsatzteams dem Spielserver beigetreten sind, sendet der Spielserver an alle Clients das Kommando zum Trainingsstart. Dies führt dazu, dass die Spielkonfiguration beendet und die vorkonfigurierte spielbasierte Simulation gestartet wird.



Abb. 4.24: Menüs zum Starten eines neuen Trainingsszenarios durch den Ausbilder (links) und dem Beitritt des Einsatzteams (rechts)

Nach der Beendigung der spielbasierten Simulation (also nach Gewinn, Niederlage oder Abbruch des gewählten Trainingsszenarios) kehrt das System zunächst zur Spielkonfiguration zurück. Von hier aus kann dann ein neues Training gestartet oder die gesamte Anwendung beendet werden.

#### 4.5.7 Spielbasierte Simulation

Wie in Abschnitt 4.5.1 erörtert wurde, konnten mit dem Einsatz der Torque Game Engine große Teile des Basissystems, insbesondere die gesamte Szenenverwaltung und die Szenenberechnung, abgedeckt werden. Bei der Realisierung der spielbasierten Simulation konnte daher der Schwerpunkt auf die Umsetzung des kollaborativen Spielkonzepts gelegt werden. In Bezug auf die Konzeption des Basissystems deckt das insbesondere die Umsetzung der Komponente Spielmechanik und der gestalterischen Aspekte der Spielvisualisierung ab. Die Integration des Spielkonzepts wurde zum größten Teil auf Basis von Torque-Script realisiert. Die Skripte werden beim Start des Systems von der Scripting-Engine kompiliert und in den Speicher geladen. Erst beim Start des Trainings erfolgt die eigentliche Ausführung der Skripte.

Das Spielkonzept beschreibt eine Konkretisierung der funktionalen Anforderungen „Training überwachen“ und „Training durchführen“, die im Abschnitt 3.4.2 ermittelt und für die verschiedenen Rollen durch spezifischere Anforderungen dargestellt wurden. In dem vorliegenden Prototyp wurden diese wie folgt berücksichtigt:

- Trainingseinsatz planen: Die Abbildung 4.25 zeigt die Benutzerschnittstellen von Einsatzleiter (links) und Einsatzkräften (rechts) in dieser Phase. Die Sicht für beide Rollen

erfolgt zu diesem Zeitpunkt aus der Vogelperspektive. Die Benutzerschnittstelle der Einsatzkräfte enthält zusätzlich nummerierte Felder, die die möglichen Startpunkte für den Einsatz markieren. Jede Einsatzkraft kann im Rahmen der Planung einen dieser Startpunkte auswählen, in dem dann in der nachfolgenden Aktionsphase der Avatar platziert wird.

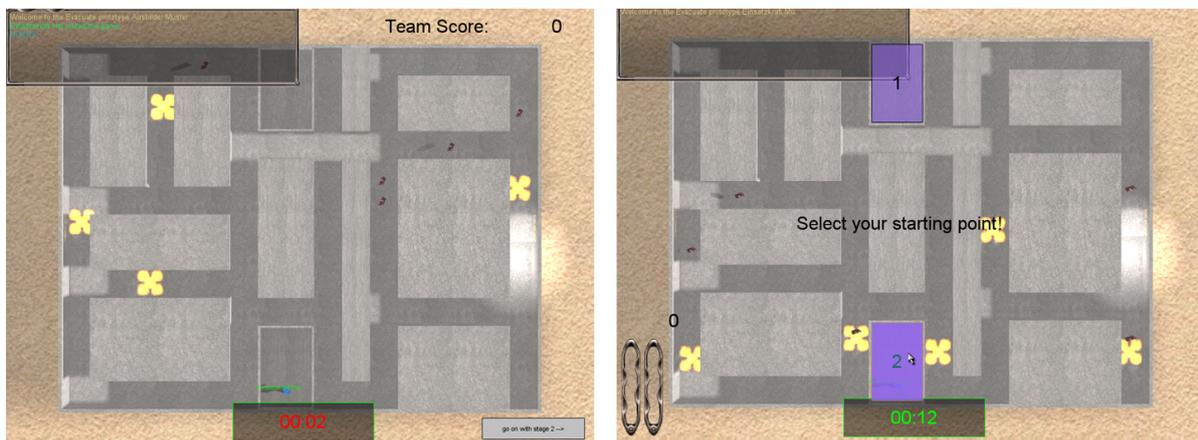


Abb. 4.25: Benutzerschnittstellen in der Planungsphase: links für den Einsatzleiter und rechts für die Einsatzkräfte

- Aufgabendurchführung überwachen: Die Aufgabendurchführung erfolgt durch den Einsatzleiter. Während der gesamten Dauer des Einsatzes bleibt für ihn daher die alles überblickende Sicht aus der Vogelperspektive bestehen (Abb. 4.26, links). Zusätzlich besteht die Möglichkeit die lokale Sicht (aus der Egoperspektive) einer bzw. mehrerer Einsatzkräfte in einem überlagerten Fenster zu öffnen.
- Aufgaben delegieren: Im momentanen Stadium des Prototyps ist keine softwaregestützte Delegation von Aufgaben möglich. Diese erfolgt stattdessen ausschließlich über direkte Face-to-Face-Kommunikation. Die Aspekte der Kommunikation und Aufgabendelegierung wird in der Evaluation genauer diskutiert.
- Aufgaben durchführen: Für die Aufgabendurchführung wird jeder Einsatzkraft nach der Wahl des Startpunktes ein Avatar zugeteilt. In der Aktionsphase übernehmen die Einsatzkräfte die direkte Steuerung und die Perspektive des Avatars (Abb. 4.26, rechts). In der derzeitigen Realisierung können mit dem Avatar Feuerlöscher aufgenommen, Brandherde gelöscht und flüchtende Menschen geblockt werden.
- Trainingsszenario manipulieren: Die Manipulation des laufenden Trainingsszenario ist die Aufgabe des Ausbilders. In der vorliegenden Version des Prototypen wurde dieser Aspekt aus zeitlichen Gründen nicht mehr realisiert.

Neben den reinen Interaktionsmöglichkeiten wurde auch der im Spielkonzept dargestellte Spielablauf realisiert. So gibt es zeitliche Begrenzungen für die Planungs- und Aktionsphase, ein

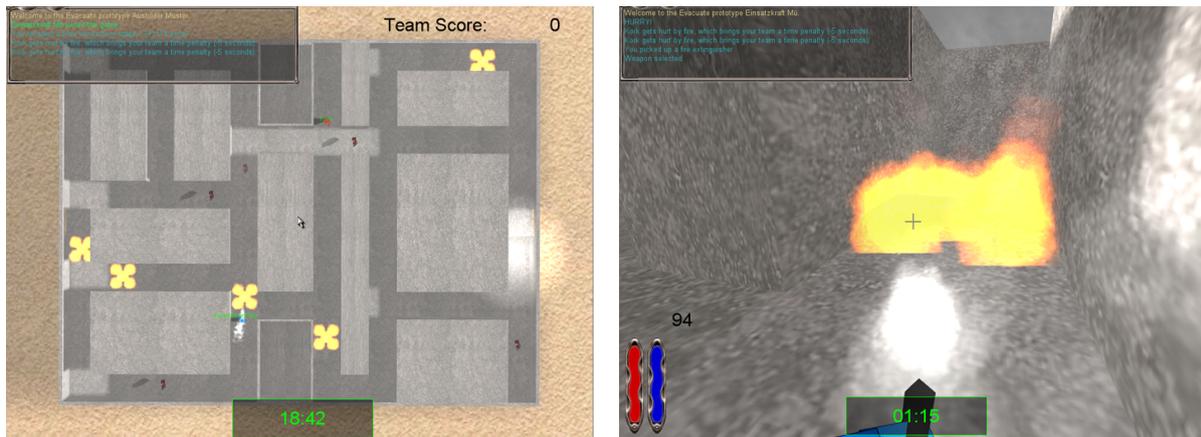


Abb. 4.26: Benutzerschnittstellen in der Aktionsphase: links für den Einsatzleiter und rechts für die Einsatzkräfte

einfaches Punktesystem und die Feststellung von Gewinn bzw. Niederlage, um das Training damit zu beenden bzw. zu wiederholen.

Die flüchtenden Menschen werden durch einen KI-Controller gesteuert, der als nebenläufiger Thread im Spielserver läuft. Das Verhalten wurde hier sehr einfach gestaltet. Dem KI-Controller werden dabei alle möglichen Wegpunkte für ein spezielles Trainingsszenario über eine manuell verfasste Konfigurationsdatei mitgeteilt. Nach dem Zufallsprinzip werden diese Wegpunkte durch die computergesteuerten Akteure abgelaufen. Eine Weiterentwicklung des eingesetzten KI-Controller sollte dahin gehen, dass die Wegpunkte eines Trainingsszenario nicht manuell definiert werden müssen, sondern auf Basis des geometrischen Aufbaus der virtuellen Umgebung automatisch ermittelt werden.

# 5 Evaluation

Um den Nutzen des im Rahmen dieser Arbeit konzipierten und prototypisch realisierten Co-located Collaborative Training Environment (CCTE) zu untersuchen, wurde eine erste Evaluation durchgeführt. Sie erfolgte auf Basis einiger stichprobenartigen Testläufe mit verschiedenen Probanden.

## 5.1 Untersuchungsschwerpunkte

Für die Überprüfung der Umsetzung sämtlicher an das System gestellten Anforderungen, die auf dem in Abschnitt 3.4.1 skizzierten Beispielszenario für Gruppentrainings basieren, wurden in dieser Evaluation die folgenden Untersuchungsschwerpunkte gelegt:

**Grundlegende Funktionsweise:** Dieser Schwerpunkt gilt der Überprüfung der Funktionalität des gesamten Systems. Da ein Großteil der in der Konzeption dargestellten Komponenten realisiert worden ist, muss sichergestellt werden, dass das System die in Abschnitt 3.4.2 skizzierten funktionalen Anforderungen erfüllt und damit die beschriebenen Anwendungsfälle der Benutzer unterstützt.

**Lerneffektivität:** Für das Beispielszenario wurden Trainingsziele definiert (Abschnitt 3.4.1), die durch das spielbasierte Gruppentraining des Einsatzteams im CCTE geschult werden sollen. Hier gilt es zu prüfen, inwieweit diese Trainingsziele in dem vorliegende System effektiv unterstützt werden.

**Aufbau der Infrastruktur:** Für die Evaluation wurde der in Abschnitt 4.1.3 vorgeschlagene Aufbau für die Infrastruktur des CCTE verwendet. Diesbezüglich muss überprüft werden, ob sich dieser Aufbau als sinnvoll erweist und welche Alternativen sich gegebenenfalls anbieten.

**Verwendung der Powerwall:** Die Powerwall nimmt in der gesamten Konzeption eine wichtige Schlüsselposition ein. Daher wird hier genauer untersucht, wie die Powerwall tatsächlich genutzt wird und ob sie für das Training die gewünschten Vorteile mit sich bringt.

Im Folgenden werden zunächst der Aufbau und der Ablauf der Untersuchung geschildert (Unterkapitel 5.2). Anschliessend werden die wichtigsten Ergebnisse bezüglich der hier dargestellten Untersuchungsschwerpunkte zusammengefasst (Unterkapitel 5.3).

## 5.2 Aufbau und Ablauf der Untersuchung

Die Infrastruktur des Trainingsraums wurde gemäß der Darstellung in der Konzeption aufgebaut (vgl. Abschnitt 4.1.3). Die Abbildung 5.1 veranschaulicht diesen exemplarischen Aufbau. Der Einsatzleiter interagiert unmittelbar mit der Powerwall. Die Interaktion erfolgt im Stehen, um die Bewegungsfreiheit und damit die Interaktionsbereiche ausnutzen zu können (vgl. Abschnitt 3.2.3). Die Einsatzkräfte agieren mit stationären Systemen, die nebeneinander im Global Context der Powerwall positioniert wurden. Der Ausbilder überblickt das Vorgehen des Einsatzteams aus dem Hintergrund. Ihm steht dafür ebenfalls ein stationäres System zur Verfügung.



