

Studiengang: Softwaretechnik

Prüfer: Prof. Dr. rer. nat. habil. P. Levi
Betreuer: Dr. rer. nat. Michael Schanz
Dipl.-Inf. Viktor Avrutin

begonnen am: 1. Januar 2004
beendet am: 30. Juni 2004

CR-Klassifikation: C.2.0, D.3.4, E.2.0, A.3.4

Diplomarbeit Nr. 2173

**Web-basiertes
Visualisierungsfrontend für den
AnT-Simulator**

Martin S. Kunzelnick

Fakultät Informatik,
Elektrotechnik und
Informationstechnik
Universität Stuttgart
Universitätsstrasse 38
70569 Stuttgart

Inhaltsverzeichnis

Inhaltsverzeichnis	1
Danksagung	7
1 Einleitung	8
1.1 Motivation	8
1.2 Aufgabenstellung	8
1.3 Überblick über den Aufbau	9
1.3.1 Aufbau der Ausarbeitung	9
1.3.2 Website zu dieser Diplomarbeit	10
2 Untersuchung der 3D-Visualisierungsmöglichkeiten	11
2.1 Allgemeines	11
2.2 Kriterien zur Beurteilung	11
2.2.1 Verbreitung	12
2.2.2 Plattformunabhängigkeit der Client-Technologie	12
2.2.3 Performance	12
2.2.4 Freie und plattformunabhängige Editoren und keine sonstigen Kosten	12
2.2.5 Leistungsfähige Sprache	12
2.2.6 Offen gelegte Formate	13
2.2.7 Standardisierte Technologie	13
2.2.8 Installation	13
2.3 Die wichtigsten Plugins	13
2.3.1 Java	14
2.3.2 Java mit Java3D	15
2.3.3 Macromedia Flash	16
2.3.4 Macromedia Shockwave	18
2.3.5 VRML	19
2.3.6 SVG	19

2.4	Sonstige Technologien	21
2.5	Fazit und Entscheidung	21
3	Grundlagen der Umsetzung in Java	23
3.1	Java Applets und Java Web Start	23
3.1.1	Einbindung in HTML	24
3.1.2	Java Web Start-Konfigurationsdatei	24
3.2	Sicherheit	26
3.2.1	Java-Policy-Dateien	26
3.2.2	Signierte JAR-Dateien	27
3.3	Java-Threads	27
3.4	Java-Netzwerkprogrammierung	28
3.4.1	Server	28
3.4.2	Client	29
3.5	Java Swing und Nebenläufigkeit	31
3.6	Der GridBagLayout-Manager	32
3.7	Java und XML mit SAX	32
4	Grundlagen der 3D-Visualisierung	35
4.1	Allgemeines	35
4.2	Mathematische Grundlagen	35
4.2.1	Homogene Koordinaten	35
4.2.2	Allgemeine Transformationen	36
4.2.3	Transformationen von Koordinatensystemen	38
4.2.4	Projektionen	38
4.2.5	Kombinationen von Transformationen	40
4.3	Viewing-Pipeline	40
4.3.1	Modellierung	41
4.3.2	Sicht-Transformation	41
4.3.3	Sichtvolumen-Transformation	43
4.3.4	Projektionstransformation für die perspektivische Transformation	45
4.3.5	Bildschirm-Transformation	46
5	Weitere Grundlagen	47
5.1	Skriptsprachen und Datenbanken	47
5.1.1	PHP	47
5.1.2	ASP.NET und Microsoft SQL Server	47
5.1.3	Weitere Datenbanken	48
5.2	Extreme Programming	48
5.2.1	Kurzüberblick	48

5.2.2	Grundwerte von Extreme Programming	49
6	Projektablauf	51
6.1	Vorprojekt	51
6.2	Vorgehensmodell	54
6.3	Anforderungen	54
6.3.1	Nichtfunktionale Anforderungen	54
6.3.2	Allgemeine funktionale Anforderungen	55
6.3.3	Anforderungen an die GUI	56
6.4	Entwurf	63
6.4.1	Überblick über die AnT-Online-Architektur	63
6.4.2	Überblick über den internen Aufbau von AnT-Online . . .	64
6.4.3	Aufbau der Konfiguration	66
6.5	Ergebnisse	74
6.5.1	Statistiken	74
6.5.2	Dokumente	74
6.5.3	Applet/Java Web Start-Applikation	74
7	Evaluation	79
7.1	Einleitung	79
7.2	Grundlagen der Evaluation	80
7.2.1	Performance der GUI	80
7.2.2	Performance der Kommunikation mit dem Server	80
7.2.3	ISO 9241: Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten, Teil 10: Grundsätze der Dialoggestaltung	80
7.3	Vorbereitung der Evaluation	82
7.3.1	Was soll mit der Evaluation erreicht werden?	82
7.3.2	Wann und wo findet der Test statt?	83
7.3.3	Wie lange dauert eine Testsitzung?	83
7.3.4	Welche Hard- und Software wird benötigt?	83
7.3.5	Von welchem Anfangszustand des Systems soll ausgegangen werden?	83
7.3.6	Wie stark soll das System im Test ausgelastet sein, und wie ist dies zu erreichen?	84
7.3.7	Wer ist der Testleiter?	84
7.3.8	Wer sind die Testbenutzer?	84
7.3.9	Wie viele Testbenutzer werden benötigt?	84
7.3.10	Welche Aufgaben werden gestellt?	85
7.3.11	Welche Hilfsmittel haben die Benutzer, und in welchem Umfang ist Hilfestellung erlaubt?	86

7.3.12	Welche Daten werden gesammelt, und wie werden diese analysiert?	86
7.3.13	Technische Umsetzung	86
7.4	Durchführung	87
7.4.1	Think-Aloud-Methode	88
7.4.2	Aufgabenstellungen	89
7.4.3	Online-Fragebogen	91
7.5	Ergebnisse	92
8	Fazit und Ausblick	93
8.1	Fazit	93
8.2	Ausblick	94
A	Technologieanalyse	96
A.1	Möglichkeiten zur 3D-Visualisierung	96
A.1.1	3D Anywhere [1]	96
A.1.2	3D Groove [2]	96
A.1.3	3DMeNow / XPlayer [3]	97
A.1.4	3DX Player [4]	97
A.1.5	3Space [7]	97
A.1.6	Activate!3D [8]	98
A.1.7	Adobe Atmosphere [9]	98
A.1.8	Anark [10]	98
A.1.9	Anfy3D [11]	99
A.1.10	Axel [12]	99
A.1.11	B3D [13]	99
A.1.12	Blaxxun Virtual Worlds Platform [14]	100
A.1.13	CharActor [15]	100
A.1.14	Cortona [16]	100
A.1.15	Cosmo Player [17]	101
A.1.16	Critical Reach [18]	101
A.1.17	Cult 3D [19]	101
A.1.18	Cybercore Entrance [20]	101
A.1.19	EON [21]	102
A.1.20	eViewer [22]	102
A.1.21	Flash [62]	102
A.1.22	Flatland Rover / 3DML [23]	102
A.1.23	Hypercosm 3D Player [24]	103
A.1.24	iPix [25]	103
A.1.25	Miner3D [27]	104
A.1.26	Objects to See [28]	104

A.1.27	Openworlds [29]	104
A.1.28	plan_b media [30]	104
A.1.29	Pulse 3D [31]	105
A.1.30	Quest3D [32]	105
A.1.31	Quicktime VR [33]	105
A.1.32	RichFX [34]	106
A.1.33	SCOL [35]	106
A.1.34	Shockwave [62]	106
A.1.35	Shout 3D [36]	106
A.1.36	SPX/Open SPX [37]	107
A.1.37	Viewpoint Media Player [38]	107
A.1.38	Virtools [39]	108
A.1.39	Virtue3D [40]	108
A.1.40	VizStream [41]	109
A.1.41	VRML [45]	109
A.1.42	WF-3D [42]	110
A.1.43	WildTangent [43]	110
A.1.44	X3D [45]	111
A.1.45	ZAP [44]	111
A.2	Möglichkeiten für die Datenhaltung	111
A.2.1	MySQL (4.1)	111
A.2.2	PostgreSQL (7.3.4)	112
A.2.3	Hypersonic SQL (1.7.1)	113
A.2.4	Interbase (7.1)	113
B	Klassendiagramme	114
B.1	Die Package-Struktur	114
B.2	Das Package „configuration“	115
B.3	Das Package „datahandling“	116
B.4	Das Package „errorhandling“	117
B.5	Das Package „gui“	118
B.6	Das Package „network“	123
B.7	Das Package „utilities“	124
C	XML-Konfigurationsdatei	125
D	Java Web Start	135
E	Evaluation	136
E.1	Fragebogen	136
E.2	Ergebnisse	142

Index	144
Abbildungsverzeichnis	146
Abkürzungsverzeichnis	148
Glossar	151
Literaturverzeichnis	156

Ich danke Michael Schanz und Viktor Avrutin sowie Prof. Paul Levi, die diese Diplomarbeit möglich gemacht, betreut und unterstützt haben.

Besonders danke ich den Teilnehmern der Evaluation und allen, die mir geholfen haben, Fehler in den Dokumenten zu finden und zu verbessern oder mich anderweitig während der Diplomarbeit unterstützt haben.

Kapitel 1

Einleitung

1.1 Motivation

Zu Beginn dieser Diplomarbeit ließ sich die in der Abteilung Bildverstehen der Fakultät Information an der Uni Stuttgart entwickelte AnT-Software (A simulation and Analysis Tool for Dynamical Systems) nur auf Linux-Systemen einsetzen. Zudem war die Installation der Software weder komfortabel noch einfach und erfordert eine Vielzahl von vorinstallierten Softwarepaketen.

Um neue Benutzer für diese interessante und mächtige Software zu gewinnen, sollte aus dieser Problematik heraus eine einfach zu installierende und gut zu bedienende Software entstehen. Hierbei bietet sich natürlich eine web-basierte Client-Server Lösung an, die idealerweise beim Benutzer nur ein Minimum an installierter Software voraussetzt, wie zum Beispiel einen Browser.

1.2 Aufgabenstellung

Aufgabe der Diplomarbeit ist die Entwicklung eines grafischen, web-basierten Frontends zur Visualisierung von Rechenergebnissen des AnT-Simulators sowie des nötigen Backends auf der Serverseite.

Das Backend hat die Aufgabe, die angemeldeten Benutzer, deren Sessions auf dem Server, die Initialisierungsdateien für den AnT-Simulator sowie die resultierenden Ergebnisdatensätze des AnT-Simulators zu verwalten.

Übergeordnetes Ziel der Diplomarbeit ist es, den AnT-Simulator über das Internet

einer möglichst großen Benutzergruppe zugänglich zu machen. Besonderer Wert wird hierbei auf eine einfache Bedienung und auf möglichst große Plattformunabhängigkeit gelegt. Besonders ist auch darauf zu achten, dass der Benutzer mit möglichst wenig Installationsaufwand zur Benutzung des AnT-Simulators und der Visualisierung der Ergebnisse in der Lage ist. Die Anforderungen an vorhandene Software sollen deswegen einen Webbrowser inklusive Plugin wie Java3D, Flash, Shockwave oder vergleichbares nicht übersteigen.

Der Entwicklung des Visualisierungsfrontends geht eine Evaluation der verfügbaren technischen Möglichkeiten zur zwei- und dreidimensionalen Visualisierung der berechneten Punktwolken voraus. Geprüft und bewertet werden hier unter anderem Technologien wie Java3D, Macromedia Flash, Macromedia Shockwave, VRML, Curl und möglichst viele andere geeignete Plugins.

Wenn eine geeignete Technologie nach dem Evaluationsprozess ausgewählt ist, wird im nächsten Schritt das Backend spezifiziert, entworfen, und es werden die Schnittstellen zum AnT-Simulator und zu den Schnittstellen anderer benötigter Software festgelegt.