Abb. 5.1: Aufbau der Infrastruktur des CCTE für die Untersuchungen der Evaluation

Für die Untersuchung wurden mehrere Testdurchläufe mit demselben Trainingsszenario durchgeführt. Das Trainingsszenario beinhaltet ein einfaches (fiktives) Übungsgebäude und jeweils fünf zufällig verteilte Brandherde und Feuerlöscher. Das Einsatzziel des Szenarios war die Rettung einer einzelnen Person, die sich wie im Spielkonzept 4.4 beschrieben durch das Übungsgebäude bewegt. Nach jedem Testdurchlauf wurden die Rollen innerhalb des Einsatzteams gewechselt, so dass jeder Proband Erfahrungen als Einsatzleiter und als Einsatzkraft sammeln konnte.

Da der aktuelle Prototyp noch keine explizite Unterstützung des Ausbilders bietet, wird diese Rolle in weiteren Ausführungen dieser Evaluation nicht weiter betrachtet.

## 5.3 Ergebnisse der Untersuchung

Die folgenden Untersuchungsergebnisse entstammen der Beobachtung des Verhaltens und der anschließenden Befragung der Probanden.

### 5.3.1 Grundlegende Funktionsweise

Die Probanden konnten sämtliche in der Realisierung dargestellten Funktionalitäten des Systems nutzen (vgl. Abschnitt 4.5). Es wurde jeweils ein neues Training mit zuvor beschriebenen Trainingsszenario erstellt und gestartet. Der Trainingseinsatz wurde in der Planungsphase gemeinsam vorbereitet und in der Aktionsphase durchgeführt. Alle Benutzer konnten dabei den rollenspezifischen Verantwortlichkeiten in dem Spiel mit den zur Verfügung stehenden Interaktions- und Kommunikationsmöglichkeiten nachgehen.

Seitens der Probanden gab es Bedenken bezüglich der Steuerung der virtuellen Avatare der Einsatzkräfte. Die Steuerung erfolgt - für solche Spiele typisch - über eine Kombination von Tastatur- und Mauseingaben (Pfeiltasten für die Bewegungsrichtung, Mauseachsen für die Blickrichtung). Für Benutzer, die mit einer solchen Steuerung nicht vertraut sind, wird die Benutzbarkeit des Systems damit negativ beeinflusst. Hier ist demnach also entweder eine gewisse Übungsphase im Vorfeld des eigentlichen Trainings oder eine intuitivere Steuerungsmöglichkeit erforderlich.

### 5.3.2 Lerneffektivität

Die Überprüfung der Lerneffektivität orientiert sich an den in Abschnitt 3.4.1 skizzierten Trainingszielen des Beispielszenarios für Gruppentrainings:

**Planung des Einsatzes:** Das konkrete Ziel ist hier, dass das Einsatzteam unter Zeitdruck mit den zur Verfügung stehenden Informationen in der Lage ist, eine möglichst optimale initiale Vorgehensweise zu planen.

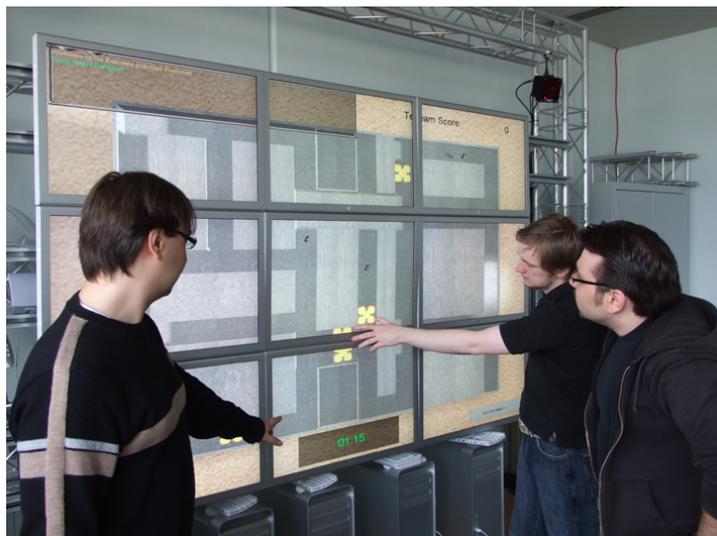


Abb. 5.2: Das Einsatzteam bei der Planung des Trainingseinsatzes

Die Abbildung 5.2 zeigt das Einsatzteam bei einer solchen Planung. Die Probanden bevorzugten es dabei überwiegend, die Planung gemeinsam im *Detail Context* der Powerwall durchzuführen. Die zur Verfügung stehende Zeit für die Planung wurde in den ersten Testläufen in der Regel vollständig ausgenutzt. In den späteren Testläufen hat sich hier ein Lerneffekt eingestellt, was sich dadurch bemerkbar machte, dass sich die durchschnittlich benötigte Zeit für die Planung zum Ende der Untersuchungen deutlich verringert hat. Bemängelt wurde an der Planungsphase lediglich, dass die Auswahl der Startpunkte nur an den Arbeitsplätzen der Einsatzkräfte möglich ist. Den Aussagen der Probanden zufolge sei es hier besser, wenn man die Startpunkte direkt im Rahmen der Planung an der Powerwall, d.h. mit der Benutzerschnittstelle des Einsatzleiters, auswählen könnte.

**Koordination des laufenden Einsatzes:** Bei der Koordination des Einsatzes stehen die Identifizierung und Delegation von Aufgaben bzw. Teilaufgaben im Trainingszenario im Vordergrund, so dass der gesamte Zeitaufwand minimiert wird.

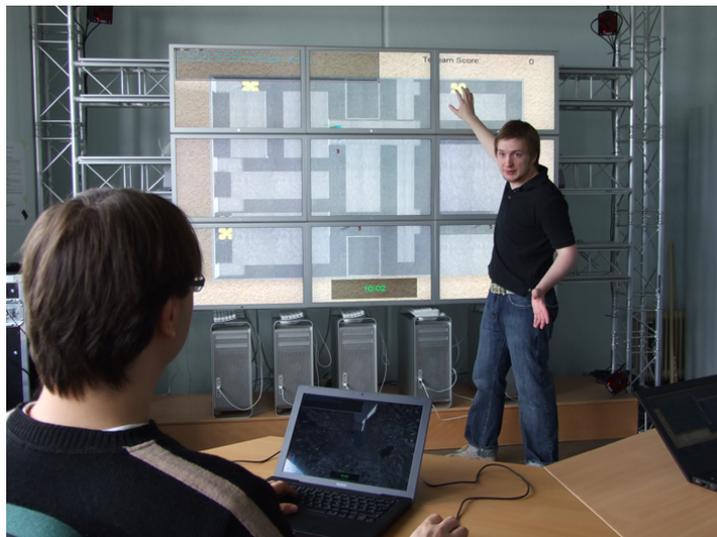


Abb. 5.3: Koordination des Trainingseinsatzes durch verbale und nonverbale Kommunikation

In dem Spielkonzept war vorgesehen, dass der Einsatzleiter die Möglichkeit hat, Wegpunkte für die Einsatzkräfte zu setzen. Dieser Aspekt wurde, wie in Abschnitt 4.5.7 besprochen, aus Zeitgründen nicht mehr realisiert. Stattdessen erfolgte die Koordination bei den Testläufen auf verbaler und nonverbaler Kommunikation. Die Einsatzleiter formulierten dazu in der Regel Sprachanweisungen, um einer Einsatzkraft die Aufgabe und die Zielposition mitzuteilen, und unterstrichen diese Anweisungen durch Gesten sowie durch die Aufnahme von Blickkontakt (Abb. 5.3).

Das Fehlen einer softwaregestützten Koordinationsmöglichkeit empfanden die Probanden aufgrund der geringen Anzahl von Einsatzkräften (zwischen 2 und 3 Personen) nicht als störend und wurde sogar positiv bewertet. Für eine größere Anzahl von Einsatzkräf-

ten wurde seitens der Einsatzleiter jedoch gemutmaßt, dass der Zeitwand für eine verbale Kommunikation jeder Aufgabenzuweisung zu hoch sei.

**Kommunikation und Kooperation der Teammitglieder:** Ein wiederholtes Durchlaufen der Trainingsszenarien soll dazu beitragen, beim Einsatz auftauchende Situationen bzw. Probleme präziser schildern und gemeinsam bewältigen zu können.

Hier konnte insgesamt festgestellt werden, dass sich die Kommunikation zwischen den Teammitgliedern im Verlauf der Untersuchungen entwickelt hat. Anweisungen und Absprachen wurden knapper und präziser. Der größte Kommunikationsfluss war zwischen der Einsatzleitung und den Einsatzkräften zu vernehmen. Die Kommunikation zwischen den Einsatzkräften erfolgte seltener und in der Regel auch nur bei „Sichtkontakt“ des Avatars des Mitspielers. Die Entwicklung der Kommunikation führte unmittelbar zu einer erhöhten Kooperation des Einsatzteams, was sich dadurch wiedergespiegelt hat, dass das Trainingsszenario häufiger erfolgreich beendet werden konnte.

**Räumliche Orientierung:** Durch wiederholtes Training im CCTE soll sich das Einsatzteam mit dem Aufbau des Übungsgebäudes vertraut machen, um damit möglicherweise eine gewisse „Ortskenntnis“ für ein real existierendes Vorbild des Gebäudes aufzubauen.

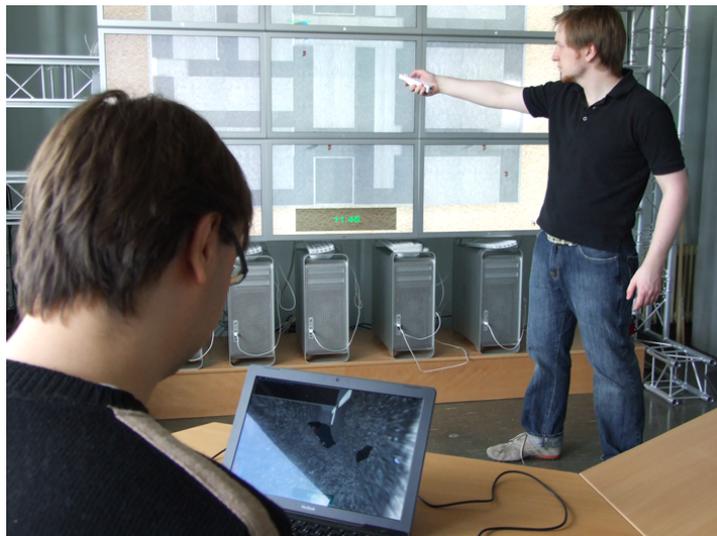


Abb. 5.4: Globale und lokale Sichten der Teammitglieder bei der Durchführung des Trainingseinsatzes

In dem verwendeten Trainingsszenario wurde ein fiktives Übungsgebäude verwendet, wodurch in dieser Evaluation kein eindeutiger Beleg für die Übertragbarkeit von virtueller auf realer Umgebung geliefert werden kann. Den Aussagen der Probanden zufolge, hat sich jedoch ihre räumliche Orientierung im dem Übungsgebäude im Verlauf der Untersuchungen verbessert. Insbesondere sei dies auf den ständigen Rollenwechsel und damit auf die unterschiedlichen Sichtweisen (global und lokal) auf das gesamte Trainingsszenario zurückzuführen (Abb. 5.4). Diese Auffassung deckt sich mit den Untersuchungen

von [Guy u. a. \(2005\)](#), bei denen gezeigt wird, dass sich unterschiedliche Perspektiven auf eine virtuelle Umgebung positiv auf die räumliche Orientierung auswirken (vgl. Abschnitt 3.2.2).

**Reaktion auf dynamische Situationsänderungen:** Da die Manipulation des laufenden Trainingsszenarios durch den Ausbilder in der vorliegenden Version des Prototypen aus zeitlichen Gründen nicht mehr realisiert wurde, konnte dieses Trainingsziel nicht evaluiert werden.

Mit Ausnahme der Reaktion auf dynamische Situationsänderungen kann insgesamt festgehalten werden, dass das vorliegende System in Bezug auf die gewünschten Trainingsziele eine effektive Unterstützung bietet.

### 5.3.3 Aufbau der Infrastruktur

Für die Evaluation des Systems wurde die Infrastruktur des Trainingsraums, wie in der Konzeption in Abschnitt 4.1.3 vorgesehen, aufgebaut. Wie dort angesprochen, muss für die gewählte Infrastruktur geklärt werden, welche alternativen Aufbaumöglichkeiten denkbar und bezüglich der Trainingseffektivität möglicherweise günstiger sind.

Für den Einsatzleiter als Hauptnutzer der Powerwall besteht wenig Diskussionsbedarf. Er ist auf dieses Anzeigemedium angewiesen und muss es für die ihn zugewiesenen Verantwortlichkeiten innerhalb des CCTE uneingeschränkt nutzen können.

Die Positionierung der Arbeitsplätze der Einsatzkräfte in der Infrastruktur des CCTE bedarf jedoch einer Überprüfung. Die Frage ist hier, ob die Einsatzkräfte tatsächlich die globale Sicht der Powerwall benötigen bzw. ob sie diese zur Simulation der realen Umstände überhaupt verwenden dürfen. Hier könnte man nämlich z.B. argumentieren, dass die Einsatzkräfte in einem realen Einsatz nur über die eingeschränkte Sicht verfügen. Diesbezüglich lassen sich alternative Aufbaumöglichkeiten für die Trainingsumgebung ableiten. Eine einfache Lösung wäre, die Arbeitsplätze der Einsatzkräfte umzudrehen, damit diese mit dem Rücken zur Powerwall agieren. Eine striktere Alternative wäre z.B. die Einsatzleitung und Einsatzkräfte in zwei verschiedenen Räumen zu platzieren. Während man bei der ersten Alternative nur die Face-to-Face-Kommunikation unterbindet, wird bei der zweiten Alternative die Eigenschaft „co-located“ komplett aufgehoben. In diesem Fall stellt sich jedoch die grundsätzliche Frage, ob sich das Training „zur gleichen Zeit am gleichen Ort“ im CCTE, gegenüber eines Trainings in einer verteilten virtuellen Trainingsumgebung (z.B. *Active Worlds Educational Universe (AWEDU, 2007)*), überhaupt als vorteilhaft erweist.

Betrachtet man die Ergebnisse bezüglich der Lerneffektivität (Abschnitt 5.3.2) ist erkennbar, dass insbesondere die Möglichkeit zu einer unmittelbaren persönlichen Kommunikation (verbal und nonverbal) in der Planungs- und in der Aktionsphase einen wesentlichen Beitrag dazu geleistet hat, die gesamte Effektivität des Einsatzteams<sup>1</sup> zu erhöhen. Die Probanden haben

<sup>1</sup>Die Effektivität des Einsatzteams wird an dieser Stelle anhand der erfolgreich beendeten Trainingsdurchläufe gemessen.

überdies angemerkt, dass sie während ihrer Tätigkeit als Einsatzkräfte, die globale Sicht der Powerwall zur allgemeinen Orientierung als Ergänzung zur lokalen Sicht mitbenutzt haben. Zusammenfassend lässt sich also festhalten, dass sich der gewählte Aufbau der Infrastruktur als vorteilhaft bezüglich der Trainingseffektivität erwiesen hat.

Inwieweit sich die oben dargestellten Alternativen auf die hier festgestellte Trainingseffektivität auswirken, wurde nicht explizit untersucht. Bei einer Unterbindung der Face-to-Face-Kommunikation bzw. einer Aufhebung der Eigenschaft „co-located“ lässt sich an dieser Stelle, aufgrund der Beobachtungen und des Feedbacks der Probanden, jedoch zunächst vermuten, dass es zu einer Verringerung der gesamten Trainingseffektivität kommen würde. Für konkrete Ergebnisse ist eine weiterführende Arbeit notwendig, bei der die Trainingseffektivität der verschiedenen Alternativen durch eine systematisch geführte Untersuchung genauer evaluiert wird.

### 5.3.4 Verwendung der Powerwall

Da die Powerwall eines der zentralen Instrumente innerhalb des CCTE darstellt, erfolgt in diesem Abschnitt eine gesonderte Betrachtung, wie deren Benutzung bei den Testläufen mit den Probanden zu bewerten ist.

#### Nutzung der Interaktionsbereiche

Der physikalische Raum vor der Powerwall wurde gemäß der Konzeption in die drei Interaktionsbereiche *Detail Context*, *Context Switch* und *Global Context* (Abschnitt 3.2.3) aufgeteilt. Die Einteilung der Entfernungen dieser Bereiche erfolgte in den Untersuchungen nach „Augenmaß“, da die zugrundeliegenden Arbeiten (z.B. [Streitz u. a. \(2003\)](#) und [Vogel und Balakrishnan \(2004\)](#)) dafür keine konkreten Werte geliefert haben. Die Abbildung 5.5 illustriert die Entfernungen, die für den prototypischen Aufbau des System zugrundegelegt wurden.



Abb. 5.5: Interaktionsbereiche des Einsatzleiters: Detail Context (links), Context Switch (mittig) und Global Context (rechts)

Wie zuvor in Abschnitt 5.3.2 beschrieben, neigten die Probanden in der Planungsphase dazu, den Einsatz vorwiegend gemeinsam im *Detail Context* vorzubereiten.

In der Aktionsphase, in der nur der Einsatzleiter *aktiv* mit der Powerwall arbeitet, zeigte sich ebenfalls eine Tendenz dazu im *Detail Context* der Powerwall zu arbeiten. Im Bereich *Global Context* haben die jeweiligen Einsatzleiter seltener agiert. Eine explizite Nutzung des Bereichs *Context Switch* konnte nicht beobachtet werden.

Nach den Befragungen der Probanden könnte diese unerwartete Nutzungsverteilung daran liegen, dass das physische Ausmaß der Powerwall noch „klein genug“ ist, um auch im *Detail Context* alle vorhandenen Informationen zu überblicken.

### Interaktion

In dem vorliegenden Prototyp wurden keine expliziten Interaktionskonzepte für die *Close-range* und *Distant Interaction* umgesetzt bzw. integriert. Die Interaktion der Einsatzleiter erfolgte in den Testläufen daher mit der *physischen Navigation* und klassischen Eingabegeräten (Maus und Tastatur).



Abb. 5.6: Der Einsatzleiter im detaillierten Kontext der Powerwall

Nach den Testläufen wurden den Probanden die möglichen Interaktionskonzepte vorgestellt und dahingehend befragt, welches Konzept sie für ihre Vorgehensweise als nützlich erachtet hätten. Ein Großteil der Befragten sprach sich dabei für ein berührungsempfindliches System aus. Dies korreliert unmittelbar mit der zuvor dargestellten Tatsache, dass die Probanden hauptsächlich im *Detail Context* der Powerwall gearbeitet haben (Abb. 5.6). Generell wurde in Ergänzung dazu ein spezielles Zeigegerät als sinnvoll angesehen, insbesondere wenn die Darstellungsfläche der Powerwall vergrößert wird.

### 5.3.5 Weitere Ergebnisse

An dieser Stelle werden einige vereinzelte Ergebnisse zusammengefasst, die sich bei den Untersuchungen ergeben haben:

**Auflösung der Powerwall:** Bei den ersten Testläufen wurde die Powerwall in einer kumulierten Auflösung von 3840x1600 Pixeln (1280x800 pro Bildschirm) betrieben. In den weiteren Testläufen wurde die Auflösung auf 7680x3200 Pixel (2560x1600 pro Bildschirm) verdoppelt. Die Probanden konnten bei der Durchführung des Trainings zwischen den beiden Betriebsmodi weder Vor- noch Nachteile erkennen. Offenbar liegt dies an der „groben“ Gestaltung des exemplarischen Trainingsszenarios. Es beinhaltet keine Detailinformationen (wie z.B. kleine Hinweisschilder in den Gängen des Gebäudes), deren Erkennbarkeit durch eine höhere Auflösung gesteigert wird. In einer weiterführenden Evaluation gilt es hier zu prüfen, wie detailliert und komplex die Trainingsszenarien gestaltet werden können bzw. müssen, um die oben angeführten Auflösungen vollständig „auszureizen“.

**Steuerung des Avatars über die globale Sicht:** Ein Proband hat in der Rolle einer Einsatzkraft zur Steuerung des Avatars hauptsächlich die globale Sicht der Powerwall verwendet. Dieses Verhalten ist interessant und lässt sich möglicherweise für weiterführende Arbeiten, die sich mit alternativen Interaktionskonzepten für hochauflösende Bildschirmwände beschäftigen, gezielt einsetzen.

Die Untersuchungen haben insgesamt gezeigt, dass das System grundsätzlich zum Training von Gruppen eingesetzt werden kann. Die Verwendung eines Serious Game als Grundlage für den Wissenstransfer ist bei den Probanden auf gute Akzeptanz gestoßen.

Wie bereits im Verlauf der Ausführungen in diesem Kapitel erwähnt, basieren sämtliche Ergebnisse dieser Evaluation auf einer unabhängigen und subjektiven Betrachtung des prototypischen CCTE. Um den tatsächlichen Mehrwert des Systems festzustellen, müssen in weiteren systematisch durchgeführten Untersuchungen qualitative Vergleiche zu alternativen Trainingsmethoden erhoben werden, z.B. zum Training in der verteilten virtuellen Umgebung von *Active Worlds Educational Universe* (AWEDU, 2007).

# 6 Resümee

Zu Beginn dieses Kapitels wird eine Zusammenfassung der gesamten Arbeit gegeben (Unterkapitel 6.1). Anschließend werden in Unterkapitel 6.2 die in dieser Arbeit gewonnenen Erkenntnisse bewertet. Zum Abschluss werden mögliche technische und organisatorische Fortsetzungen der entwickelten Co-located Collaborative Training Environment sowie allgemein zu erwartende Fortschritte in den Bereichen Serious Games und Collaborative Workspaces angesprochen (Unterkapitel 6.3).

## 6.1 Zusammenfassung

Im Rahmen dieser Arbeit wurde mit dem Co-located Collaborative Training Environment (CC-TE) für spielbasierte Teamtrainings eine spezielle Nutzform eines Co-located Collaborative Workspace (CCW) realisiert. Für den CCTE wurde ein exemplarisches Serious Game namens „Evacuate“ entwickelt, auf dessen Basis Einsatzteams der Feuerwehr auf spielerische Art und Weise die Evakuierung von Personen aus brennenden Gebäuden trainieren können. Die unterschiedlichen Rollen werden beim Teamtraining im CCTE unterstützt, indem dieser als ein System mit skalierbarer Visualisierung umgesetzt wurde. Verschiedene Anzeigemedien im CCTE werden zu einem Visualisierungskontext zusammengeschlossen und bieten unterschiedliche Perspektiven auf die zugrundeliegende virtuelle Trainingsumgebung des Spiels. Eine Schlüsselfunktion nimmt im CCTE eine hochauflösende Powerwall ein, die als strategisches Planungsinstrument und als gemeinsame Übersicht für das gesamte Einsatzteam fungiert.

In Kapitel 2 wurden dazu zunächst die wesentlichen Grundlagen für diese Arbeit dargestellt. Neben dem Forschungsgebiet der Computergestützten Gruppenarbeit (CSCW) wurden dort verschiedene Einsatzorte und Installationsformen von skalierbaren Visualisierungssystemen vorgestellt. Weiterhin wurde ein Überblick der Forschungen und Projekte im Bereich Serious Games gegeben. Dabei konnte aufgezeigt werden, dass sich kollaborative Serious Games als eine Ausprägung von CSCW betrachten lassen.

Im ersten Teil der Analyse im Kapitel 3 wurde untersucht, welche Aspekte bei der Entwicklung von Serious Games zu beachten sind und wie sich hochauflösende Anzeigemedien effizient einsetzen lassen. Im Rahmen dieser Untersuchungen wurden einige Designimplikationen abgeleitet, die als Leitfaden bei der Entwicklung von kollaborativen Serious Games herangezogen werden können. Weiterhin wurden die Vorteile bei dem Einsatz von Powerwalls zusammengefasst und verschiedene Strategien vorgestellt, die zur Skalierung von Visualisierungssystemen

verwendet werden können. Im zweiten Teil der Analyse wurden die gesammelten Erkenntnisse zusammengeführt. Dafür wurde ein exemplarisches Anwendungsszenario für spielbasierte Teamtrainings in einem speziellen computergerstützten Trainingsraum (CCTE) skizziert, erwünschte Trainingsziele formuliert und die verschiedenen Rollen sowie deren individuellen Anforderungen identifiziert. Die auf diese Weise zusammengetragenen Informationen wurden daraufhin in funktionale und nicht-funktionale Anforderungen überführt, die an das System dieser Arbeit zu stellen waren.