Parallel zur Spezifikation und zum Entwurf des Backends wird auch das Visualisierungsfrontend spezifiziert und entworfen. Im letzten Schritt der Diplomarbeit wird das Visualisierungsfrontend getestet und die Gebrauchstauglichkeit evaluiert.

1.3 Überblick über den Aufbau

1.3.1 Aufbau der Ausarbeitung

Diese Ausarbeitung besteht im Wesentlichen aus fünf großen Blöcken:

- **Untersuchung der 3D-Visualisierungsmöglichkeiten im WWW**
Hier finden Sie einen Überblick über die sechs wichtigsten Technologien aus den 45 evaluierten Technologien und Plugins. Neben dieser Vorstellung werden in diesem Kapitel die wichtigsten Kriterien, die der Beurteilung der Plugins zugrunde gelegt wurden, ausführlicher erläutert, und zum Abschluss wird kurz auf die Entscheidung sowie auf deren Gründe eingegangen.
- **Grundlagen der Umsetzung**
In diesem Abschnitt werden die benutzten Grundkonzepte aus den Bereichen Applet- und Java Web Start-Programmierung, Thread-Programmierung,

Netzwerk-Programmierung und GUI-Programmierung mit Swing kurz erläutert. Anschließend wird auf die mathematischen Grundlagen der 3D-Visualisierung eingegangen. Abschließend werden noch verschiedene weitere Grundlagen erläutert.

- **Projektablauf**

Dieses Kapitel beschreibt kurz den Ablauf der Softwareentwicklung im Rahmen dieser Diplomarbeit. Besonders interessant für zukünftige Wartung und Weiterentwicklung der entstandenen Software könnten hierbei die Kapitel zu Spezifikation, Entwurf und Implementierung sein.

- **Evaluation**

Gegenstand dieses Kapitels ist die am Ende der Diplomarbeit durchgeführte Evaluation der entstandenen Software. Besonders wird hier auf die Vorbereitung und die Grundlagen der Evaluation abgehoben. Soweit schon zur Laufzeit der Diplomarbeit vorhanden, werden dann noch die Ergebnisse vorgestellt.

- **Fazit und Ausblick**

Im letzten Kapitel finden Sie ein abschließendes Fazit der Diplomarbeit sowie einen Überblick über mögliche zukünftige Erweiterungen und nötige Verbesserungen der entstandenen Software, die im zeitlichen Rahmen dieser Diplomarbeit nicht mehr umgesetzt werden konnten.

1.3.2 Website zu dieser Diplomarbeit

Da das Ergebnis der Softwareentwicklung im Rahmen dieser Diplomarbeit ein Applet beziehungsweise eine Java Web Start-Applikation ist, die tatsächlich als eine Art Werbung für die AnT-Software eingesetzt werden soll, finden Sie das Applet/die Web Start-Applikation unter anderem auf der AnT-Website (<http://www.ant4669.de>).

Weitere Informationen und Ergebnisse dieser Arbeit finden Sie auf einer eigenen Website (<http://www.kunzelnick.de/da>).

Kapitel 2

Untersuchung der 3D-Visualisierungsmöglichkeiten

2.1 Allgemeines

Dieses Kapitel beschäftigt sich mit der Anfang des Jahres 2004 durchgeführten Untersuchung aktuell verfügbarer Technologien, die sich zur dreidimensionalen Visualisierung von Punktwolken eignen, wie sie im Rahmen dieser Diplomarbeit benötigt wird.

Dieser Teil der Ausarbeitung beschränkt sich allerdings auf eine Bewertung der wichtigsten und vielversprechendsten Technologien.

Eine komplette Liste aller untersuchten Möglichkeiten, deren Stärken, Schwächen und eine Bewertung im Kontext dieser Diplomarbeit finden Sie im Anhang (Kapitel 8.2) und auf der Website (siehe Kapitel 1.3.2) zu dieser Diplomarbeit.

2.2 Kriterien zur Beurteilung

Bei der Erstellung der Software im Rahmen dieser Diplomarbeit wurde von Anfang an besonders großer Wert darauf gelegt, dass auf der Seite des Benutzers keine oder möglichst wenig Software installiert werden muss. Zudem sollte die Lösung sehr einfach über das Internet zu starten und unabhängig von der vom Benutzer verwendeten Plattform und Hardware-Ausstattung sein. Aus diesen Gründen kam nur eine Lösung in Frage, die in jedem Browser, beispielsweise mittels eines passenden Plugins, lauffähig ist.

Die folgenden Abschnitte beschreiben kurz die wichtigsten Kriterien zur Beurteilung verschiedener Plugin-Technologien nach absteigender Priorität.

2.2.1 Verbreitung

Da das entsprechende Plugin vom Benutzer nicht extra wegen der AnT-Online-Visualisierung installiert werden soll, ist das wichtigste Kriterium die Verbreitung des Plugins bei den potenziellen Nutzern.

2.2.2 Plattformunabhängigkeit der Client-Technologie

Neben der Verbreitung spielte bei der Untersuchung eine große Rolle, dass das Plugin für möglichst viele Plattformen verfügbar ist, zumindest aber für Windows und Linux-Umgebungen.

2.2.3 Performance

Da die Software auch auf älteren Rechnern noch eine sinnvolle Mindestzahl von Punkten (ca. 5000) einigermaßen flüssig, das heißt mit Frame-Raten über 15 Frames/s, darstellen können soll, muss das Plugin möglichst effizient implementiert sein und im Idealfall eventuell vorhandene 3D-Beschleunigerhardware nutzen.

2.2.4 Freie und plattformunabhängige Editoren und keine sonstigen Kosten

Da für die Implementierung der Software keine zusätzlichen Mittel verfügbar sind, kommen nur kostenlose Editoren beziehungsweise Entwicklungsumgebungen in Frage, die idealerweise für verschiedene Plattformen verfügbar sind. Aus diesen Gründen kommen auch keine sonstigen Lizenzpflichtigen Technologien in Frage.

2.2.5 Leistungsfähige Sprache

Für die Visualisierung der Punktwolken ist neben der eigentlichen Möglichkeit der Visualisierung auch die Verfügbarkeit programmierbarer Interaktionen mit dem Benutzer über Programmier- oder Skriptsprachen zwingend notwendig.

2.2.6 Offen gelegte Formate

Bei der Bewertung wurde auch Wert darauf gelegt, dass die verwendeten Dateiformate offen gelegt und möglichst nicht proprietär sein sollten.

2.2.7 Standardisierte Technologie

Begrüßenswert wäre die Verwendung standardisierter Technologien, unter anderem weil dann gewährleistet sein sollte, dass gegebenenfalls mehrere Plugin-Implementierungen verfügbar sind, die sich alle gleich oder zumindest vergleichbar verhalten.

2.2.8 Installation

Falls das Plugin vom Benutzer noch nicht installiert wurde, sollte es möglichst einfach sein, dies nachzuholen. Dazu sollte sich das Plugin plattformübergreifender Mittel bedienen, damit keine extra Einarbeitung durch den Benutzer nötig ist.

2.3 Die wichtigsten Plugins

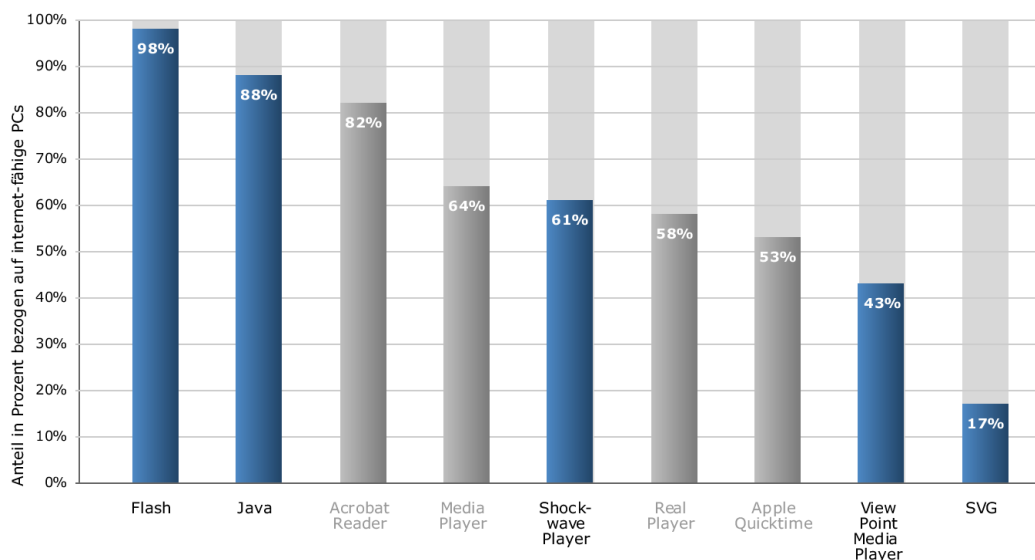


Abbildung 2.1: NPD online survey vom März 2004

Anhand der im vorherigen Abschnitt vorgestellten Kriterien werden im Folgenden die sechs der am besten geeigneten Plugins ausführlicher vorgestellt.

Abbildung 2.1 zeigt die Verbreitung verschiedener Plugins bezogen auf die Gesamtzahl der „internetfähigen PCs“ nach einer Online-Umfrage die im März 2004 veröffentlicht wurde [64]. Alle für diese Arbeit interessanten Plugins sind hervorgehoben, die anderen Plugins sind nur der Vollständigkeit halber aufgeführt.

2.3.1 Java

Vorzüge:

- + Java ist plattformunabhängig, bietet ein großes Maß an Stabilität, Sicherheit und vieles mehr.
- + Java ist eine sehr mächtige Sprache, besonders im Vergleich zu den Skriptsprachen vieler anderer Technologien.
- + Java stellt einen Quasistandard dar und ist damit sehr verbreitet. Es existieren eine Vielzahl von freien Entwicklungsumgebungen, Komponenten, Foren, Literaturquellen, Tutorials et cetera.
- + Da Java eine mächtige Sprache ist und auch über eine Vielzahl von Netzwerk-Komponenten verfügt, ist ein Nachladen und Streamen von Daten selbstverständlich möglich.
- + Java unterstützt eine Vielzahl von Standards, wie zum Beispiel auch XML und damit auch SOAP, beispielsweise um mittels Webservices Firewall-Probleme zu umgehen.
- + Java-Anwendungen können entweder in der Umgebung des Browser eingebettet oder eigenständig als Java Web Start-Applikationen laufen, wobei die Java Web Start-Applikationen auch bequem aus dem Browser heraus gestartet werden können oder später vom Desktop aus.

Nachteile:

- Java an sich verfügt ohne die zusätzlichen Komponenten von Java3D über keine originären 3D-Fähigkeiten. Das bedeutet, dass ein Software-Renderer als zusätzliche Komponente benötigt wird.
- Die Java VM von Microsoft wird eingestellt und ist nicht kompatibel zur Java VM von Sun, was die Verbreitung negativ beeinflussen könnte.

- Eine komplette Installation einer Java RE wird vorausgesetzt. Diese Installation ist üblicherweise deutlich größer und aufwendiger als die meisten Plugins.
- Viele Browser bringen zwar schon eine Java VM mit, unterstützen meist aber nur Java 1.x. Das bedeutet, bei Verwendung neuerer Versionen muss die komplette Java 2 RE von Sun nachinstalliert werden.
- Java hat nicht den Ruf, die schnellste und performanteste Sprache zu sein, stellt aber auf jeden Fall relativ hohe Anforderungen an das System des Benutzers.
- Für eine komfortable, performante und plattformunabhängige GUI-Programmierung bietet sich eigentlich nur die Grafikbibliothek Swing an. Swing ist erst ab der Java-Version 1.2 verfügbar und erst ab der Version 1.4 durch „native Bindungen“ (siehe Glossar) wirklich performant, was die Systemvoraussetzungen heraufsetzt.
- Die Unterstützung von XML-Daten durch den SAX-Parser ist erst ab der Java-Version 1.4 gegeben, wodurch das Applet/die Applikation unter Java-Version 1.3 und kleiner nicht lauffähig ist, ohne dass eine entsprechende JAR-Datei mitgeliefert wird.