Die herausgearbeiteten Anforderungen wurden in einem Systemkonzept umgesetzt, welches in Kapitel 4 erläutert wurde. Den Ausgangspunkt des Systementwurfs bildete die Darstellung der grundsätzlichen Gestaltung und des Aufbaus des CCTE. Anhand der Eigenschaften von einem *gemeinsamen Anwendungskontext* und einem *gemeinsamen Anwendungsbereich* wurden dort die Analogien zum Co-located Collaborative Workspace aufgezeigt und wie sich diese mit skalierbaren Visualisierungssystemen auf den CCTE übertragen lassen. Im Anschluss daran wurde die Systemarchitektur vorgestellt, deren wesentlicher Bestandteil das Basissystem für spielbasierte Simulationen darstellt. Mit diesem ist es möglich, verschiedene kollaborative Spielkonzepte für Trainingsmaßnahmen in den CCTE zu integrieren. Zudem wurde im Rahmen dieses Kapitels der Entwurf und die Realisierung einer hochauflösenden Bildschirmwand, die eine Schlüsselfunktion in der Gesamtkonzeption des CCTE einnimmt, dokumentiert. Für eine erste Evaluation des Systems in dieser Arbeit wurde ein exemplarisches Spielkonzept namens „Evacuate“ ausgearbeitet und mit Hilfe des Basissystems umgesetzt. Zum Abschluss des Kapitels wurde der momentane Stand der gesamten Realisierung diskutiert. Dabei erfolgte eine Erörterung aller verwendeten Technologien und Entwicklungswerkzeuge, sowie der getroffenen Realisierungsentscheidungen in Bezug auf die einzelnen Komponenten des Systementwurfs.

Zur Überprüfung der Funktionalität und dem Nutzen des in dieser Arbeit konzipierten Systems wurde eine erste Evaluation durchgeführt. Kapitel 5 enthält die Darstellung der Schwerpunkte, des Aufbaus, des Ablaufs und der Ergebnisse der im Rahmen der Evaluation vollzogenen Untersuchung.

## 6.2 Bewertung

Die theoretische Schwerpunkt dieser Arbeit galt der die Untersuchung, wie sich die Konzepte von computergestützten Gruppenarbeitsräumen (CCWs) für spielbasierte co-located Teamtrainings nutzen lassen, um die Zusammenarbeit der Teammitglieder zu fördern und möglicherweise den Lerneffekt bzw. die Qualität der Ausbildung durch kollaborative Serious Games zu erhöhen. Das praktische Ziel dabei war, die Entwicklung einer spielbasierten Lern- und Trainingsumgebung für *co-located* Gruppen, sowie eines exemplarischen Serious Game, welches die Funktionalität dieser Umgebung veranschaulichen soll.

Mit der Entwicklung des *Co-located Collaborative Training Environment* (CCTE) und dem exemplarischen Serious Game „Evacuate“, kann das praktische Ziel als erreicht betrachtet wer-

den. Bei der Gestaltung des CCTE wurde darauf geachtet, die „typischen“ Eigenschaften eines Co-located Collaborative Workspace zu übernehmen, die durch Projekte wie z.B. dem *iRoom* (Johanson u. a., 2004) oder *i-Land* (Fraunhofer-IPSI, 2002) geprägt wurden, und diese durch die Identifikation verschiedener Rollen in geeigneter Form in das Teamtraining zu integrieren.

Bei der Konzeption des Serious Game wurden neben den spielrelevanten Aspekten auch lerntheoretische Modelle berücksichtigt, um die grundlegende Qualität der Ausbildung sicherzustellen. Durch die Trennung des konkreten Spielkonzepts von dem zugrundeliegenden Basissystem für spielbasierte Simulationen, lassen sich für den CCTE verschiedene kollaborative Serious Games entwickeln.

Mit der Konzeption und Realisierung der Powerwall nach einem nicht-invasiven Ansatz wurde ein skalierbares Visualisierungssystem zur Verfügung gestellt, das ohne Anpassungen auch für andere bzw. weiterführende Forschungsarbeiten eingesetzt werden kann.

Im gesamten CCTE kamen ausschließlich Standardhardware-Komponenten zum Einsatz, womit eine vergleichsweise günstige Realisierbarkeit, z.B. gegenüber spezieller Simulationssysteme, belegt werden konnte.

In der ersten Evaluation konnte aufgezeigt werden, dass das System grundsätzlich zum Training von Gruppen eingesetzt werden kann. Die Verwendung eines Serious Game als Grundlage für den Wissenstransfer ist bei den Probanden auf gute Akzeptanz gestoßen. Bislang konnte jedoch noch nicht festgestellt werden, inwieweit sich die Qualität der Ausbildung im CCTE gegenüber alternativer Trainingsmethoden (z.B. klassischer Seminare) einordnen lässt.

## 6.3 Ausblick

Im folgenden werden technische und konzeptionelle Weiterentwicklungsmöglichkeiten des CCTE (Unterkapitel 6.3.1), alternative Anwendungsbereiche (Unterkapitel 6.3.2) und die allgemeinen Zukunftsaussichten von Serious Games (Unterkapitel 6.3.3) diskutiert.

### 6.3.1 Technische und konzeptionelle Weiterentwicklung des CCTE

#### Integration von Interaktionstechnologien

Ein wesentlicher Schritt zur Weiterentwicklung des CCTE stellt die Integration konkreter Interaktionstechnologien für die Powerwall dar. Dafür kommen die verschiedenen Interaktionskonzepte für die *Close Range* und *Distant Interaction* in Betracht, die in Abschnitt 3.2.3 diskutiert wurden.

Im Rahmen des UbiComp-Projekts der HAW Hamburg existieren bereits verschiedene Arbeiten, die sich diesbezüglich nutzen lassen. Eine Möglichkeit für die Interaktion aus der Distanz

stellt z.B. das von [Fischer \(2007\)](#) entwickelte Eingabegerät dar. Die Interaktion aus der unmittelbaren Nähe wird gegenwärtig in einer Reihe von Arbeiten im Kontext von Multitouch-Systemen untersucht (z.B. [Gehn \(2007\)](#)). Besonders interessant ist hier, die verwendeten Technologien miteinander zu kombinieren und einen fließenden Übergang zwischen der Arbeit aus nächster Nähe und aus der Entfernung zu ermöglichen ([Ni u. a., 2006](#)).

### **Integration von physiologischen Indikatoren**

Ein Aspekt der Arbeiten im Rahmen des Rescue-Szenarios ist der Einsatz von tragbaren Computersystemen (Wearable Computing) für die Einsatzkräfte, mit denen unter anderem ein sogenanntes „Body Monitoring“ stattfinden kann ([Hink, 2007a](#)). Damit ist es z.B. möglich, anhand der Herz- oder Atemfrequenz auf den Gesundheitszustand der Einsatzkräfte zu schließen.

Eine Weiterentwicklungsmöglichkeit des Systems besteht hier darin, solche „Body Monitoring“-Mechanismen in das spielbasierte Training einzubinden. Dazu könnte z.B. das Basissystem um physiologische Indikatoren erweitert werden, die sich entsprechend der aktuellen Situation in der virtuellen Trainingsumgebung auf die (virtuelle) Gesundheit der Einsatzkräfte und damit auf ihre Handlungsmöglichkeiten auswirken.

Das Training im CCTE könnte mit dieser Weiterentwicklung dazu beitragen, die Arbeit mit Wearable Computing und Body Monitoring zu schulen, deren Nutzen zu evaluieren und gleichzeitig eine grundlegende Akzeptanz dieser Systeme seitens der Einsatzkräfte zu schaffen.

### **Unterstützung realistischer Darstellungen**

In der momentanen Realisierung wurde die Torque Game Engine verwendet, die nur ein sehr einfaches Physik-System und im direkten Vergleich mit aktuellen Computerspielen eine nicht mehr ganz zeitgemäße, grafische Qualität aufweist. In Hinblick auf Trainingsszenarien, die eine möglichst realitätsgetreue Darstellung von existierenden Schauplätzen erfordern, sollte hier an der Integration moderner Physik- und Grafik-Engines gearbeitet werden.

### **Import bestehender Gebäudemodelle und geografischer Daten**

Die Erstellung von Trainingsszenarien ist mit einem nicht unerheblichen Zeitaufwand verbunden. Insbesondere die Erstellung von realitäts- und maßstabsgetreuen Modellen real existierender Gebäude und Geländeabschnitten kann sich dabei als sehr arbeitsintensiv und damit auch sehr kostspielig erweisen. Ein großer Vorteil wäre die Möglichkeit, bereits bestehende Modelle in die spielbasierte Simulation des CCTE importieren zu können. Wichtige Grundlagen hierfür sind zum einen die Verfügbarkeit solcher Modelldaten und zum anderen eine standardisierte Spezifikation, um diese dreidimensionalen Daten beschreiben zu können.

Eine Ansatzmöglichkeit hierfür bietet [CityGML \(2008\)](#). Dabei handelt es sich um ein einfaches semantisches Informationsmodell, das zur Repräsentation von städtischen Objekten in

3D dient. Es ist als ein offenes Datenmodell und XML-basiertes Format entworfen worden, um virtuelle städtische 3D-Modelle abzulagern und auszutauschen. CityGML kann sowohl für große Bereiche als auch für kleine Gebiete oder einzelne Gebäude angewendet werden, da es verschiedene Abstraktionsstufen der Darstellung anbietet. Insgesamt gibt es fünf aufeinanderfolgende *Levels of Detail* (LOD), bei denen die Modelle mit jedem Level mit Details angereichert werden. Neben den geometrischen Eigenschaften eines Modells können mit CityGML auch topologische, semantische und visuelle Charakteristika berücksichtigt werden. Bereits jetzt existiert eine Vielzahl von Modelldaten, die auf der CityGML-Spezifikation basieren und durch verschiedene Städte und Unternehmen kontinuierlich weiterentwickelt und ergänzt werden.

Der Spezifikationssumfang von CityGML stellt eine Obermenge der für die spielbasierte Simulation erforderlichen Modellbeschreibungen dar (vgl. Abschnitt 4.5.4). Eine Importfunktionalität von CityGML-basierten Daten sollte damit grundsätzlich möglich sein, da bei der Konvertierung in „spielkonforme“ Datenformate keine Informationen fehlen.

### Weitere Evaluationen

Auf Basis weiterer Untersuchungen sollten der Nutzen und der Mehrwert der spielbasierten Ausbildung im CCTE systematischer evaluiert. Hierbei sollten die folgenden Fragestellungen im Vordergrund stehen:

- Ist der vermittelte Inhalt auf die Realität übertragbar?
- Wie lässt sich das System im Vergleich zu alternativen bzw. klassischen Ausbildungsmethoden einordnen?
- Kann dergleiche Lerneffekt in einer Umgebung erreicht werden, in der die Gruppenmitglieder sich nicht zur derselben Zeit an demselben Ort befinden?

Das Ziel sollte sein, aus den Ergebnissen dieser Untersuchungen weitere Weiterentwicklungsmöglichkeiten für das System zu identifizieren oder gegebenenfalls Anpassungen an der Konzeption abzuleiten.

### 6.3.2 Übertragung auf andere Anwendungsbereiche im Bereich CSCW

Die Entwicklung des CCTE basiert auf den Forschungen im Bereich CSCW und hat sich dabei konkret an bestehenden Konzeptionen von Co-located Collaborative Workspaces orientiert. Durch den Schwerpunkt auf die spielbasierte Ausbildung von Gruppen wurde die Nutzform des CCTE in dieser Arbeit spezialisiert. Die Trennung des Basissystems von der eigentlichen spielbasierten Simulation ermöglicht es jedoch, das System wieder zu generalisieren. Damit ist es prinzipiell auf beliebige Anwendungsbereiche übertragbar, in denen die Zusammenarbeit von Gruppen durch hochauflösende Visualisierungen und verschiedener Perspektiven unterstützt werden soll.



Abb. 6.1: Der „Planungsraum Digitale Fabrik“ von [DaimlerChrysler \(2007\)](#)

Ein möglicher Anwendungsbereich findet sich z.B. bei der Planung von Fabrik- bzw. Produktionsanlagen. Eine wesentliche Herausforderung ist hier die optimale Gestaltung der Prozess- und Fertigungsstraßen. Die Abhängigkeiten zwischen menschlichen und maschinellen Gliedern in der Produktionskette müssen sorgsam aufeinander abgestimmt werden. Eine Weiterentwicklung des CCTE ist in diesem Zusammenhang als Planungsinstrument für die beteiligten Ingenieure denkbar. CAD-Modelle neuer Produktionsanlagen könnten importiert, visualisiert und interaktiv „erlebbar“ gemacht werden, um auf diese Weise z.B. unnötige Laufwege oder Fehler in der Fertigungsreihenfolge zu vermeiden. Die Abbildung 6.1 zeigt den „Planungsraum Digitale Fabrik“ von der DaimlerChrysler AG in dem ein ähnliches Konzept verfolgt wird.

Natürlich ist auch ein Einsatz als „klassischer“ CCW denkbar, bei dem die Nutzung als Konferenzraum im Vordergrund steht. Mit der virtuellen Umgebung des Basissystems könnte das System in diesem Zusammenhang z.B. zur Realisierung des von [Köckritz \(2007\)](#) skizzierten verteilten 3D-Desktops eingesetzt werden.

### 6.3.3 Die Zukunft von ernsthaften Spielen

Das Interesse an Serious Games ist seit der Entstehung des Begriffs stetig gewachsen. Aufgrund der zunehmenden Anzahl von Organisationen und Konferenzen, die sich ausschließlich diesem Thema widmen, wird sich der Trend voraussichtlich auch noch in den kommenden Jahren fortsetzen. Zwar konnten bis zum gegenwärtigen Zeitpunkt nur einige wenige Projekte, wie z.B. America's Army<sup>1</sup>, einen nennenswerten Erfolg verzeichnen, die konituiertlichen Be-

<sup>1</sup><http://www.americasarmy.com>

strebungen in diesem Bereich zeigen aber, dass sich die von [Sawyer \(2002\)](#) prognostizierte Zusammenführung der klassischen Industrie mit der Spielebranche „auf einem guten Weg“ befindet.

Bis sich Serious Games jedoch aus ihrem Nischendasein befreien und als eigenständiger und profitabler Wirtschaftszweig etablieren können, wird wohl noch einige Zeit vergehen. Momentan mangelt es noch einem Großteil der existierenden Produkte bzw. Projekte aus diesem Bereich an eigenem Innovationspotential, da in der Regel auf die alleinige Verwendung von bestehenden Konzepten und Technologien aus der „klassischen“ Spielebranche gesetzt wird. Für zukünftige Arbeiten sollte diese einseitige Abhängigkeit überdacht werden. In welcher Art und Weise lassen sich z.B. auch Technologien aus anderen Bereichen für die Entwicklung von Serious Games heranziehen?

Hier ist ein hohes Maß an Kreativität und Experimentierfreude gefragt. Man darf also gespannt sein, welche interessanten Entwicklungen zukünftig im Bereich der ernsthaften Spiele noch zu erwarten sind.

# A Anhang

## A.1 Technische Komponenten der Powerwall

### A.1.1 Displays

Bei dem momentanen Laboraufbau setzt sich die Powerwall aus einer 3x3-Matrix von 30“-Monitoren zusammen. Die genaueren Systemspezifikationen der verwendeten Displaymodelle kann der folgenden Übersicht entnommen werden:

**Modellbezeichnung:** Apple Cinema HD Displays

**Modellnummer:** Modell M9179

**Gesamtanzahl:** 9 Stück

**Anordnung:** 3x3-Matrix

**Bildschirmgröße:** 30“ (75,4 cm)

**Bildschirmtyp:** TFT-Aktivmatrix-LCD-Flachbildschirm

**max. Auflösung:** 2560 x 1600 Pixel

### A.1.2 Rechner

Für den Anschluss und die Ansteuerung der einzelnen Displays sowie zur verteilten Darstellung von Anwendungen wird zur Zeit ein Cluster aus sechs Rechnern eingesetzt:

**Modellbezeichnung:** MacPro Quad Xeon 64-bit Workstation

**Gesamtanzahl:** 6 Stück

**Anordnung:** 5 Clients (Ansteuerung der Displays), 1 Master (Verteilung von Anwendungen)

**Prozessorgeschwindigkeit:** 2.66 GHz (mit 2 Prozessoren und insgesamt 4 Kernen)

**Arbeitsspeicher:** 2 GB (Clients) und 4 GB (Master)

**Busgeschwindigkeit:** 1.33 GHz

**Grafikkarte:** ATI Radeon X1900 XT

## A.2 Inhalt der CD-ROM

Dieser Arbeit ist eine CD-ROM beigefügt. Unter dem Wurzelverzeichnis der CD-ROM befinden sich **Ordner** und `Dateien` mit folgendem Inhalt:

**Masterarbeit** beinhaltet dieses Dokument der Arbeit als Datei `Masterarbeit.pdf`

**Software** enthält den Prototyp des in dieser Arbeit für den CCTE entwickelten Serious Game „Evacuate“. Eine Installation ist nicht erforderlich. Es müssen lediglich alle Inhalte dieses Ordners kopiert werden. Der Start der Anwendung erfolgt über `Evacuate.exe` (Windows) bzw. `Evacuate.bin` (Linux)

# Abbildungsverzeichnis

2.1	Raum-Zeit-Matrix ( <a href="#">Teufel u. a., 1995</a> ) . . . . .	9
2.2	3K-Klassifizierung nach <a href="#">Teufel u. a. (1995)</a> . . . . .	10
2.3	i-LAND: Oben die erste Konzeption und einzelne Roomware-Komponenten, darunter ein mögliches Einsatzszenario ( <a href="#">Fraunhofer-IPSI, 2002</a> ) . . . . .	12
2.4	iRoom der Stanford University ( <a href="#">Johanson u. a., 2004</a> ) . . . . .	13
2.5	Systeminfrastruktur des Ambient-Labors der HAW Hamburg (nach <a href="#">Fischer (2007)</a> )	14
2.6	Links der konzeptionelle Aufbau eines CAVE-Systems ( <a href="#">Robert, 2006</a> ) und daneben eine Realisierung im 3D Lab der University Michigan ( <a href="#">UM3D, 2008</a> ) . . . . .	15
2.7	Powerwall als skalierbares Visualisierungssystem (links von <a href="#">Robert (2006)</a> und rechts von <a href="#">Mechdyne (2007b)</a> ) . . . . .	16
2.8	Curved Screens bieten durch die Wölbung ähnliche Raumeffekte wie in CAVE-Systemen ( <a href="#">Robert, 2006</a> ) . . . . .	17
3.1	Das Media-Richness-Modell für die Telekooperation (nach <a href="#">Reichwald u. a. (1998)</a> )	30
3.2	Medieneigenschaften in der Media-Synchronicity-Theorie ( <a href="#">Schwabe, 2001</a> ) . . .	32
3.3	Traditionelle Arbeitstechniken und Kollaboration an großformatigen, senkrechten Flächen in der Automobilkonstruktion (links von <a href="#">Mini (2006)</a> , rechts von <a href="#">Volkswagen-AG (2007)</a> ) . . . . .	35
3.4	Powerwalls in der Automobilkonstruktion: Links für das Design eines Fahrzeuginnenraums ( <a href="#">Daimler-AG, 2007</a> ) und daneben zur Visualisierung eines komplexen CAD-Modells ( <a href="#">Barco, 2007</a> ) . . . . .	36
3.5	Vergleich verschiedener Visualisierungssysteme im natürlichen Sichtfeld ( <a href="#">Campos u. a., 2007</a> ) . . . . .	40
3.6	Interaktionsebenen nach <a href="#">Streitz u. a. (2003)</a> (links) und nach <a href="#">Vogel und Balakrishnan (2004)</a> (rechts) . . . . .	43
3.7	Generalisiertes Modell für Interaktionsebenen und Nutzungsbereiche . . . . .	44
3.8	Informationseinheiten bei verschiedenen Synchronisationsstrategien in einem verteilten Visualisierungssystem ( <a href="#">Roth, 2005</a> ) . . . . .	48
3.9	Synchronized Execution Model nach <a href="#">Chen u. a. (2001b)</a> . . . . .	50
3.10	Vollständige Replikation (links) und partielle Replikation (rechts) im Synchronized Execution Model ( <a href="#">Chen u. a., 2001b</a> ): . . . . .	51
3.11	Synchronisationsebene bei der System-level programm synchronization ( <a href="#">Chen u. a., 2001b</a> ) . . . . .	52
3.12	Dekomposition und Komposition beim parallelen Rendering (in Anlehnung an <a href="#">Robert (2006)</a> ) . . . . .	53
3.13	Klassifizierung paralleler Rendering-Verfahren ( <a href="#">Roth, 2002</a> ) . . . . .	54

3.14 Aufgaben und Rollen in dem Beispielszenario . . . . .	59
3.15 Allgemeine Anwendungsfälle im CCTE . . . . .	60
3.16 Anwendungsfalldiagramm für die Rolle des Ausbilders . . . . .	62
3.17 Anwendungsfalldiagramm für die Rolle des Einsatzleiters . . . . .	63
3.18 Anwendungsfalldiagramm für die Rolle der Einsatzkraft . . . . .	65
4.1 CCTE als ein System mit skalierbarer Visualisierung: Anzeigemedien als Fenster zu einem gemeinsamen virtuellen 3D-Modell . . . . .	69
4.2 Globale und lokale Sichten auf die virtuelle Trainingsumgebung im CCTE . . . . .	70
4.3 Nutzung der Interaktionsbereiche im CCTE (vgl. Abb. 3.7) . . . . .	71
4.4 MVC in der Spieleentwicklung: links die allgemeinen MVC-Komponenten sowie deren Zusammenspiel und rechts die Übertragung des Entwurfsmusters auf die Komponenten einer Spielanwendung . . . . .	74
4.5 Allgemeine Rollenverteilung beim Client-Server-Modell . . . . .	75
4.6 Aufgaben von Client und Server im Kontext von mehrbenutzerfähigen Spielen . . . . .	75
4.7 Eine grobe Übersicht des CCTE-Systems . . . . .	77
4.8 Die Komponenten des Basissystems nach dem MVC-Entwurfsmuster . . . . .	79
4.9 Control/View-Paare bilden die rollenspezifischen Benutzerschnittstellen zum System . . . . .	81
4.10 Verteilungsmöglichkeit der MVC-Komponenten: eine zentrale Model-Komponente für alle Clients . . . . .	82
4.11 Verteilungsmöglichkeit der MVC-Komponenten: alle Clients verfügen über eine eigene Model-Komponente . . . . .	83
4.12 Wesentliche Architekturkomponenten des Basissystems für spielbasierte Simulationen . . . . .	85
4.13 Spielzustandsschnittstelle als Fassade zum Model: Erweiterbarkeit und Änderbarkeit der Spielsimulation durch Spezialisierung bzw. Austausch der Basisimplementierung . . . . .	88
4.14 Schematische Darstellung der Schritte in einer Grafikipipeline . . . . .	91
4.15 Virtuelle Grafik-API zur Aufteilung der Grafikipipeline nach dem Sort-First-Verfahren . . . . .	92
4.16 Übersicht des Systems für hochauflösende Bildschirmwände . . . . .	93
4.17 Verteilung der Komponenten des Visualisierungssystems . . . . .	94
4.18 Integration des Visualisierungssystems in das Basissystem . . . . .	95
4.19 Die Eingabeverarbeitung dient als Verteilungs- und Synchronisationsschnittstelle des Basissystems . . . . .	97
4.20 Die skalierbare Bildschirmwand im Ambient Labor . . . . .	104
4.21 Gestaltung eines Trainingsgebäudes für das Szenario im Torque Constructor . . . . .	106
4.22 Erstellung eines Trainingssszenarios über den Missionseditor (hier verwendet: Torque World Editor) . . . . .	108
4.23 Das Hauptmenü des Prototypen „Evacuate“ . . . . .	109
4.24 Menüs zum Starten eines neuen Trainingssszenarios durch den Ausbilder (links) und dem Beitritt des Einsatzteams (rechts) . . . . .	110
4.25 Benutzerschnittstellen in der Planungsphase: links für den Einsatzleiter und rechts für die Einsatzkräfte . . . . .	111