2.3.2 Java mit Java3D

Vorzüge:

- + Java3D verfügt über alle Vorzüge von Java wie bereits oben benannt.
- + Java3D stellt eine komplette 3D-Umgebung mit Scenegraph, Beleuchtung, Texturen und weiteren Features zur Verfügung. Natürlich ist hier auch schon die komplette Viewing-Pipeline implementiert.
- + Lösungen in Java3D sind sehr performant, da eventuell vorhandene 3D-Hardware direkt über DirectX und OpenGL ausgenutzt werden kann.

Nachteile:

- Java3D ist im Gegensatz zu Java selbst nicht sehr weit verbreitet. Die nötigen Komponenten müssen getrennt von Java heruntergeladen und installiert werden.
- Java3D ist im Vergleich zu vielen anderen Technologien deutlich schwerer zu erlernen und hat eine relativ steile Lernkurve. Dies gilt besonders dann, wenn man mit den Prinzipien von Scenegraphen nicht genauer vertraut ist.

- Es wird neben der Installation der Java3D-Komponenten natürlich eine komplette Java 2 RE-Installation vorausgesetzt.
- Je nach Plattform ist auch eine korrekte OpenGL- oder DirectX-Installation und Konfiguration erforderlich, ansonsten kann kein Nutzen aus den hardwareseitig vorhandenen Vorteilen von modernen Grafikkarten gezogen werden. Das Rendern der 3D-Inhalte muss dann wieder durch Software emuliert werden, was zu Lasten der Geschwindigkeit und Qualität geht.
- Zusammen ergibt sich dadurch (OpenGL oder DirectX + Java + Java3D) eine komplizierte Installation und Konfiguration im Vergleich zu anderen Plugins.
- Wie Java stellt Java3D relativ hohe Anforderungen an das System des Benutzers, besonders auch an die eingesetzte Grafikhardware, bei der ein entsprechender 3D-Beschleuniger benötigt wird, der aber in modernen Grafikkarten (zum Beispiel GeForce-Karten von Nvidia, Radeon-Karten von ATI, usw.) durchaus vorausgesetzt werden kann.
- Der Scenegraph von Java3D ist nicht für das Anzeigen von Punktwolken, sondern für das Anzeigen von zusammenhängenden Objekten optimiert. Das bedeutet, sowohl bei der Verwaltung als auch bei der Anzeige ergibt sich ein unverhältnismäßig großer Aufwand.

2.3.3 Macromedia Flash

Vorzüge:

- + Macromedia Flash, das momentan als Flash MX 2004 verfügbar ist, gilt in verschiedenen Versionen als das Plugin mit der größten Verbreitung überhaupt. Unabhängige Studien, die von Macromedia genannt werden [62], sprechen in der Summe von einer Verbreitung von 98 Prozent gemessen an allen „internetfähigen PCs“.
- + Macromedia selbst stellt Plugins für die wichtigsten Plattformen (Windows, MacOS, Unix, Linux, ...) zur Verfügung. Unabhängig davon gibt es auch viele OpenSource-Implementierungen auf verschiedensten Plattformen, vornehmlich aber auf Linux.
- + Flash ist nach einer Eingewöhnungsphase relativ einfach zu bedienen. Allenfalls die Kombination aus zeitliniengesteuerter und scriptgesteuerter Programmierung ist etwas ungewöhnlich.

- + Flash MX 2004 verfügt inzwischen mit ActionScript 2.0 über eine mächtige Sprache, die komplett objektorientiert und eventbasiert ist.
- + Das Datenformat, in dem Flash-Projekte auf dem Webserver liegen (.swf-Dateien), ist offen gelegt und nicht mehr durch Patente belastet und damit ein Open Source-Dateiformat [65].
- + Das Nachladen und Streamen von Daten ist über verschiedene Schnittstellen, die ActionScript ab Version 1.0 bietet, sehr einfach möglich.
- + Die Erzeugung von .swf-Dateien über C++ oder Java-Schnittstellen ist durch das OpenSWF-Projekt [65] möglich.
- + Es sind schon verschiedene frei verfügbare und sehr effizient implementierte 3D-Engines vorhanden.
- + Durch den hohen Verbreitungsgrad und die Beliebtheit von Flash sind gute und umfangreiche Dokumentationen, Tutorials, Beispiele, Foren und vieles mehr verfügbar.

Nachteile:

- Je nach Hardwareausstattung des Benutzers ist Flash nicht ausreichend performant für die Darstellung von fünftausend Punkten oder mehr. Erst ab ActionScript 2.0 wurde die Performance deutlich verbessert [62]. Allerdings ist das Flash MX 2004-Plugin Anfang 2004 für manche Plattformen noch nicht verfügbar (Linux-Version erst ab Juni verfügbar).
- Selbst das neue Flash MX 2004 verfügt über keine originären 3D-Fähigkeiten, diese müssen mittels ActionScript nachgebildet werden, aber die Grafik-API ist nicht sehr performant und wenig komfortabel.
- Es sind zur Erstellung der .swf-Dateien eventuell spezielle, nicht komplett kostenfreie Editoren erforderlich, zum Beispiel der Editor von Macromedia, Adobe oder anderen.
- Die .swf-Dateien können zwar auf einem Server erzeugt werden (siehe oben), aber trotzdem ist dies relativ kompliziert, und das Ergebnis sind immer binäre Dateien, die für den Menschen nicht lesbar sind.
- Von Flash sind verschiedene Versionen im Einsatz (Flash 3, Flash 4, 5, MX, MX 2004), die nicht komplett kompatibel zueinander sind. Erst die neuesten Versionen von Flash (ab MX 2004) bringen eine automatische Updatefunktion mit, die bei der schnellen Verbreitung neuerer Versionen hilft. Die neuesten Versionen sind aber immer zuerst für Windows, dann für Apple und erst später für andere Plattformen, wie etwa Linux, verfügbar.

2.3.4 Macromedia Shockwave

Vorzüge:

- + Laut Macromedia [63] ist Shockwave nach Flash und Java das von den untersuchten Plugins das Plugin mit der größten Verbreitung (siehe Abbildung 2.1). Dadurch ist entsprechend viel Literatur vorhanden, genauso wie Tutorials, Foren, Communities und so weiter.
- + Das Shockwave-Plugin ist etwas größer als das Flash-Plugin, aber genauso einfach und komfortabel zu installieren.
- + Shockwave ist deutlich mächtiger und performanter als das Flash-Plugin, auch wenn sich letzteres mit jeder neuen Version an Shockwave annähert.
- + Anders als Flash verfügt Shockwave schon immer über eine mächtige Sprache (Lingo)
- + Wenn man das gemischte Konzept mit Zeitleisten, Frame-Aktionen, Event-Aktionen, Ebenen und so weiter gewohnt ist, bietet Shockwave einen relativ leichten Einstieg und eine flache Lernkurve, unter anderem da Lingo an Java und C++ angelehnt ist.
- + Shockwave verfügt über verschiedene Module (sogenannte Extras), die für die Entwicklung kompletter Multimedia-Applikationen (on- oder offline) genutzt werden können.
- + Mittels des Extras Shockwave 3D sind originäre 3D Fähigkeiten vorhanden.
- + Ähnlich wie in Java3D wird vorhandene 3D-Hardware über OpenGL oder DirectX genutzt, was die 3D-Darstellung sehr performant macht.
- + Das Nachladen und Streamen von Daten ist hier auch sehr einfach möglich.

Nachteile:

- Zum Erstellen von Shockwave-Anwendungen sind spezielle, kostenpflichtige Editoren erforderlich (zum Beispiel Macromedia Director MX), die nur unter Windows oder MacOS verfügbar sind.
- Im Gegensatz zum Flash-Ausgabeformat (.swf) verfügt Shockwave nicht über ein freigegebenes Dateiformat.
- Bis jetzt sind offizielle Player nur für Windows und MacOS verfügbar, aber die Nachfrage, besonders der Linux-Community, wächst ständig.

2.3.5 VRML

Vorzüge:

- + VRML ist seit der Version 1.0 von 1995 ein Quasistandard zur 3D-Visualisierung über das World Wide Web und seit 1997 auch durch ISO-Normen standardisiert.
- + Dadurch, dass VRML eine der ersten Lösungen zur Darstellung von 3D-Inhalten über das World Wide Web war, ist VRML weit verbreitet.
- + Es existiert eine Vielzahl von freien und kostenpflichtigen Plugins für nahezu alle Plattformen.
- + In VRML stehen je nach Version verschiedene Sprachen wie JavaScript, VRMLScript beziehungsweise ECMAScript zur Verfügung.

Nachteile:

- Für VRML existieren so viele verschiedene Plugins, die jeweils sehr unterschiedlich zu bedienen sind, unterschiedliche Features anbieten und so weiter, dass diese Vielfalt verwirrend sein kann und eher abschreckend auf Anwender wirken könnte, die vor dem Problem der Wahl des am besten für sie geeigneten Plugins stehen.
- Nahezu alle Hersteller von Browser-Plugins für VRML-Inhalte halten sich nicht exakt an die VRML-Standards und erweitern ihre Plugins um proprietäre Features. Somit ist es ähnlich wie bei der Beschreibungssprache HTML nahezu unmöglich, mit allen VRML-Plugins eine exakt gleiche Darstellung derselben Inhalte zu erreichen.
- Das Nachladen und Streamen von Daten ist, unter anderem wegen der geringen Mächtigkeit der verfügbaren Skriptsprachen, nur bedingt möglich.
- VRML soll nach Angaben des W3C [80] noch während der Laufzeit dieser Diplomarbeit durch den neueren Standard X3D ersetzt werden.

2.3.6 SVG

Vorzüge:

- + Es sind Plugins für nahezu alle Plattformen verfügbar. Allein das kostenlose Plugin von Adobe existiert in verschiedenen Versionen für Windows, MacOS, Redhat und Solaris.

- + SVG ist ein relativ neuer Standard.
- + SVG basiert auf XML, besteht nur aus reinem ASCII-Text und ist somit sehr einfach dynamisch zu erzeugen.
- + SVG ist eigentlich eine Beschreibungssprache für Grafiken, die auf Vektordarstellung basiert. SVG bietet daher keine originären 3D-Fähigkeiten.
- + Die Programmierung von Interaktionsmöglichkeiten, 3D-Darstellung und anderem ist nur über eine Art Lightweight Javascript steuerbar.
- + SVG ist gut zur Erzeugung von „Data-Driven Graphics“ mit kleineren Interaktionsmöglichkeiten geeignet [66].
- + Es ergibt sich eine relativ flache Lernkurve, da sich SVG auch aus DHTML bekannten Techniken wie zum Beispiel JavaScript (DOM) bedient.
- + SVG unterstützt als Scriptsprache SMIL (Synchronized Multimedia Integration Language) [67].
- + Neben kostenlosen Plugins existiert auch eine Vielzahl von freien Viewern, Generatoren und Editoren für SVG.

Nachteile:

- Bei SVG ist wegen der eingeschränkten Mächtigkeit der verfügbaren Sprachen kein Streaming oder eine Interaktion mit dem Webserver oder einem anderen Server möglich.
- Für die XML-Inhalte der SVG-Dateien ist, anders als zum Beispiel bei VRML, keine Kompression verfügbar, was das zu übertragende Datenvolumen sehr groß machen kann.
- Zumindest durch die wenig effiziente Implementierung bei den bisherigen Plugins und bedingt durch die wenig mächtigen Scriptsprachen bieten die verfügbaren SVG-Lösungen sehr wenig Performance, was aber auch daran liegt, dass SVG weniger zur Erstellung komplexer Anwendungen gedacht ist.
- Die Installation des verbreiteten Adobe-Plugins ist etwas ungewöhnlich, zumindest aber anders als bei den meisten Browser-Plugins üblich.
- Obwohl SVG bei Erscheinen des neuen Formats oft als „Flash-Killer“ gepriesen wurde, ist und bleibt die Verbreitung, der Einsatz und die Weiterentwicklung stockend. Nach Schätzungen aus verschiedenen Quellen im World Wide Web liegt die aktuelle Verbreitung bei weniger als 10-17 Prozent und steigt weiterhin nicht nennenswert.