---

4.26 Benutzerschnittstellen in der Aktionsphase: links für den Einsatzleiter und rechts für die Einsatzkräfte . . . . .	112
5.1 Aufbau der Infrastruktur des CCTE für die Untersuchungen der Evaluation . . .	114
5.2 Das Einsatzteam bei der Planung des Trainingseinsatzes . . . . .	115
5.3 Koordination des Trainingseinsatzes durch verbale und nonverbale Kommunikation	116
5.4 Globale und lokale Sichten der Teammitglieder bei der Durchführung des Trainingseinsatzes . . . . .	117
5.5 Interaktionsbereiche des Einsatzleiters: Detail Context (links), Context Switch (mittig) und Global Context (rechts) . . . . .	119
5.6 Der Einsatzleiter im detaillierten Kontext der Powerwall . . . . .	120
6.1 Der „Planungsraum Digitale Fabrik“ von <a href="#">DaimlerChrysler (2007)</a> . . . . .	127

# Literaturverzeichnis

- [Ahn u. a. 2004] AHN, Sang C. ; LEE, Tae-Seong ; KIM, Ig-Jae ; KWON, Yong-Moo ; KIM, Hyoung-Gon: Large Display Interaction Using Video Avatar and Hand Gesture Recognition. In: *ICIAR (1)*, 2004, S. 261–268
- [Alvarez u. a. 2007] ALVAREZ, Julian ; RAMPNOUX, Olivier ; JESSEL, Jean-Pierre ; METHEL, Gilles: Serious Game: just a question of posture ? In: *Artificial and Ambient Intelligence convention (Artificial Societies for Ambient Intelligence)*, The Society for the Study of Artificial Intelligence and Simulation of Behaviour, April 2007, S. 420–426
- [AWEDU 2007] AWEDU: *Active Worlds Educational Universe*. Webseite. 2007. – URL <http://www.activeworlds.com/edu>. – Letzter Zugriff am 15.06.2008
- [Bakdash u. a. 2006] BAKDASH, Jonathan Z. ; AUGUSTYN, Jason S. ; PROFFITT, Dennis R.: Large displays enhance spatial knowledge of a virtual environment. In: *APGV '06: Proceedings of the 3rd symposium on Applied perception in graphics and visualization*. New York, NY, USA : ACM, 2006, S. 59–62. – ISBN 1-59593-429-4
- [Ball und North 2005] BALL, Robert ; NORTH, Chris: Effects of tiled high-resolution display on basic visualization and navigation tasks. In: *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*. New York, NY, USA : ACM, 2005, S. 1196–1199. – ISBN 1-59593-002-7
- [Ballagas u. a. 2005] BALLAGAS, Rafael ; ROHS, Michael ; SHERIDAN, Jennifer G.: Sweep and point and shoot: phonecam-based interactions for large public displays. In: *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*. New York, NY, USA : ACM, 2005, S. 1200–1203. – ISBN 1-59593-002-7
- [Barco 2007] BARCO: *CADWall*. Webseite. 2007. – URL <http://www.barco.com>. – Letzter Zugriff am 15.06.2008
- [Baumgartner und Payr 1999] BAUMGARTNER, Peter ; PAYR, Sabine: *Lernen mit Software*. 2. Auflage. Studien Verlag, 1999. – ISBN 978-3706514446
- [Berne 1967] BERNE, Eric: *Games people play: the psychology of human relationships*. New York : Grove Press, 1967
- [Bezerianos und Balakrishnan 2005] BEZERIANOS, Anastasia ; BALAKRISHNAN, Ravin: View and Space Management on Large Displays. In: *IEEE Computer Graphics and Applications* 25 (2005), Nr. 4, S. 34–43. – ISSN 0272-1716

- [Bowman 1999] BOWMAN, Douglas: *Interaction Techniques for Common Tasks in Immersive Virtual Environments*, Georgia Institute of Technology, Dissertation, Juni 1999. – URL <http://people.cs.vt.edu/~bowman/thesis/thesis.pdf>. – Letzter Zugriff am 15.06.2008
- [Brown und Bell 2004] BROWN, Barry ; BELL, Marek: CSCW at play: 'there' as a collaborative virtual environment. In: *CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work*. New York, NY, USA : ACM, 2004, S. 350–359. – ISBN 1-58113-810-5
- [Brown u. a. 1985] BROWN, John S. ; CASHMAN, Paul M. ; MALONE, Thomas: Interfaces in organizations (panel session): supporting group work. In: *SIGCHI Bull.* 16 (1985), Nr. 4, S. 65. – ISSN 0736-6906
- [Buschmann u. a. 1996] BUSCHMANN, Frank ; MEUNIER, Regine ; ROHNERT, Hans ; SOMMERLAD, Peter ; STAL, Michael: *Pattern-Oriented Software Architecture: A System of Patterns*. Bd. 1. Wiley, 1996. – ISBN 0471958697
- [Buxton u. a. 2000] BUXTON, William ; FITZMAURICE, George ; BALAKRISHNAN, Ravin ; KURTENBACH, Gordon: Large Displays in Automotive Design. In: *IEEE Computer Graphics and Applications* 20 (2000), Nr. 4, S. 68–75. – ISSN 0272-1716
- [Campos u. a. 2007] CAMPOS, J. ; NUSSEK, H.-G. ; WALLRAVEN, C. ; MOHLER, B.J. ; BUELT-HOFF, H.H.: Vizualisation and (Mis)Perceptions in Virtual Reality. In: *Tagungsband 10. Workshop Sichtsysteme*. Aachen, Deutschland, November 2007
- [Castel u. a. 2005] CASTEL, Alan D. ; PRATT, Jay ; DRUMMOND, Emily: The effects of action video game experience on the time course of inhibition of return and the efficiency of visual search. In: *Acta Psychologica* 119 (2005), S. 217–230
- [Chen u. a. 2001a] CHEN, Han ; CHEN, Yuqun ; FINKELSTEIN, Adam ; FUNKHOUSER, Thomas ; LI, Kai ; LIU, Zhiyan ; SAMANTA, Rudrajit ; WALLACE, Grant: Data distribution strategies for high-resolution displays. In: *Computers and Graphics* 25 (2001), Nr. 5, S. 811–818
- [Chen u. a. 2001b] CHEN, Han ; CLARK, Douglas W. ; LIU, Zhiyan ; WALLACE, Grant ; LI, Kai ; CHEN, Yuqun: Software Environments For Cluster-Based Display Systems. In: *CCGRID '01: Proceedings of the 1st International Symposium on Cluster Computing and the Grid*. Washington, DC, USA : IEEE Computer Society, 2001, S. 202. – ISBN 0-7695-1010-8
- [Chromium 2007] CHROMIUM: *Chromium: A system for interactive rendering on clusters of graphics workstations*. Webseite. 2007. – URL <http://chromium.sourceforge.net>. – Letzter Zugriff am 15.06.2008
- [CityGML 2008] CITYGML: *Exchange and Storage of Virtual 3D City Models*. Webseite. 2008. – URL <http://www.citygml.org>. – Letzter Zugriff am 18.06.2008
- [Czerwinski u. a. 2003] CZERWINSKI, Mary P. ; SMITH, Greg ; REGAN, Tim ; MEYERS, Brian ; ROBERTSON, George G. ; STARKWEATHER, Gary: Toward characterizing the productivity

- benefits of very large displays. In: *Human-Computer Interaction – INTERACT'03*, IOS Press, 2003, S. 9–16. – ISBN 978-1586033637
- [Daft und Lengel 1986] DAFT, R.L. ; LENGEL, R.: Organizational information requirements, media richness, and structural design. In: *Management Science* 32 (1986), S. 554–571
- [Daimler-AG 2007] DAIMLER-AG: *Design Of The 2007 Mercedes S-Class*. Webseite. 2007. – URL <http://www.daimler.com>. – Letzter Zugriff am 15.06.2008
- [DaimlerChrysler 2007] DAIMLERCHRYSLER: *Planungsraum Digitale Fabrik*. Webseite. 2007. – URL <http://www.tecchannel.de/server/1728252/index4.html>. – Letzter Zugriff am 18.06.2008
- [Dennis und Valacich 1999] DENNIS, Alan R. ; VALACICH, Joseph S.: Rethinking Media Richness: Towards a Theory of Media Synchronicity. In: *HICSS '99: Proceedings of the Thirty-Second Annual Hawaii International Conference on System Sciences* Bd. 1. Washington, DC, USA : IEEE Computer Society, 1999, S. 1017. – ISBN 0-7695-0001-3
- [Eldridge u. a. 2000] ELDRIDGE, Matthew ; IGEHY, Homan ; HANRAHAN, Pat: Pomegranate: a fully scalable graphics architecture. In: *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. New York, NY, USA : ACM Press/Addison-Wesley Publishing Co., 2000, S. 443–454. – ISBN 1-58113-208-5
- [Ellis u. a. 1991] ELLIS, Clarence A. ; GIBBS, Simon J. ; REIN, Gail: Groupware: some issues and experiences. In: *Commun. ACM* 34 (1991), Nr. 1, S. 39–58. – ISSN 0001-0782
- [Faulhaber 1996] FAULHABER, Sven: *Einsatz und Entwicklung von computerunterstützten Lernprogrammen in der medizinischen Aus- und Weiterbildung*. Würzburg, Bayerischen Julius-Maximilians-Universität, Studienarbeit, 1996. – URL <http://ki.informatik.uni-wuerzburg.de/forschung/publikationen/studienarbeiten/faulhaber>. – Letzter Zugriff am 15.06.2008
- [Fischer 2007] FISCHER, Christian: *Entwicklung eines multimodalen Interaktionssystems für computergestützte Umgebungen*, HAW Hamburg, Masterarbeit, 2007. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/fischer.pdf>. – Letzter Zugriff am 15.06.2008
- [Fox u. a. 2000] FOX, Armando ; JOHANSON, Brad ; HANRAHAN, Pat ; WINOGRAD, Terry: Integrating Information Appliances into an Interactive Workspace. In: *IEEE Computer Graphics and Applications* 20 (2000), Nr. 3, S. 54–65. – ISSN 0272-1716
- [Fraunhofer-IPSI 2002] FRAUNHOFER-IPSI: *i-LAND: An Interactive Landscape for Creativity and Innovation*. Webseite. 2002. – URL [http://www.ipsi.fraunhofer.de/ambiente/projekte/projekte/i\\_land.html](http://www.ipsi.fraunhofer.de/ambiente/projekte/projekte/i_land.html). – Letzter Zugriff am 15.06.2008
- [Fraunhofer-IuK 2007] FRAUNHOFER-IUK: *Spielen im Dienste der Wirtschaft*. Webseite. 2007. – URL [http://www.iuk.fraunhofer.de/index2.html?Dok\\_ID=1342&Sp=1](http://www.iuk.fraunhofer.de/index2.html?Dok_ID=1342&Sp=1). – Letzter Zugriff am 15.06.2008