2.4 Sonstige Technologien

Im Anhang finden Sie, genauso wie auf der Website zu dieser Diplomarbeit, eine Kurzübersicht über insgesamt 45 verschiedene Plugins, die sich zur Darstellung von 3D-Inhalten über das World Wide Web eignen.

Abgesehen davon, dass die meisten untersuchten Plugins eine sehr geringe Verbreitung aufweisen, findet sich meist noch eine Vielzahl von Gründen, warum sich deren Einsatz weniger anbietet.

Besonders fällt dabei auf, dass eine Vielzahl von Plugin-Herstellern, anscheinend bedingt durch die Verbreitung des Internet-Explorers von über 75 Prozent (zwischen 75 Prozent und 96 Prozent, je nach Quelle, wie zum Beispiel [68], [69], [70], [71] und andere), dazu übergegangen sind, Plugins nur noch als ActiveX-Komponenten anzubieten.

Dadurch sind die Plugins natürlich nur noch für Windows-Plattformen in Kombination mit dem Internet-Explorer verfügbar. In vielen Fällen wird sogar explizit von den Herstellern auf deren Websites eingeräumt, dass sich eine plattformunabhängige Implementierung ihrer Plugins nicht mehr lohnt, besonders da von Microsoft schon oft angekündigt wurde, Java nicht mehr zu unterstützen.

2.5 Fazit und Entscheidung

	Java	Java3D	Flash	Shockwave	VRML	SVG
weit verbreitet	✓	✗	✓	✓	✗	✗
plattformunabhängig	✓	✓	✓	✗	✓	✓
performant	✓	✓	✗	✓	✗	✗
freie Editoren	✓	✓	✓	✗	✓	✓
mächtige Sprache	✓	✓	✓	✓	✗	✗
offengelegte Formate	✓	✓	✓	✗	✓	✓
standardisiert	✓	✓	✗	✗	✓	✓
einfache Installation	✓	✗	✓	✓	✗	✗

Abbildung 2.2: Resultate der Evaluation

In Abbildung 2.2 sind zusammenfassend die wichtigsten Technologien gegen die wichtigsten Kriterien zur Auswahl einer 3D-Visualisierungsplattform aufgetragen.

Wie hier deutlich zu sehen ist, erfüllt Java alle wesentlichen Kriterien. Aus diesem Grund und wegen verschiedener weiterer oben genannter Vorteile von Java ist das Fazit der vorgenommenen Untersuchung, dass die am besten geeignete Technologie für die Implementierung der AnT-Online-Visualisierung Java ist.

Nach einer Präsentation aller wesentlichen Untersuchungsergebnisse der verfügbaren Möglichkeiten zur Visualisierung von 3D-Daten über das World Wide Web wurde eingehend über die Vor- und Nachteile der wichtigsten Technologien von verschiedenen Mitarbeitern des AnT-Projekts diskutiert.

Entsprechend der Empfehlung wurde schließlich Java ab der Version 1.2 als Technologie für die Implementierung der AnT-Online-Visualisierung ausgewählt.

Eine Beschränkung auf Java ab Version 1.2 muss vorgenommen werden, da die Benutzungsoberfläche mit Hilfe von Java Swing implementiert werden soll, das erst ab dieser Version verfügbar ist.

Kapitel 3

Grundlagen der Umsetzung in Java

3.1 Java Applets und Java Web Start

Da es relativ einfach ist, eine Java-Anwendung so zu schreiben, dass sie sowohl als Applet innerhalb eines Browsers als auch als selbstständige Java Web Start-Applikation ausführbar ist, wurde dies in dieser Arbeit von Anfang an vorgesehen. Das Ausführen der Anwendung als Java Web Start-Applikation bringt gegenüber der Ausführung als Applet einige Vorteile mit sich:

- Die Applikation muss nicht bei jedem Start neu vom Webserver geladen werden, da Java Web Start die Applikationen zwischenspeichert und beim Start nur prüft, ob eine neuere Version vorhanden ist. Natürlich ist somit auch eine Liste von kürzlich ausgeführten Applikationen in Java Web Start verfügbar.
- Es können automatisch Verknüpfungen zu Applikationen auf dem Desktop angelegt werden, sodass das Starten so komfortabel wie das einer lokal installierten Applikation ist.
- Theoretisch könnten so auch Anwendungen ohne aktive Verbindung zum Internet ausgeführt werden. Dies funktioniert im Fall dieser Arbeit natürlich nicht, da die Applikation auf die Daten des Servers angewiesen ist.
- Auch ohne dass die Anwendung signiert sein muss, kann sie über das Java Network Launching Protocol Zugriff auf verschiedene Ressourcen wie lokale Dateien, Netzwerkverbindungen und so weiter ersuchen. Der Anwender kann über entsprechende Dialoge entscheiden, ob er diese Zugriffe aus der Sandbox heraus zulässt oder nicht (siehe auch Kapitel 3.2).

- Der Speicherverbrauch ist unter Umständen deutlich geringer, da kein zusätzliches Programm wie der Browser benötigt wird.

3.1.1 Einbindung in HTML

Um die Anwendung als Applet verfügbar zu machen, muss sie in eine JAR-Datei kompiliert und über eine HTML-Seite für Browser verfügbar gemacht werden. Hierzu muss in einer HTML-Datei lediglich eine Applet-Umgebung vergleichbar der Folgenden erzeugt werden:

```
<applet
    code="<Klassenname>"
    archive="<JAR-Datei>"
    name="<Name>"
    width="<Breite>"
    height="<Höhe>"
    id="<ID>"
>
  <param
    name="configURI"
    value="http://localhost/config/config.xml" />
</applet>
```

Hier muss im Applet-Tag nur angegeben werden, wie der Klassenname des Applets lautet und wie der Name der JAR-Datei und gegebenenfalls ihr relativer Pfad ist. Zusätzlich kann hier auch noch die Breite und Höhe angegeben werden, die das Applet auf der angezeigten HTML-Seite einnimmt. Name und ID sind optional und werden nur benötigt, um das Applet über eine Scriptsprache wie JavaScript anzusprechen.

3.1.2 Java Web Start-Konfigurationsdatei

Etwas komplizierter als die Einbindung eines Applets ist die Erstellung einer Konfigurationsdatei, um die Anwendung als Java Web Start-Applikation starten zu können.

Die Konfigurationsdatei ist eine XML-Datei nach folgendem Schema:

```
<jnlp codebase="http://www.kunzelnick.de/da/antwebvis/"
      href="http://www.kunzelnick.de/da/antwebvis/
        webstart_en.jnlp">
<information>
```

```

<title>AnT online visualisation</title>
<vendor>University of Stuttgart / Martin Kunzelnick</
  vendor>
<homepage href="http://www.kunzelnick.de/da/" />
<description>Visualisation for AnT-files</description>
</information>
<security>
  <j2ee-application-client-permissions />
</security>
<resources>
  <j2se version="1.2+" />
  <jar href="antvis.jar" />
</resources>
<application-desc main-class="AntWebVis">
  <param name="configURI" value="http://www.kunzelnick.de
    /da/antwebvis/config/config_en.xml" />
</application-desc>
</jnlp>

```

Besonders zu beachten ist dabei, dass die Konfigurationsdatei, bei einer Verlinkung mit dem richtigen Content-Type zurückgeliefert wird. Die Konfigurationsdatei muss dabei nicht auf demselben Webserver liegen, wie die JAR-Datei.

Dies ist entweder über eine Scriptsprache wie etwa PHP

```
<?header("Content-Type:_application/x-java-jnlp-file");?>
```

oder ASP.NET

```
<%@ Page ContentType="application/x-java-jnlp-file"%>
```

möglich, über eine entsprechende Zeile in einer .htaccess-Datei

```
Addtype application/x-java-jnlp-file jnlp
```

oder natürlich über die Konfiguration des jeweiligen Webservers selbst.

Gestartet werden kann die Java Web Start-Applikation dann entweder durch den Aufruf des oben beschriebenen Links auf die entsprechende jnlp-Datei oder direkt, indem über eine Konsole „javaws <URI>“ aufgerufen wird, wobei die URI der Pfad zur jnlp-Datei ist.

3.2 Sicherheit

Da Applets und auch Java Web Start-Applikationen normalerweise in einer Sandbox ausgeführt werden, sind sie einigen Restriktionen unterworfen. Sie haben dadurch weder die Möglichkeit, auf lokale Datenträger zuzugreifen, noch können sie Netzwerkverbindungen zu anderen Servern herstellen. Es können lediglich neue Verbindungen zum „Home-Server“ aufgebaut werden, also zu dem Server, von dem das Applet oder die Java Web Start-Applikation heruntergeladen wurde.

Um diese Restriktion zu umgehen, bieten sich entweder die Verwendung von Policy-Dateien oder das Signieren von JAR-Dateien an.

3.2.1 Java-Policy-Dateien

Java-Policy-Dateien heißen „java.policy“ und müssen im Home-Verzeichnis eines Benutzers liegen (unter Linux z.B. „/home/kunzelmn“ und unter Windows „C:\Dokumente und Einstellungen\Martin\“). Java-Policy-Dateien haben den Vorteil, dass sie auf dem Computer des Benutzers liegen und der Benutzer damit sicher sein kann, dass ein Applet nur Zugriff auf ganz bestimmte Ressourcen bekommt. Das folgende Beispiel zeigt eine Java-Policy-Datei, die nur den Netzwerk-Zugriff auf zwei Server zulässt.

```
grant {
    permission
        java.net.SocketPermission
            "test.kunzelnick.de:4669",
            "accept,_connect,_listen,_resolve";
    permission
        java.net.SocketPermission
            "www.informatik.uni-stuttgart.de:80",
            "accept,_connect,_listen,_resolve";
};
```

Was einerseits ein Vorteil ist, nämlich dass der Benutzer die Datei selbst anlegen muss und damit volle Kontrolle hat, ist natürlich dann auch ein Nachteil, wenn man es dem Benutzer besonders einfach machen will, ein Applet zu benutzen, da dieser erst die Datei anlegen muss. Dies ist unter Windows schon nicht ganz trivial, da Dateien eigentlich nicht mit einem Punkt beginnen dürfen. In jedem Fall sind Policy-Dateien eine weitere mögliche Fehlerquelle. Um diese zu umgehen, bieten sich signierte JAR-Dateien an.

3.2.2 Signierte JAR-Dateien

Anders als bei Policy-Dateien hat das Signieren von JAR-Dateien den Vorteil, dass der Benutzer lediglich zustimmen muss, dass dem Applet/der Applikation weitergehende Zugriffsrechte eingeräumt werden.

Zum Signieren von JAR-Dateien müssen folgende Schritte unternommen werden:

1. Erzeugen eines Schlüsselpaares (mittels „keytool“)
2. Das Schlüsselpaar muss von einer CA zertifiziert werden.
3. Die JAR-Datei wird signiert (mittels „jarsigner“).

Wenn das Schlüsselpaar von einer CA zertifiziert wurde, der der Benutzer vertraut, entfällt auch die oben erwähnte Sicherheitsabfrage, da dann demjenigen, der das Applet signiert hat, implizit vertraut wird (Web of Trust).