- [Gamma u. a. 1995] GAMMA, Erich ; HELM, Richard ; JOHNSON, Ralph ; VLISSIDES, John: *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1995. – ISBN 0201633612
- [Gehn 2007] GEHN, Stefan: *Intuitive Gesten für Multitouch-Displays*, HAW Hamburg, Seminararbeit, 2007. – URL <https://users.informatik.haw-hamburg.de/~ubicomp/projekte/master07-08/gehn/bericht.pdf>. – Letzter Zugriff am 15.06.2008
- [Gerstenmaier und Mandl 1994] GERSTENMAIER, Jochen ; MANDL, Heinz: Wissenserwerb unter konstruktivistischer Perspektive. In: *Zeitschrift für Pädagogik* 41 (1994), S. 867–888
- [Grudin 1994] GRUDIN, Jonathan: Computer-supported cooperative work: history and focus. In: *Computer* 27 (1994), Nr. 5, S. 19–26. – ISSN 0018-9162
- [Grudin 2001] GRUDIN, Jonathan: Partitioning digital worlds: focal and peripheral awareness in multiple monitor use. In: *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM, 2001, S. 458–465. – ISBN 1-58113-327-8
- [Guy u. a. 2005] GUY, Ben ; BIDWELL, Nicola J. ; MUSUMECI, Phillip: GamePlan: Serious Gaming for Place Making. In: *IE2005: Proceedings of the second Australasian conference on Interactive entertainment*. Sydney, Australia, Australia : Creativity & Cognition Studios Press, 2005, S. 252. – ISBN 0-9751533-2-3
- [Haag 1995] HAAG, Martin: *Gesamtkonzept für die Entwicklung und den Einsatz von computerunterstützten Lehr-/Lernsystemen in der Mediziner Ausbildung an der Universität Heidelberg*, Universität Heidelberg/Fachhochschule Heilbronn, Diplomarbeit, 1995
- [Harteveld u. a. 2007] HARTEVELD, Casper ; GUIMARÃES, Rui ; MAYER, Igor ; BIDARRA, Rafael: Balancing Pedagogy, Game and Reality Components Within a Unique Serious Game for Training Levee Inspection. In: *Edutainment* Bd. 4469, Springer, 2007, S. 128–139. – ISBN 978-3-540-73010-1
- [Hasenkamp u. a. 1994] HASENKAMP, Ulrich ; KIRN, Stefan ; SYRING, Michael: *CSCW - Computer supported cooperative work: Informationssysteme für dezentralisierte Unternehmensstrukturen*. Bonn : Addison-Wesley, 1994. – ISBN 3-89319-648-X
- [HAW-Hamburg 2002] HAW-HAMBURG: *UbiComp Projekt*. Webseite. 2002. – URL <http://users.informatik.haw-hamburg.de/~ubicomp>. – Letzter Zugriff am 15.06.2008
- [Hink 2007a] HINK, Steffen: *Einsatz von Wearable Computing in Disaster Szenarien*, HAW Hamburg, Seminararbeit, 2007. – URL <https://users.informatik.haw-hamburg.de/~ubicomp/projekte/master06-07/hinck/report.pdf>. – Letzter Zugriff am 15.06.2008
- [Hink 2007b] HINK, Steffen: *RESCUE: Überblick über Wearable Computing in extremen Situationen*, HAW Hamburg, Seminararbeit, 2007. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master06-07/hinck/report.pdf>

[informatik.haw-hamburg.de/~ubicomp/projekte/master06-07-aw/hinck/report.pdf](http://informatik.haw-hamburg.de/~ubicomp/projekte/master06-07-aw/hinck/report.pdf). – Letzter Zugriff am 15.06.2008

- [Huizinga 1956] HUIZINGA, Johan: *Homo ludens: vom Ursprung der Kultur im Spiel*. Hamburg : Rowohlt-Taschenbuch-Verlag, 1956
- [Humphreys u. a. 2001] HUMPHREYS, Greg ; ELDRIDGE, Matthew ; BUCK, Ian ; STOLL, Gordon ; EVERETT, Matthew ; HANRAHAN, Pat: WireGL: a scalable graphics system for clusters. In: *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. New York, NY, USA : ACM, 2001, S. 129–140. – ISBN 1-58113-374-X
- [Humphreys u. a. 2002] HUMPHREYS, Greg ; HOUSTON, Mike ; NG, Ren ; FRANK, Randall ; AHERN, Sean ; KIRCHNER, Peter D. ; KLOSOWSKI, James T.: Chromium: a stream-processing framework for interactive rendering on clusters. In: *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*. New York, NY, USA : ACM, 2002, S. 693–702. – ISBN 1-58113-521-1
- [Irrlicht 2008] IRRLICHT: *Irrlicht Engine - A free open source 3d engine*. Webseite. 2008. – URL <http://irrlicht.sourceforge.net>. – Letzter Zugriff am 15.06.2008
- [Jiang u. a. 2006] JIANG, Hao ; OFEK, Eyal ; MORAVEJI, Neema ; SHI, Yuanchun: Direct pointer: direct manipulation for large-display interaction using handheld cameras. In: *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*. New York, NY, USA : ACM, 2006, S. 1107–1110. – ISBN 1-59593-372-7
- [Johansen 1988] JOHANSEN, Robert: *GroupWare: Computer Support for Business Teams*. New York, NY, USA : The Free Press, 1988. – ISBN 0029164915
- [Johanson u. a. 2002] JOHANSON, Brad ; FOX, Armando ; WINOGRAD, Terry: The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms. In: *IEEE Pervasive Computing* 1 (2002), Nr. 2, S. 67–74. – ISSN 1536-1268
- [Johanson u. a. 2004] JOHANSON, Brad ; FOX, Armando ; WINOGRAD, Terry: *The Stanford Interactive Workspace Project.*, Stanford University, Forschungsbericht, August 2004. – URL <http://hci.stanford.edu/cstr/reports/2004-05.pdf>. – Letzter Zugriff am 15.06.2008
- [Johnson-Lenz und Johnson-Lenz 1981] JOHNSON-LENZ, Peter ; JOHNSON-LENZ, Trudy: Consider the Groupware: Design and Group Process Impacts on Communication in the Electronic Medium. In: *Studies of Computer-Mediated Communications Systems: A Synthesis of the Findings, Computerized Conferencing and Communications Center*, Forschungsbericht 16 (1981)
- [Kahlbrandt 1998] KAHLBRANDT, Bernd: *Software-Engineering. Objektorientierte Software-Entwicklung mit der Unified Modeling Language*. 1998. – ISBN 3-540-63309-X
- [Köckritz 2007] KÖCKRITZ, Oliver: *Verteilter 3D-Desktop mit Remote-Windows für Collaborative Workspaces*, HAW Hamburg, Seminararbeit, 2007. – URL

- <https://users.informatik.haw-hamburg.de/~ubicomp/projekte/master06-07-aw/koeckritz/report.pdf>. – Letzter Zugriff am 15.06.2008
- [Khan u. a. 2005] KHAN, Azam ; MATEJKA, Justin ; FITZMAURICE, George ; KURTENBACH, Gordon: Spotlight: directing users' attention on large displays. In: *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM, 2005, S. 791–798. – ISBN 1-58113-998-5
- [Kindberg und Fox 2002] KINDBERG, Tim ; FOX, Armando: System Software for Ubiquitous Computing. In: *IEEE Pervasive Computing* 1 (2002), Nr. 1, S. 70–81. – ISSN 1536-1268
- [König u. a. 2007] KÖNIG, Werner A. ; BIEG, Hans-Joachim ; SCHMIDT, Toni ; REITERER, Harald: Position-independent interaction for large high-resolution displays. In: *IHCI'07: Proceedings of IADIS International Conference on Interfaces and Human Computer Interaction 2007*, IADIS Press, Juli 2007, S. 117–125
- [Knöpfle 2007] KNÖPFLE, Christian: *High End Projektion und Bedienbarkeit - Interaktion in 3D Umgebungen*, Fraunhofer Institut für Grafische Datenverarbeitung, Workshop, 2007. – URL <http://www.vde.com/de/fg/ITG/Arbeitsgebiete/Fachbereich%202/Documents/Workshop2007/Vortrag%20Knoepfle.pdf>. – Letzter Zugriff am 15.06.2008
- [LambdaVision 2007] LAMBDAVISION: *Ultra-high-resolution Visualization and Networking Instrument for Research and Education in Geoscience and Computer Science*. Webseite. 2007. – URL <http://www.evl.uic.edu/cavern/lambdavisoin>. – Letzter Zugriff am 15.06.2008
- [Mandl u. a. 1997] MANDL, Heinz ; GRUBER, Hans ; RENKL, Alexander: *Information und Lernen mit Multimedia und Internet. 2. Aufl.* Kap. Situiertes Lernen in multimedialen Lernumgebungen, S. 167–178. Weinheim : L.J. Issing; P. Klimsa, 1997. – ISBN 3621273743
- [Maurina 2006] MAURINA, Edward F.: *The Game Programmer's Guide to Torque: Under the Hood of the Torque Game Engine (GarageGames)*. AK Peters, Ltd., März 2006. – ISBN 1568812841
- [McCahill und Lombardi 2004] MCCAHILL, Mark P. ; LOMBARDI, Julian: Design for an Extensible Croquet-Based Framework to Deliver a Persistent, Unified, Massively Multi-User, and Self-Organizing Virtual Environment. In: *C5 '04: Proceedings of the Second International Conference on Creating, Connecting and Collaborating through Computing*. Washington, DC, USA : IEEE Computer Society, 2004, S. 71–77. – ISBN 0-7695-2166-5
- [Mechdyne 2007a] MECHDYNE: *CAVELib*. Webseite. 2007. – URL <http://www.mechdyne.com/integratedSolutions/software/products/CAVELib/CAVELib.htm>. – Letzter Zugriff am 15.06.2008
- [Mechdyne 2007b] MECHDYNE: *Fakespace Systems PowerWall*. Webseite. 2007. – URL <http://www.mechdyne.com/integratedSolutions/displaySystems/products/PowerWall/PowerWall.htm>. – Letzter Zugriff am 15.06.2008

- [Mini 2006] MINI: *Design of the new Mini*. Webseite. 2006. – URL <http://www.carbodydesign.com>. – Letzter Zugriff am 15.06.2008
- [Molnar u. a. 1994] MOLNAR, Steven ; COX, Michael ; ELLSWORTH, David ; FUCHS, Henry: A Sorting Classification of Parallel Rendering. In: *IEEE Computer Graphics and Applications* 14 (1994), Nr. 4, S. 23–32. – ISSN 0272-1716
- [Napitupulu 2007] NAPITUPULU, Jan: *Multihead-Display als interaktives Informationsmedium*, HAW Hamburg, Seminararbeit, 2007. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master06-07/napitupulu/report.pdf>. – Letzter Zugriff am 15.06.2008
- [Ni u. a. 2006] NI, Tao ; SCHMIDT, Greg S. ; STAADT, Oliver G. ; BALL, Robert ; MAY, Richard: A Survey of Large High-Resolution Display Technologies, Techniques, and Applications. In: *VR '06: Proceedings of the IEEE conference on Virtual Reality*. Washington, DC, USA : IEEE Computer Society, 2006, S. 31. – ISBN 1-4244-0224-7
- [Ogre3D 2007] OGRE3D: *OGRE 3D: Open source graphics engine*. Webseite. 2007. – URL <http://www.ogre3d.org>. – Letzter Zugriff am 15.06.2008
- [OI 2006] OI: *Open Inventor: An object-oriented 3D toolkit*. Webseite. 2006. – URL <http://oss.sgi.com/projects/inventor>. – Letzter Zugriff am 15.06.2008
- [OSG 2007] OSG: *OpenSceneGraph: An open source high performance 3D graphics toolkit*. Webseite. 2007. – URL <http://www.openscenegraph.org/projects/osg>. – Letzter Zugriff am 15.06.2008
- [Piening 2007] PIENING, Andread: *Katastrophen-Leitstand: Current Work and projects*, HAW Hamburg, Seminararbeit, 2007. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master06-07-aw/piening/report.pdf>. – Letzter Zugriff am 15.06.2008
- [Raybourn und Bos 2005] RAYBOURN, Elaine M. ; BOS, Nathan: Design and evaluation challenges of serious games. In: *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*. New York, NY, USA : ACM, 2005, S. 2049–2050. – ISBN 1-59593-002-7
- [Raybourn und Waern 2004] RAYBOURN, Elaine M. ; WAERN, Annika: Social learning through gaming. In: *CHI '04: CHI '04 extended abstracts on Human factors in computing systems*. New York, NY, USA : ACM, 2004, S. 1733–1734. – ISBN 1-58113-703-6
- [Reichwald u. a. 1998] REICHWALD, Ralf ; MÖSLEIN, Kathrin ; SACHENBACHER, Hans ; ENGLBERGER, Hermann: *Telekooperation - Verteilte Arbeits- und Organisationsformen*. Springer : Berlin, 1998. – ISBN 3-540-62013-3
- [Rice 1992] RICE, Ronald E.: Task analyzability, use of new media, and effectiveness: a multi-site exploration of media richness. In: *Organization Science* 3 (1992), Nr. 4, S. 475–500