3.3 Java-Threads

In Java ist es sehr einfach, nebenläufige Prozesse zu erzeugen. Neben Ansätzen mit verschiedenen Timer-Klassen kann man im Wesentlichen zwei Vorgehensweisen unterscheiden [79]:

- **Ableiten von der Klasse Thread und Überschreiben der run()-Methode**
Beispiel für eine Klasse, die als eigener Thread laufen soll:

```
public class MeinThread extends Thread {  
  
    public MeinThread() {  
        // beliebige Aktionen im Konstruktor  
    }  
  
    public void run() {  
        //eigentliche Aufgabe der Klasse, z.B. lang-  
        //wierige Berechnungen, Datentransfers, ...  
    }  
}
```

In einer anderen Klasse können damit beliebig viele Instanzen dieser Klasse erzeugt werden, die dann parallel ausgeführt werden:

```
MeinThread thread1 = new MeinThread();  
MeinThread thread2 = new MeinThread();
```

```
thread1.start();
thread2.start();
```

- **Implementieren des Interfaces Runnable**

Beispiel für eine Klasse, die als eigener Thread laufen soll:

```
public class MeinRunnable implements Runnable {

    public MeinRunnable() {
        // beliebige Aktionen im Konstruktor
    }

    public void run() {
        //eigentliche Aufgabe der Klasse, z.B. lang-
        //wierige Berechnungen, Datentransfers, ...
    }
}
```

Wie bei Klassen, die von Thread abgeleitet sind, können dann mehrere Klassen, die Runnable implementieren, parallel ausgeführt werden:

```
MeinRunnable runnable1 = new MeinRunnable();
MeinRunnable runnable2 = new MeinRunnable();
runnable1.run();
runnable2.run();
```

3.4 Java-Netzwerkprogrammierung

3.4.1 Server

Der Server stellt einen Port (zum Beispiel 4444) zur Verfügung, auf dem er auf eingehende Verbindungen wartet. Für jede eingehende Verbindung wird dabei ein neuer Thread gestartet, sodass es theoretisch beliebig viele Socket-Verbindungen geben kann.

Das folgende Beispiel zeigt den Versuch, einen neuen ServerSocket an einen Port zu binden. Schlägt dies fehl, wird der Server beendet.

Anschließend tritt der Server in eine Endlosschleife ein, in der er auf eingehende Verbindungen wartet. Für jede neue Socket-Verbindung wird eine neue Instanz der Klasse „ConnectionHandler“ erzeugt. Die Klasse „ConnectionHandler“ leitet dabei von der Klasse Thread ab (public class ConnectionHandler extends Thread)

und bekommt von der Server-Klasse den entsprechenden Socket über den Konstruktor übergeben.

```
public class Server {

    public static void main(String[] args) {
        int port = 4444;
        ServerSocket welcomeSocket = null;
        try {
            welcomeSocket = new ServerSocket(port);
        }
        catch (Exception e) {
            System.out.println("Server_could_not_get_socket");
            System.exit(-1);
        }
        while (true) {
            try {
                Socket connectionSocket = welcomeSocket.accept();
                ConnectionHandler newHandler =
                    new ConnectionHandler(connectionSocket);
                newHandler.run();
            }
            catch (Exception e) {
                System.err.println("Server-error:" + e);
                System.exit(-1);
            }
        }
    }
}
```

3.4.2 Client

Im Rahmen dieser Arbeit erstellt der Client über eine Art Factory für alle zu übertragenden Daten eine neue Verbindung über einen neuen Socket, aber natürlich auf demselben Port. So ist es dem Client möglich, ohne das Modellieren von Zuständen Daten mit dem Server auszutauschen. Dieser Austausch verläuft dabei nebenläufig, sodass zum Beispiel beliebig viele AnT-Dateien, Bilder, Beschreibungen und so weiter übertragen werden können.

Das folgende Beispiel zeigt, wie über eine Methode mit Servername und Port als Parameter eine Verbindung mit einem Server aufgebaut werden kann. Dabei sind

„outToServer“ und „inFromServer“ Streams, über welche nach erfolgreicher Ab-
arbeitung der Methode Daten zum Server gesendet beziehungsweise vom Server
empfangen werden können.

```
private DataOutputStream outToServer = null;
private InputStream inFromServer = null;

private void initConnection(String server, int port) {
    try {
        System.out.println("connecting, please wait!\r\n");
        clientSocket = new Socket(server, port);
        outToServer =
            new DataOutputStream(
                clientSocket.getOutputStream ()
            );
        inFromServer = clientSocket.getInputStream ();
    } catch (Exception e) {
        System.out.println("connection-error: "+e+"\r\n");
    }
}
```

Für die jeweiligen Bedürfnisse der Datenübertragung müssen dann nur noch die
geöffneten Streams umgeleitet werden, zum Beispiel durch einen BufferedInput-
Stream, BufferedOutputStream, BufferedReader, BufferedWriter, ...

Textkommandos können aber auch einfach so an den Server übertragen werden:

```
String cmd = "help_\n\r";
outToServer.writeBytes(cmd);
```

Die Reaktion kann so entgegengenommen werden:

```
int i = inFromServer.read();
String s = String.valueOf((char)i);
System.out.print(s);
```

Diese Methode, einzelne Zeichen zu lesen, ist allerdings sehr ineffizient. Es bietet
sich an - zum Beispiel mittels eines BufferedReader - immer ganze Zeilen zu lesen,
oder Byte-Arrays:

```
byte[] buffer = new byte[1024];
inFromServer.read(buffer);
```

3.5 Java Swing und Nebenläufigkeit

Anders als das AWT ist Swing ausdrücklich nicht threadsicher. Das große Problem, das sich schon während des Vorprojektes zeigte, ist, dass einmal erzeugte GUI-Komponenten nur noch von einem Thread geändert werden können, dem sogenannten Event Dispatcher Thread. Dies hat unter anderem zur Folge, dass Datentransfers, die in eigenen Threads ablaufen, keine Möglichkeit haben, zum Beispiel Anzeigen in JProgressBars, JPanels und so weiter zu aktualisieren. Besonders ungünstig ist dabei auch, dass die eigentliche Nebenläufigkeit durch den Event Dispatcher Thread serialisiert wird. Damit ist die parallele Übertragung oder Bearbeitung von mehreren Datenströmen nicht möglich.

Zur Lösung dieses Problems stellt Sun verschiedene Möglichkeiten vor. Im Rahmen dieser Arbeit wird die Variante mit der sogenannten „SwingWorker“-Klasse verwendet. Diese Klasse ist nicht Bestandteil des Java-SDK, und die Klassendatei muss somit in das eigene Projekt integriert werden.

Um Nebenläufigkeit unter Swing zu implementieren, muss dabei folgendes unternommen werden:

- Die eigene Klasse muss von der SwingWorker-Klasse abgeleitet sein und sie muss mindestens die construct()-Methode implementieren:

```
public class MeineKlasse extends SwingWorker {  
  
    public Object construct() {  
        // Teil, der parallel zur  
        // GUI ausgeführt werden soll  
    }  
}
```

- Dann kann die eigene Klasse gestartet werden, und die Ausführung findet tatsächlich im Hintergrund statt, während die GUI noch weiter reagiert:

```
public void actionPerformed(ActionEvent e) {  
    ...  
    MeineKlasse meineKlasse =  
        new MeineKlasse();  
    meineKlasse.start();  
    ...  
}
```

Weitere Details und Download der SwingWorker-Klasse siehe [81].

3.6 Der GridBagLayout-Manager

Um eine möglichst große Flexibilität beim Layout der GUI-Komponenten zu haben, wurde in dieser Arbeit der GridBagLayout-Manager gewählt. Der GridBagLayout-Manager gilt als der komplexeste, aber auch als der mächtigste Layoutmanager von Swing. Er basiert auf einem flexiblen Gitter von Spalten und Zeilen, deren Anzahl durch die hinzugefügten GUI-Komponenten bestimmt wird. Für alle Zellen innerhalb dieser Tabelle kann unter einer Vielzahl von Ausrichtungsvarianten gewählt werden. Zusätzlich kann auch für die einzelnen Zellen eine Gewichtung in Form eines float-Wertes angegeben werden, die im Zusammenspiel mit den anderen Gewichtungen die Größe der Zelle bestimmt. Für weitere Details und Informationen zum GridBagLayout-Manager siehe [82].

3.7 Java und XML mit SAX

XML findet in dieser Arbeit Anwendung für die Konfiguration des Applets/der Java Web Start-Applikation. Java bietet zum Einlesen und Verarbeiten von XML-Daten eine Vielzahl von Möglichkeiten. Im Rahmen dieser Diplomarbeit wurde die serielle Verarbeitung mittels SAX gewählt.

Java bietet, wie viele andere Sprachen, mit SAX eine schnelle, extrem einfache und speicherschonende Möglichkeit an, XML-Daten zu nutzen. Hierzu wird eine neue Klasse erstellt, die von der Klasse „org.xml.sax.helpers.DefaultHandler“ ableitet. In der neu erstellten Klasse müssen dann nur noch folgende Methoden implementiert werden, um die Konfigurationsdaten aus einer XML-Datei in eine interne Repräsentation zu übertragen:

```
public void startDocument()  
    throws SAXException  
{  
    ...  
}  
  
public void endDocument()  
    throws SAXException  
{  
    ...  
}  
  
public void startElement
```

```

        (String namespaceURI,
         String sName,
         String qName,
         Attributes attrs)
        throws SAXException
    {
        ...
    }

    public void endElement
        (String namespaceURI,
         String sName,
         String qName)
        throws SAXException
    {
        ...
    }

```

Die vier genannten Callback-Methoden sind nur ein Ausschnitt der SAX-Implementierung von Java (weitere Informationen siehe auch [77]).

Neben dem Überschreiben der vererbten Methoden muss auch eine Instanz des SAX-Parsers erzeugt werden und die eigene Klasse als Handler eingetragen werden:

```

String path = "datei.xml";
DefaultHandler handler = this;
SAXParserFactory factory =
    SAXParserFactory.newInstance();
try {
    SAXParser saxParser = factory.newSAXParser();
    saxParser.parse( path, handler );
} catch (Exception e) {
    System.out.println(
        "error_reading_xml-config-file:"
        + e.fillInStackTrace()
    );
    System.exit(-1);
}

```

Die SAX Implementierung hat natürlich auch verschiedene Nachteile. Unter anderem steht durch die serielle Natur von SAX immer nur ein Teil der XML-Daten im Speicher zur Verfügung, was zwar zu Problemen führen kann, die aber in der Anwendung im Rahmen dieser Arbeit keine Rolle spielen. Hier muss nicht nach

verschiedenen Namen, Werten oder Pfaden gesucht werden, sondern lediglich die Syntax überprüft und Wert für Wert in eine interne Speicherrepräsentation eingelesen werden.

Kapitel 4

Grundlagen der 3D-Visualisierung

4.1 Allgemeines

Zur Visualisierung gibt es verschiedene allgemein anerkannte Modelle. Bevor hier etwas näher auf eines dieser Modelle eingegangen wird, müssen noch kurz einige mathematische Grundlagen erläutert werden.

4.2 Mathematische Grundlagen

4.2.1 Homogene Koordinaten

Da verschiedene Transformationen, wie zum Beispiel die Translation im dreidimensionalen Fall, nicht durchgängig als Matrixmultiplikation angewendet werden können, müssen die homogenen Koordinaten eingeführt werden. Das bedeutet, dass die hier relevanten dreidimensionalen Punktkoordinaten um eine w -Koordinate ergänzt werden.

Der Punkt P mit $P = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ hat die homogenen Koordinaten $[x_h, y_h, z_h, w]$

mit $w \neq 0$ und $x_h = x \cdot w, y_h = y \cdot w, z_h = z \cdot w$. Die w -Koordinate kann also auch als eine Art Skalierungsfaktor betrachtet werden. In der Computergrafik wird aber einfachheitshalber $w = 1$ gewählt. Besonders beachtet werden müssen hier nichtlineare Transformationen (siehe ??).

4.2.2 Allgemeine Transformationen

Durch die Nutzung homogener Koordinaten müssen die grundlegenden Matrixoperationen etwas angepasst werden. Dafür kann aber immer einheitlich mit Matrixmultiplikationen gerechnet werden. Die Translation stellt dann ebenfalls keinen Sonderfall mehr da.

Auf die Transformationen „Scherung“ und „Spiegelung“ wird in diesem Kapitel nicht eingegangen.