- [Roßberger 2008] ROSSBERGER, Philipp: *Physikbasierte Interaktion in kollaborativen computergestützten Umgebungen*, HAW Hamburg, Masterarbeit, 2008. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/rossberger.pdf>. – Letzter Zugriff am 15.06.2008
- [Robert 2006] ROBERT, Philippe C.: *Methoden zur skalierbaren Visualisierung*, Universität Bern, CGG Seminar, 2006. – URL [http://cgg.unibe.ch/teaching/lectures/ss06/seminar-computergeometrie-und-grafik/hmm/philippe\\_robert\\_skalierbare\\_visualisierung.pdf](http://cgg.unibe.ch/teaching/lectures/ss06/seminar-computergeometrie-und-grafik/hmm/philippe_robert_skalierbare_visualisierung.pdf). – Letzter Zugriff am 15.06.2008
- [Robertson u. a. 2005] ROBERTSON, George ; CZERWINSKI, Mary ; BAUDISCH, Patrick ; MEYERS, Brian ; ROBBINS, Daniel ; SMITH, Greg ; TAN, Desney: The Large-Display User Experience. In: *IEEE Comput. Graph. Appl.* 25 (2005), Nr. 4, S. 44–51. – ISSN 0272-1716
- [Rollings und Morris 2003] ROLLINGS, Andrew ; MORRIS, Dave: *Game Architecture and Design: A New Edition*. New Riders Games, 2003. – ISBN 0735713634
- [Roth 2002] ROTH, Marcus: *Integration paralleler Rendering-Verfahren für lose gekoppelte Systeme in OpenSG*, Fraunhofer Institut für graphische Datenverarbeitung, Forschungsbericht, 2002. – URL [http://www.opensg.org/OpenSGPLUS/symposium/Papers2002/Roth\\_Cluster.pdf](http://www.opensg.org/OpenSGPLUS/symposium/Papers2002/Roth_Cluster.pdf). – Letzter Zugriff am 15.06.2008
- [Roth 2005] ROTH, Marcus: *Parallele Bildberechnung in einem Netzwerk von Workstations*, Universität Darmstadt, Dissertation, 2005. – URL [http://deposit.ddb.de/cgi-bin/dokserv?idn=979227887&dok\\_var=d1&dok\\_ext=pdf&filename=979227887.pdf](http://deposit.ddb.de/cgi-bin/dokserv?idn=979227887&dok_var=d1&dok_ext=pdf&filename=979227887.pdf). – Letzter Zugriff am 15.06.2008
- [Rumbke 2006] RUMBKE, Leif: *Kinetische Semiotik im klassischen Computerspiel*. Köln, Kunsthochschule für Medien, Diplomarbeit, 2006. – URL [http://www.rumbke.de/data/text/\\_x++%20-%20leif%20rumbke%202006.pdf](http://www.rumbke.de/data/text/_x++%20-%20leif%20rumbke%202006.pdf). – Letzter Zugriff am 18.06.2008
- [Sacher 1990] SACHER, Werner: *Computer und die Krise des Lernens. Eine pädagogisch-anthropologische Untersuchung zur Zukunft des Lernens in der Informationsgesellschaft*. Bad Heilbrunn : Klinkhardt, 1990. – ISBN 3781506479
- [Sawyer 2002] SAWYER, Ben: *Serious Games: Improving Public Policy through Game-Based Learning and Simulation*. Webseite. 2002. – URL <http://wwics.si.edu/subsites/game>. – Letzter Zugriff am 15.06.2008
- [Schink 1993] SCHINK, Peter: *Kritik des Behaviorismus - Studienreihe Psychologische Forschungsergebnisse*. Bd. 5. Hamburg, 1993. – ISBN 3-86064-125-5
- [Schlichter u. a. 2001] SCHLICHTER, Johann ; REICHWALD, Ralf ; KOCH, Michael ; MÖSLEIN, Kathrin: Rechnergestützte Gruppenarbeit (CSCW). In: *i-com Zeitschrift für interaktive und kooperative Medien* (2001), Nr. 0, S. 5–11. – ISSN 1618-162X

- [Schwabe 2001] SCHWABE, Gerhard: *CSCW Kompendium - Lehr- und Handbuch zum computerunterstützten kooperativen Arbeiten*. Kap. Theorien zur Mediennutzung bei der Gruppenarbeit. Berlin / Heidelberg : Springer, 2001. – ISBN 3540675523
- [SGI 2002] SGI: *OpenGL Multipipe SDK White Paper*. Silicon Graphics, Inc., 2002. – URL <http://techpubs.sgi.com/library/manuals/4000/007-4516-002/pdf/007-4516-002.pdf>. – Letzter Zugriff am 15.06.2008
- [SGI 2008a] SGI: *OpenGL Performer*. Webseite. 2008. – URL <http://www.sgi.com/products/software/performer>. – Letzter Zugriff am 15.06.2008
- [SGI 2008b] SGI: *OpenGL Vizserver*. Webseite. 2008. – URL <http://www.sgi.com/products/software/vizserver>. – Letzter Zugriff am 15.06.2008
- [SGIOrg 2002] SGIORG: *Serious Games Initiative*. Webseite. 2002. – URL <http://seriousgames.org/index2.html>. – Letzter Zugriff am 15.06.2008
- [Skinner 1953] SKINNER, Burrhus F.: *Science and Human Behaviour*. New York : Macmillan, 1953. – ISBN 0029290406
- [Smed und Hakonen 2003] SMED, Jouni ; HAKONEN, Harri: *Towards a Definition of a Computer Game*, Turku Centre for Computer Science, Forschungsbericht Nr.553, September 2003. – URL [staff.cs.utu.fi/~jounsmed/papers/TR553.pdf](http://staff.cs.utu.fi/~jounsmed/papers/TR553.pdf). – Letzter Zugriff am 15.06.2008
- [Stone 2005] STONE, Bob: Serious Gaming. In: *Defense Management Journal* (2005), Dezember, Nr. 32. – URL <http://www.publicservice.co.uk/pdf/dmj/issue31/DMJ31%202241%20Bob%20Stone%20ATL.pdf>. – Letzter Zugriff am 15.06.2008
- [Streitz u. a. 2003] STREITZ, Norbert ; RÖCKER, Carsten ; PRANTE, Thorsten ; STENZEL, Richard ; ALPHEN, Daniel van: Situated Interaction with Ambient Information: Facilitating Awareness and Communication in Ubiquitous Work Environments. In: *10th International Conference on Human-Computer Interaction (HCI International 2003)*, 2003
- [Streitz u. a. 1999] STREITZ, Norbert A. ; GEISSLER, Jörg ; HOLMER, Torsten ; KONOMI, Shin'ichi ; MÜLLER-TOMFELDE, Christian ; REISCHL, Wolfgang ; REXROTH, Petra ; SEITZ, Peter ; STEINMETZ, Ralf: i-LAND: an interactive landscape for creativity and innovation. In: *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM, 1999, S. 120–127. – ISBN 0-201-48559-1
- [Tan u. a. 2006] TAN, Desney S. ; GERGLE, Darren ; SCUPELLI, Peter ; PAUSCH, Randy: Physically large displays improve performance on spatial tasks. In: *ACM Trans. Comput.-Hum. Interact.* 13 (2006), Nr. 1, S. 71–99. – ISSN 1073-0516
- [Tan u. a. 2004] TAN, Desney S. ; GERGLE, Darren ; SCUPELLI, Peter G. ; PAUSCH, Randy: Physically large displays improve path integration in 3D virtual navigation tasks. In: *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM, 2004, S. 439–446. – ISBN 1-58113-702-8

- [Teufel u. a. 1995] TEUFEL, Stephanie ; SAUER, Christian ; MÜHLHERR, Thomas ; BAUKNECHT, Kurt: *Computerunterstützung für die Gruppenarbeit*. Oldenbourg, 1995. – ISBN 3486243705
- [Tyndiuk u. a. 2005] TYNDIUK, F. ; THOMAS, G. ; LESPINET-NAJIB, V. ; SCHLICK, C.: Cognitive comparison of 3D interaction in front of large vs. small displays. In: *VRST '05: Proceedings of the ACM symposium on Virtual reality software and technology*. New York, NY, USA : ACM, 2005, S. 117–123. – ISBN 1-59593-098-1
- [UM3D 2008] UM3D: *University of Michigan 3D Lab*. Webseite. 2008. – URL <http://um3d.dc.umich.edu>. – Letzter Zugriff am 15.06.2008
- [Verbraeck und van Houten 2005] VERBRAECK, Alexander ; HOUTEN, Stijn-Pieter A. van: From simulation to gaming: an object-oriented supply chain training library. In: *WSC '05: Proceedings of the 37th conference on Winter simulation*, Winter Simulation Conference, 2005, S. 2346–2354. – ISBN 0-7803-9519-0
- [Vogel und Balakrishnan 2004] VOGEL, Daniel ; BALAKRISHNAN, Ravin: Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users. In: *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*. New York, NY, USA : ACM, 2004, S. 137–146. – ISBN 1-58113-957-8
- [Vogel und Balakrishnan 2005] VOGEL, Daniel ; BALAKRISHNAN, Ravin: Distant freehand pointing and clicking on very large, high resolution displays. In: *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*. New York, NY, USA : ACM, 2005, S. 33–42. – ISBN 1-59593-271-2
- [Volkswagen-AG 2007] VOLKSWAGEN-AG: *Volkswagen Designers at work*. Webseite. 2007. – URL <http://www.volkswagen.de>. – Letzter Zugriff am 15.06.2008
- [Walser und Graf 2004] WALSER, Marina ; GRAF, Erika: *Virtuelle Teams - Anforderungen, Möglichkeiten und Grenzen*. 2004. – URL [www.fassnachtct.com/informationen/docs/PoV\\_VirtualTeams.pdf](http://www.fassnachtct.com/informationen/docs/PoV_VirtualTeams.pdf). – Letzter Zugriff am 15.06.2008
- [Weiser 1991] WEISER, Mark: The Computer for the 21st Century. In: *Scientific American* 265 (1991), September, Nr. 3, S. 94–104. – URL <http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html>. – Letzter Zugriff am 15.06.2008
- [Wilson 1991] WILSON, Paul: *Computer Supported Work*. Norwell, MA, USA : Kluwer Academic Publishers, 1991. – ISBN 0792314468
- [WireGL 2006] WIREGL: *WireGL - Software for Tiled Rendering*. Webseite. 2006. – URL <http://graphics.stanford.edu/software/wiregl>. – Letzter Zugriff am 15.06.2008
- [Wong u. a. 2007] WONG, Wee L. ; SHEN, Cuihua ; NOCERA, Luciano ; CARRIAZO, Eduardo ; TANG, Fei ; BUGGA, Shiyamvar ; NARAYANAN, Harishkumar ; WANG, Hua ; RITTERFELD, Ute: Serious video game effectiveness. In: *ACE '07: Proceedings of the international conference*

on *Advances in computer entertainment technology*. New York, NY, USA : ACM, 2007, S. 49–55. – ISBN 978-1-59593-640-0

[XDMX 2003] XDMX: *Distributed Multihead X Project: Xdmx Design Document*. Webseite. 2003. – URL <http://dmx.sourceforge.net/dmx.html>. – Letzter Zugriff am 15.06.2008

[van der Zee und Slomp 2005] ZEE, Durk-Jouke van der ; SLOMP, Jannes: Simulation and gaming as a support tool for lean manufacturing systems: a case example from industry. In: *WSC '05: Proceedings of the 37th conference on Winter simulation*, Winter Simulation Conference, 2005, S. 2304–2313. – ISBN 0-7803-9519-0

[Zimmermann und Fimm 1994] ZIMMERMANN, Peter ; FIMM, Bruno: *Test for Attentional Performance*. Webseite. 1994. – URL [http://www.psytest.net/TAP1.7\\_uk.html](http://www.psytest.net/TAP1.7_uk.html). – Letzter Zugriff am 15.06.2008

## **Versicherung über die Selbstständigkeit**

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(5) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 20. Juni 2008

\_\_\_\_\_  
Ort, Datum

\_\_\_\_\_  
Unterschrift