Rotation

Bei der Rotation muss bei der Erstellung der Rotationsmatrix unterschieden werden, um welche Achse rotiert werden soll. Um jede beliebige Rotation darzustellen, können Translationen und die einzelnen Rotationen für die x-, y- und z-Rotation hintereinander ausgeführt werden. Auf die etwas kompliziertere Rotation um beliebige Punkte und Achsen wird hier nicht eingegangen.

Rotationsmatrix für die Rotation um die x-Achse

$$R_x(\delta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\delta) & -\sin(\delta) & 0 \\ 0 & \sin(\delta) & \cos(\delta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotationsmatrix für die Rotation um die y-Achse

$$R_y(\delta) = \begin{pmatrix} \cos(\delta) & 0 & \sin(\delta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\delta) & 0 & \cos(\delta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotationsmatrix für die Rotation um die z-Achse

$$R_z(\delta) = \begin{pmatrix} \cos(\delta) & -\sin(\delta) & 0 & 0 \\ \sin(\delta) & \cos(\delta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Skalierung

Für die drei Skalierungsfaktoren s_x , s_y und s_z mit $s_x \neq 0$, $s_y \neq 0$ und $s_z \neq 0$ ergibt sich die Skalierungsmatrix

$$S(s_x, s_y, s_z) = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Für den Fall, dass die Skalierung uniform sein soll (Seitenverhältnisse bleiben erhalten), muss lediglich $s_x = s_y = s_z$ gelten. Ist einer der Skalierungsfaktoren kleiner als 1, wird entlang der betreffenden Achse eine Stauchung durchgeführt. Ist der Skalierungsfaktor größer als 1, wird eine entsprechende Streckung durchgeführt.

Der Fixpunkt bei der oben vorgestellten Skalierungsmatrix liegt dabei immer im Ursprung. Soll ein anderer Punkt als Fixpunkt dienen, muss vor und nach der Skalierung eine entsprechende Translation durchgeführt werden.

Hintereinander ausgeführte Skalierungen sind multiplikativ.

Translation

Speziell für die Translation, die normalerweise als Addition eines Translationsvektors $\vec{t} = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}$ bezüglich eines Punktes $P = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ durchgeführt wird, ergibt sich mit der Homogenisierung

$$P' = \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = T(t_x, t_y, t_z) \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

für die Translation folgende Translationsmatrix

$$T(t_x, t_y, t_z) = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Hintereinander ausgeführte Translationen sind additiv.

4.2.3 Transformationen von Koordinatensystemen

Einzelne Objekte besitzen in der Computergrafik üblicherweise ein eigenes, lokales Koordinatensystem. Die Transformation eines lokalen Koordinatensystems B in ein globales oder Weltkoordinatensystem A ist dabei nichts anderes als die Verschiebung des Ursprungs von B in den Ursprung von A , gefolgt von den entsprechenden Rotationen, sodass die Koordinatenachsen von B auf denen von A zu liegen kommen. Vorausgesetzt wird hier allerdings noch, dass es sich jeweils um kartesische Koordinatensysteme handelt, und die Vektoren, welche die Koordinatensysteme aufspannen, auf die Länge 1 normiert sind.

Damit ergibt sich folgende Matrix für die Transformation des lokalen Koordinatensystems B in das globale Koordinatensystem A :

$$T_{BA} = \begin{pmatrix} B_{x_1} & B_{y_1} & B_{z_1} & T_x \\ B_{x_2} & B_{y_2} & B_{z_2} & T_y \\ B_{x_3} & B_{y_3} & B_{z_3} & T_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

4.2.4 Projektionen

Für die Abbildung der Weltkoordinaten in eine Bildebene sind verschiedene Projektionen denkbar. Abbildung 4.1 zeigt eine Übersicht über mögliche Projektionsarten und deren Zusammenhang.

Die einfachste Projektion ist die orthogonale Parallelprojektion $P_{orthogonal}$ in die xy -Ebene mit

$$P_{orthogonal} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Diese Art der Projektion ist zwar sehr einfach, vermittelt aber keinen räumlichen Eindruck. Aus diesem Grund wurde für die 3D-Visualisierung in dieser Arbeit eine Art der perspektivischen Projektion gewählt. Eine einfache Projektionsmatrix für die perspektivische Projektion auf die xy -Ebene ist

$$P_{pers}(a) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/a & 1 \end{pmatrix}$$

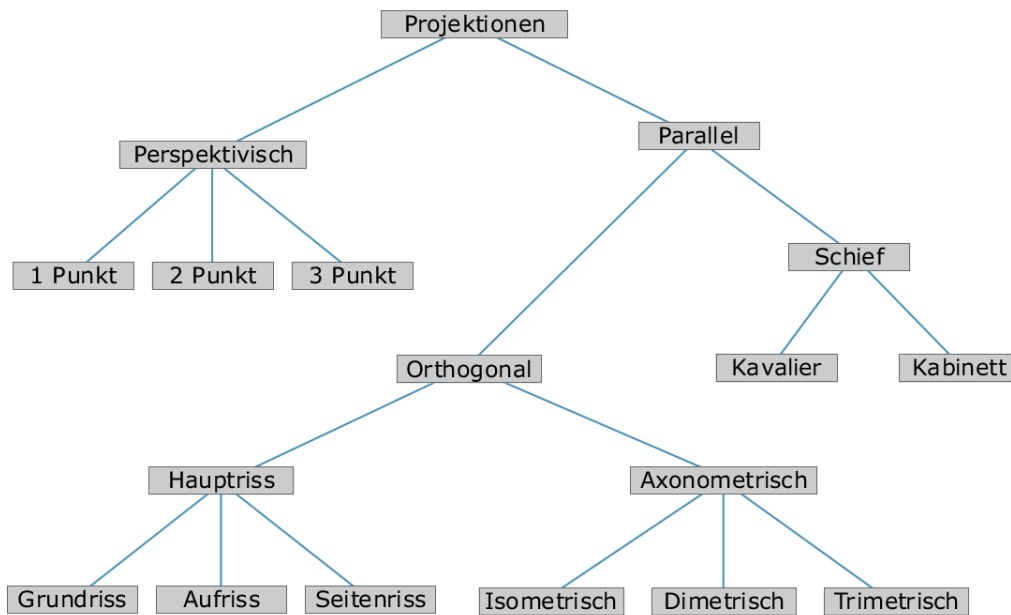


Abbildung 4.1: Übersicht der Projektionsarten

wobei angenommen wird, dass sich das Projektionszentrum auf der negativen z -Achse im Punkt $Z = (0, 0, -a)$ befindet. a ist also der Abstand des Projektionszentrums von der xy -Ebene. Zu beachten ist bei dieser Art der Projektion, dass es sich dabei, anders als bei allen anderen beschriebenen Transformationen, nicht um eine affine Transformation handelt, was bedeutet, dass Streckenverhältnisse und Winkel nicht erhalten bleiben.

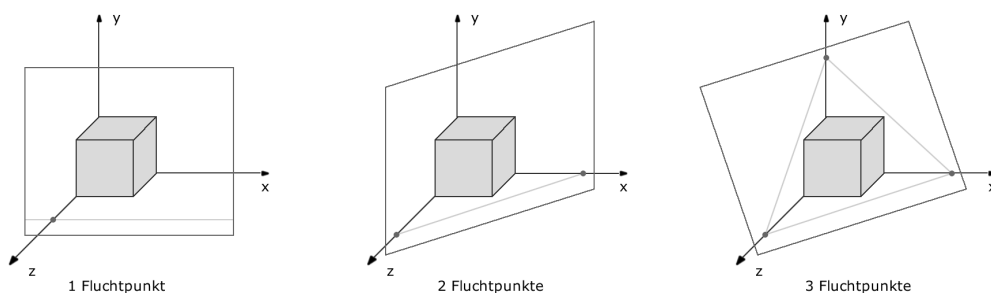


Abbildung 4.2: Anzahl der Fluchtpunkte bei der perspektivischen Projektion

P_{pers} beschreibt eine perspektivische Projektion mit einem Fluchtpunkt. Die Anzahl der Fluchtpunkte ergibt sich daraus, wie viele der 3 Koordinatenachsen von

der Projektionsebene geschnitten werden, wie Abbildung 4.2 zeigt.

4.2.5 Kombinationen von Transformationen

Ein weiterer Vorteil der Homogenisierung ist, dass beliebige Transformationen, die nacheinander ausgeführt werden sollen, zu einer einzigen Transformationsmatrix multipliziert werden können.

Sollen zum Beispiel die drei beliebigen Transformationen T_1 , T_2 und T_3 hintereinander auf einen Punkt P angewendet werden, ergibt sich der endgültige Punkt P''' durch:

$$\begin{aligned}P' &= T_1 \cdot P \\P'' &= T_2 \cdot P' \\P''' &= T_3 \cdot P''\end{aligned}$$

Durch Multiplikation aller Transformationsmatrizen in umgekehrter Reihenfolge ausgedrückt:

$$P''' = T_3 \cdot T_2 \cdot T_1 \cdot P$$

4.3 Viewing-Pipeline

Nachdem aus den vorangegangenen Abschnitten jetzt die wesentlichen mathematischen Grundlagen bekannt sind, wird in diesem Abschnitt das Verfahren beschrieben, mit dem aus den dreidimensionalen Ausgangskordinaten die zweidimensionalen Koordinaten auf dem Bildschirm hervorgehen.



Abbildung 4.3: Weg von 3D-Welt- zu 2D-Bildkoordinaten

Dieses Verfahren, das auch als Viewing-, Render-, oder 3D-Pipeline bezeichnet wird, definiert eine Reihe von Arbeitsschritten. Die Bezeichnung „Pipeline“ soll dabei verdeutlichen, dass die Rechenergebnisse einer Stufe als Ausgangswerte für die nächste Stufe dienen. Die einzelnen Stufen dieser Pipeline werden im Folgenden genauer beschrieben.

4.3.1 Modellierung

Im ersten Schritt der Viewing-Pipeline wird normalerweise jedes relevante lokale Koordinatensystem (Modell- oder Objekt-Koordinatensystem) in das globale Koordinatensystem (Weltkoordinatensystem) überführt (siehe Kapitel 4.2.3).

Dieser Schritt ist im Rahmen dieser Arbeit jedoch nicht nötig, da außer dem Weltkoordinatensystem keine weiteren Koordinatensysteme existieren. Die einzelnen Elemente der Punktwolken sind bereits in Weltkoordinaten gegeben, genauso wie die weiteren Punkte, zum Beispiel von Koordinatensystemen, Bounding Boxes und so weiter.

4.3.2 Sicht-Transformation

In diesem Schritt der Viewing-Pipeline wird die Projektionsebene (oder auch View-Plane) VP eingeführt und ein Koordinatensystem auf dieser Ebene definiert.

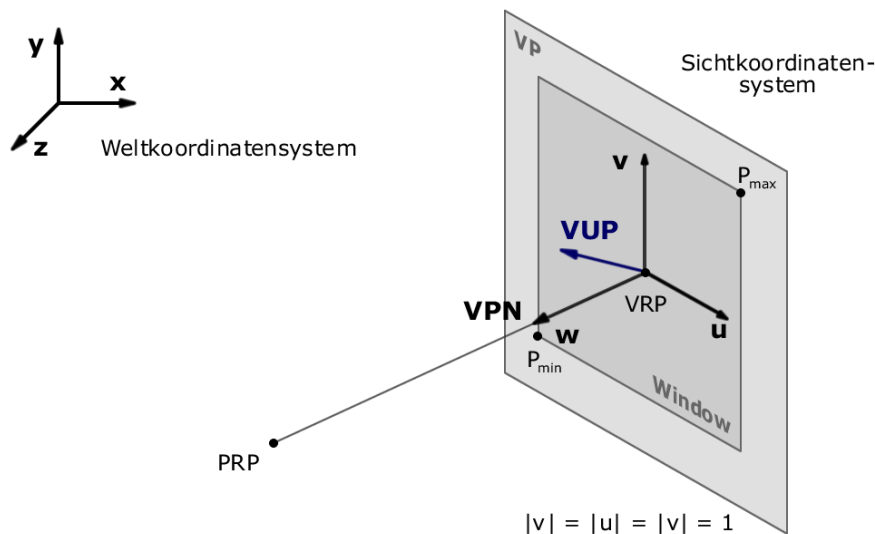


Abbildung 4.4: Aufbau des Sichtkoordinatensystems

Wie Abbildung 4.4 zeigt, wird das Koordinatensystem von den Vektoren u , v und w aufgespannt. Als Ursprung dient der sogenannte View-Reference-Point (VRP), der natürlich auf der Ebene VP liegen muss.

Zusätzlich werden noch die zwei Vektoren VPN und VUP eingeführt. Der View-Plane-Normal-Vector (VPN) steht senkrecht auf der Ebene VP im Punkt VRP und hat die Länge 1.

Der View-Up-Vector (VUP) liegt, anders als u und v , nicht in der Ebene VP (siehe Abbildung 4.4) und darf auch nicht senkrecht auf VP stehen. Der View-Up-Vector bestimmt, wie der Name schon nahelegt, wo bei der Ebene VP oben ist. Die Abbildung des Vektors VUP liegt dabei wieder auf der v -Achse.

In der Ebene VP liegen die zwei Punkte P_{min} und P_{max} mit $P_{min} = (u_{min}, v_{min})$ und $P_{max} = (u_{max}, v_{max})$. Sie legen den eigentlichen Sichtbereich fest (Window). Deswegen muss auch das Seitenverhältnis des definierten Rechtecks (Window) mit dem des Ausgabegerätes übereinstimmen.

Der Punkt PRP (Projection-Reference-Point) gibt die Position der Kamera oder des Betrachters an und wird deswegen auch Kamerastandpunkt oder Augenpunkt genannt. Der Abstand des Punktes PRP zur Ebene VP (Abstand der Punkte PRP und VRP) heißt d .

Dieses neue Modell (Abbildung 4.4) nennt man auch synthetisches Kameramodell. Das neu definierte Koordinatensystem wird als View-Reference-Coordinate-System (VRC) bezeichnet und muss im letzten Schritt noch in das Weltkoordinatensystem überführt werden, wozu folgende Schritte nötig sind (siehe auch Kapitel 4.2.3):

1. Verschieben des Punktes VRP in den Ursprung des Weltkoordinatensystems mit der Translation

$$T_{BA} = \begin{pmatrix} 1 & 0 & 0 & -vrp_x \\ 0 & 1 & 0 & -vrp_y \\ 0 & 0 & 1 & -vrp_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

2. Rotation des VRC um die y - und x -Achse, sodass die w -Achse auf der z -Achse des Weltkoordinatensystems zu liegen kommt wobei

$$R_y = \begin{pmatrix} \frac{vpn_z}{p} & 0 & -\frac{vpn_x}{p} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{vpn_x}{p} & 0 & \frac{vpn_z}{p} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

und

$$R_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & p & -vpn_y & 0 \\ 0 & vpn_y & p & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

die jeweiligen Rotationsmatrizen sind.

3. Rotation des VRC mit R_z um die z -Achse, sodass danach die v -Achse auf der y -Achse und damit auch die u -Achse auf der x -Achse liegt, wobei

$$R_z = \begin{pmatrix} \frac{vup'_y}{q} & -\frac{vup'_x}{q} & 0 & 0 \\ \frac{vup'_x}{q} & \frac{vup'_y}{q} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

sei.

Hierbei sei VUP' der Vektor, der durch Projektion des Vektors VUP auf die Ebene VP hervorgeht, und es gelte $|VUP'| = 1$. Außerdem sei $p = \sqrt{(vpn_x)^2 + (vpn_z)^2}$ und $q = \sqrt{(vup_x)^2 + (vup_y)^2}$.

4.3.3 Sichtvolumen-Transformation

Zieht man zwischen der Kameraposition und den Eckpunkten des oben festgelegten Rechtecks Linien, ergibt sich das in Abbildung 4.5 dargestellte Sichtvolumen, das durch die zwei Ebenen FP und BP abgeschlossen wird.

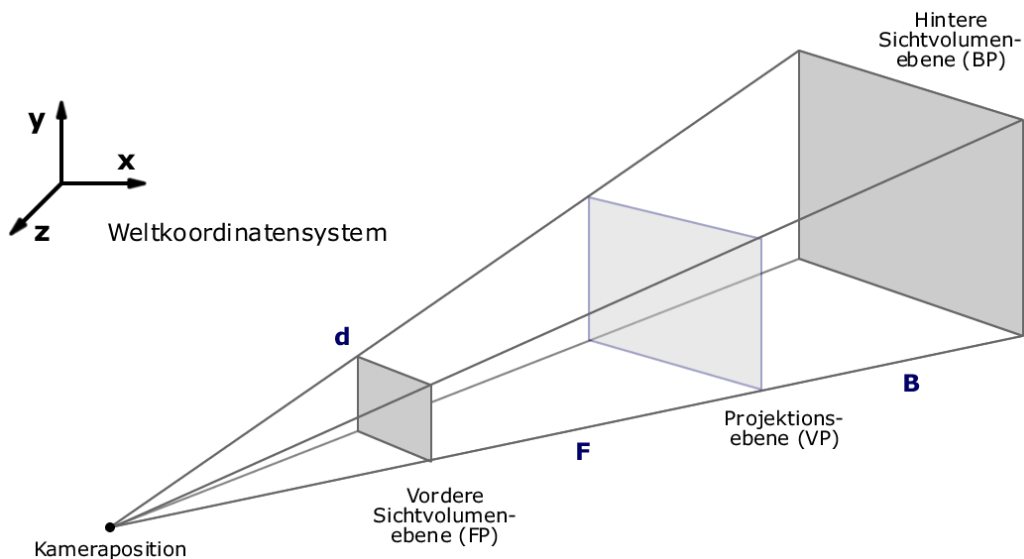


Abbildung 4.5: Sichtvolumen bei perspektivischer Projektion

Alle Objekte, die nicht innerhalb des Sichtvolumens liegen, werden bei der weiteren Abarbeitung der Viewing-Pipeline nicht weiter beachtet.

Im nächsten Schritt muss das Sichtvolumen in ein kanonisches Sichtvolumen überführt werden. Im Falle der hier betrachteten perspektivischen Projektion ist das kanonische Sichtvolumen ein regelmäßiger Pyramidenstumpf.

Dazu muss Folgendes getan werden:

1. Verschiebung des Augenpunktes PRP mit $PRP = (prp_x, prp_y, prp_z, 1)$ in den Ursprung des Weltkoordinatensystems mit Hilfe der Translationsmatrix T_A mit

$$T_A = \begin{pmatrix} 1 & 0 & 0 & -prp_x \\ 0 & 1 & 0 & -prp_y \\ 0 & 0 & 1 & -prp_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

2. Eine Scherung SH_{per} anwenden, um die Mittellinie des Sichtvolumens auf die z -Achse zu bringen (siehe Abbildung 4.6), wobei

$$SH_{per} = \begin{pmatrix} 1 & 0 & -\frac{vrp'_x + \frac{1}{2}(u_{min} + u_{max})}{vrp'_z} & 0 \\ 0 & 1 & -\frac{vrp'_y + \frac{1}{2}(v_{min} + v_{max})}{vrp'_z} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Die erwähnte Mittellinie ist definiert durch die beiden Punkte PRP und CW (center of view window). Der Punkt VRP' mit $VRP' = (vrp'_x, vrp'_y, vrp'_z, 1)$ ist der Punkt, der durch die vorhergehende Stufe der Viewing-Pipeline aus dem Punkt VRP hervorgegangen ist.

3. Das Sichtvolumen mit den Skalierungen S_1 und S_2 in einen regelmäßigen Pyramidenstumpf überführen (siehe Abbildung 4.6), wobei

$$S_1 = \begin{pmatrix} \frac{2 \cdot d}{u_{max} - u_{min}} & 0 & 0 & 0 \\ 0 & \frac{2 \cdot d}{v_{max} - v_{min}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

und

$$S_2 = \begin{pmatrix} \frac{1}{d+B} & 0 & 0 & 0 \\ 0 & \frac{1}{d+B} & 0 & 0 \\ 0 & 0 & \frac{1}{d+B} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Dabei ist B der Abstand der Ebene VP von der Ebene BP (siehe Abbildung 4.5) und d , wie schon oben beschrieben, der Abstand vom Augenzentrum zur Ebene VP .

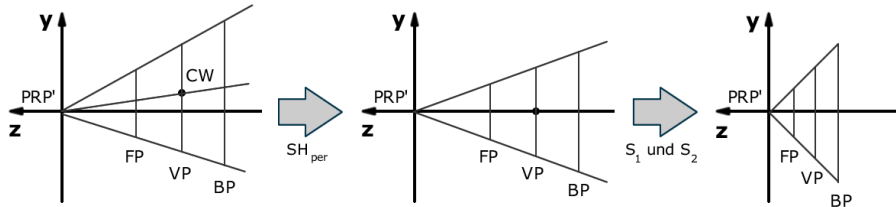


Abbildung 4.6: Transformation der Mitte des Sichtvolumens auf die z -Achse und in das kanonische Sichtvolumen

4.3.4 Projektionstransformation für die perspektivische Transformation

Aufgabe dieser Stufe der Viewing-Pipeline ist die in Abbildung 4.7 dargestellte Überführung des regelmäßigen Pyramidenstumpfes in einen Quader und die anschließende Umwandlung in den Einheitswürfel.

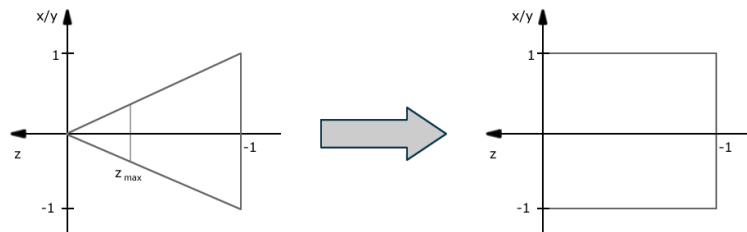


Abbildung 4.7: Überführen des Pyramidenstumpfes in einen Quader

Zusammenfassend ergibt sich die Transformationsmatrix A_E mit

$$A_E = \begin{pmatrix} -\frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & -\frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & \frac{-1}{1+z_{max}} & \frac{z_{max}}{1+z_{max}} \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

wobei gelten muss, dass $z_{max} = -\frac{d-F}{d+B}$ und $z_{max} \neq -1$.

Beachtet werden muss bei der Anwendung von A_E , dass dies eine nichtlineare Transformation ist. Das bedeutet, dass der Punkt $P = (x, y, z, 1)$ in den Punkt

$P'_{AE} = (x', y', z', w)$ überführt wird, bei dem $w \neq 1$ sein kann, weswegen noch eine Division aller Koordinaten durch w durchgeführt werden muss, so dass $P_{AE} = (x'/w, y'/w, z'/w, 1)$.

4.3.5 Bildschirm-Transformation

Im letzten Schritt der Viewing-Pipeline müssen nur noch die dreidimensionalen Koordinaten, die inzwischen im Einheitswürfel liegen, in das zweidimensionale Koordinatensystem des Ausgabemediums Device-Coordinate-System (DC) überführt werden. Die entsprechende Transformationsmatrix hierfür lautet:

$$T_{DC} = \begin{pmatrix} x_{max} & 0 & 0 & 0 \\ 0 & -y_{max} & 0 & y_{max} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Hier wird davon ausgegangen, dass im DC darstellbare Koordinaten nur zwischen 0 und x_{max} beziehungsweise zwischen 0 und y_{max} liegen. Da die dreidimensionalen Koordinaten aus der vorhergehenden Stufe der Viewing-Pipeline nur in $[0..1]$ liegen können, entspricht die Transformation T_{DC} also gleichzeitig einer Skalierung auf die verfügbare Auflösung des Ausgabegerätes.

Mit dieser letzten Transformation ist die Viewing-Pipeline abgearbeitet, und die Koordinaten liegen jetzt in der benötigten Form (2D, auf Ausgabegerät-Auflösung skaliert) vor.

Kapitel 5

Weitere Grundlagen

5.1 Skriptsprachen und Datenbanken

5.1.1 PHP

Die Skriptsprache PHP kommt im Rahmen dieser Arbeit zur Anwendung, um korrekte HTTP-Header für die Java Web Start-Konfigurationsdateien (.jnlp-Dateien) zu erzeugen, damit diese über den Browser des Benutzers der richtigen Applikation zugeordnet werden können, und gegebenenfalls, um GET-Variablen auszuwerten, die an PHP-Seiten gesendet und in Parameter für Applets umgewandelt werden müssen. Zudem sollen diese Parameter, die unter anderem bestimmen, welche Benutzersysteme dem Anwender angezeigt werden, auf dem Rechner des Benutzers in Form von Cookies gespeichert werden. So ist es möglich, dass der Benutzer immer nur die Systeme angezeigt bekommt, die für ihn interessant sind.

5.1.2 ASP.NET und Microsoft SQL Server

Da die Datenbank (Microsoft SQL Server 2000), die zur Speicherung der Evaluationsergebnisse dient, auf einem Windows 2003-Server liegt, und der Zugriff von außen auf den Port 1434 unterbunden wurde, müssen die Funktionen zur Erfassung und Auswertung der Fragebogendaten in ASP.NET und mittels ADO.NET implementiert werden. Da auf diesem Server PHP nicht verfügbar ist, mussten auch alle anderen dynamischen Seiten mit ASP.NET umgesetzt werden.

5.1.3 Weitere Datenbanken

Die im Vorprojekt durchgeführten Untersuchungen der verfügbaren Technologien wurden in zwei Bereiche geteilt. Zum einen wurden möglichst viele Plugin-Technologien für die 3D-Visualisierung (Frontend) und zum anderen einige mögliche Datenbanken für die Speicherung verschiedener Daten (Backend) geprüft.

Die berücksichtigten, frei verfügbaren Datenbanken waren dabei:

- MySQL (4.1)
- PostgreSQL (7.3.4)
- Hypersonic SQL (1.7.1)
- Interbase (7.1)

Eine detaillierte Bewertung der einzelnen Datenbanken ist im Anhang A zu finden.

Im weiteren Verlauf der Arbeit wurde auf Wunsch der Betreuer auf die angestrebte Session-Verwaltung mit Login- und Email-Datenhaltung mittels einer Datenbank verzichtet. Die Daten sollen zukünftig vom AnT-Server beziehungsweise verschiedenen Skripten in den jeweiligen Verzeichnissen verwaltet werden.

5.2 Extreme Programming

Als Vorgehensmodell wurde für diese Diplomarbeit eine etwas abgewandelte Version des Extreme Programmings genutzt. Im Folgenden wird dieses Prozessmodell kurz erläutert, und danach wird näher auf die Vorteile in Bezug auf diese Arbeit eingegangen.

5.2.1 Kurzüberblick

Extreme Programming ist ein agiler Softwareentwicklungsprozess, zu dem es ab 1995 erste Veröffentlichungen gab, und der ursprünglich unter anderem von Kent Beck und Martin Fowler vorgestellt wurde.

Extreme Programming gilt als leichtgewichtiger Prozess, was bedeutet, dass der Prozess codezentriert ist und die rasche Erstellung eines lauffähigen Systems zum Ziel hat. Er ist besonders für kleinere Projekte mit kleinen oder mittleren Teams geeignet.

Anders als bei den meisten schwergewichtigen Entwicklungsprozessen wie zum Beispiel RUP, V-Modell, Wasserfallmodell und so weiter wird beim Extreme Programming in den Entwurf relativ wenig Aufwand investiert, während Implementierung und Test eine bedeutendere Stellung einnehmen. Besonders gut geeignet ist Extreme Programming, wenn eine enge Zusammenarbeit mit dem Kunden möglich ist, sodass dessen Vorstellungen und Wünsche in jeden Iterationsdurchgang einfließen können.

Der Extreme Programming-Prozess besteht im Wesentlichen aus nur zwei Phasen. Die Planungsphase dient zum Verständnis und zur Dokumentation der Anforderungen, während in der Iterationsphase in beliebig vielen Teilschritten das System implementiert und getestet wird.

5.2.2 Grundwerte von Extreme Programming

- **Kommunikation**

Kommunikation ist eine der wesentlichen Säulen dieses Prozessmodells. Hierbei ist sowohl die Kommunikation innerhalb des Projektteams gemeint, als auch besonders die Kommunikation mit dem Kunden. Kommunikation ist so bedeutend, dass es auch Grundlage vieler in XP beschriebenen Praktiken (wie zum Beispiel das Programmieren in Paaren) ist.

- **Einfachheit**

Für Probleme innerhalb eines Projektes sollte immer die einfachste Möglichkeit zur Lösung verfolgt und umgesetzt werden. Dies ist besonders deswegen unerlässlich, weil der Quellcode einen wichtigen Teil der Dokumentation darstellt und für alle zugänglich ist. In einer Iteration sollen nur die unbedingt nötigen Ziele umgesetzt werden, und es soll somit möglichst viele und möglichst kleine Iterationsschritte geben. Zudem soll nie mehr umgesetzt werden, als verlangt wurde.

- **Feedback**

Mit Feedback sind hier zum einen die Rückmeldungen des Auftraggebers nach Abschluss eines Iterationsschrittes an das Team gemeint, zum anderen aber auch die Ergebnisse von Tests innerhalb eines Iterationsschrittes für die Programmierer.

- **Mut**

Im Extreme Programming-Entwicklungsprozess ist es essenziell, dass man auch den Mut für einschneidende Änderungen aufbringt. Man muss zum Beispiel bereit sein, bei Problemen größere Teile des entstandenen Codes zu verwerfen und mit den gesammelten Erfahrungen neu anzufangen.

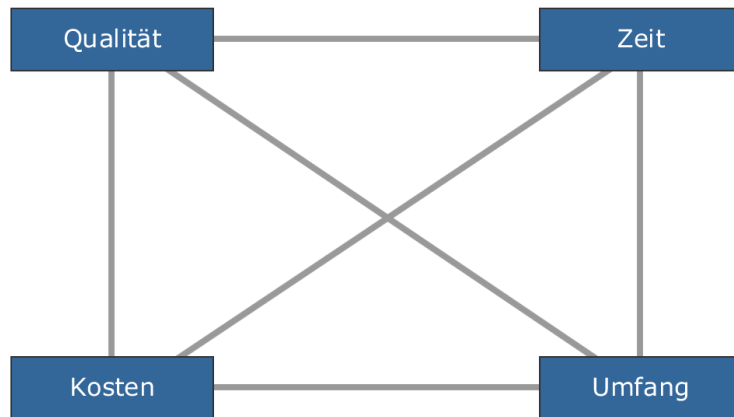


Abbildung 5.1: Abhängigkeiten bei der Softwareentwicklung

Abbildung 5.1 zeigt die verschiedenen Einflussfaktoren beim Erstellen von Software und deren Zusammenspiel. Keine der Größen kann unabhängig von den anderen geändert werden.

Kapitel 6

Projektlauf

6.1 Vorprojekt

Im Rahmen des Vorprojektes, das noch sehr eng mit der Untersuchung der verfügbaren Visualisierungsmöglichkeiten verknüpft war, sind für Java, Java3D, Flash, Shockwave und VRML erste Prototypen zur 3D-Darstellung von zufällig generierten Punktwolken entstanden.

Die Darstellungen waren, abgesehen von der Java-Version, noch nicht vollständig steuerbar und sollten unter anderem dazu dienen, einen besseren Einblick in die jeweilige Plattform zu erhalten und die Performance im Vergleich zu allen anderen Möglichkeiten besser beurteilen zu können.

Nachdem die Entscheidung für Java gefallen war, wurde der Java-Prototyp weiter ausgebaut. Neben dem eigentlichen Prototyp zur Visualisierung musste dazu auch ein vorläufiger Server implementiert werden, der in der Lage war, Punktdaten und Bilder auf Anfrage zurückzuliefern. Allerdings wurden hierzu nicht die eigentlichen Datensätze verwendet, wie sie von AnT erzeugt werden, sondern vom Server dynamisch berechnete Punkte eines Lorenz-Systems.

Mit Hilfe dieses Servers konnte dann der Prototyp soweit entwickelt werden, dass er bereits über die wesentlichsten Fähigkeiten der fertigen Anwendung (Anzeige von zwei- und dreidimensionalen Darstellungen von Punktwolken) verfügte.

Einen kleinen Überblick über die Möglichkeiten des Prototyps geben auch die Abbildungen 6.1, 6.2 und 6.3.

Abbildung 6.1 zeigt die Steuerungsoberfläche des Prototyps. Hier ist es möglich, einen Server anzugeben, mit dem eine Verbindung hergestellt werden soll. Zu-

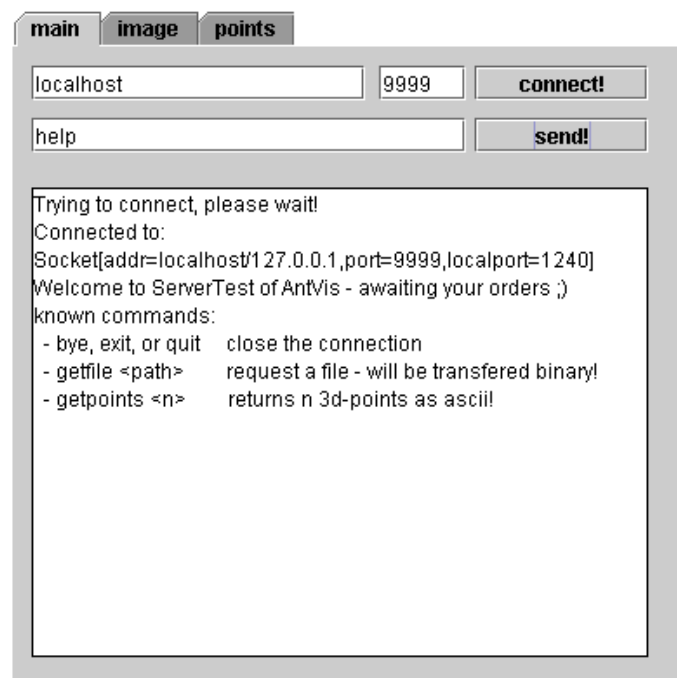


Abbildung 6.1: Steuerungsoberfläche des Prototyps

sätzlich ist es über diese Oberfläche auch möglich, Befehle einzugeben und diese an den Server abzusetzen (zum Beispiel „help“, „getimage bild.png“, „getpoints 100“). Die Reaktion kann in einem Textfeld beobachtet werden.

Abbildung 6.2 zeigt den Prototyp mit einem geladenen Bild. Hier ist es lediglich möglich, die Ebene mit dem Bild zu scrollen. Ein Interface für gnuplot, wie es für die endgültige Version des Applets nötig ist, um je nach Bedarf verschiedene Bilder von Systemen zu generieren, wurde hier noch nicht realisiert.

In die erste Implementierung einer dynamischen, dreidimensionalen Visualisierung wurde deutlich mehr Aufwand investiert. Wie Abbildung 6.3 zeigt, ist bereits hier eine interaktive, dreidimensionale Darstellung der übertragenen Punkte möglich. So kann die dargestellte Punktwolke beispielsweise bereits gedreht und verschoben werden.

Der Prototyp erwies sich als ideale Plattform, um verschiedene Techniken und Konzepte, wie sie in Kapitel 3 beschrieben wurden, im praktischen Einsatz zu testen.

Die gesammelten Erfahrungen und begangenen Fehler machten sich besonders beim Entwurf der endgültigen Software, der Abschätzung des Aufwandes und bei der eigentlichen Implementierung bezahlt.

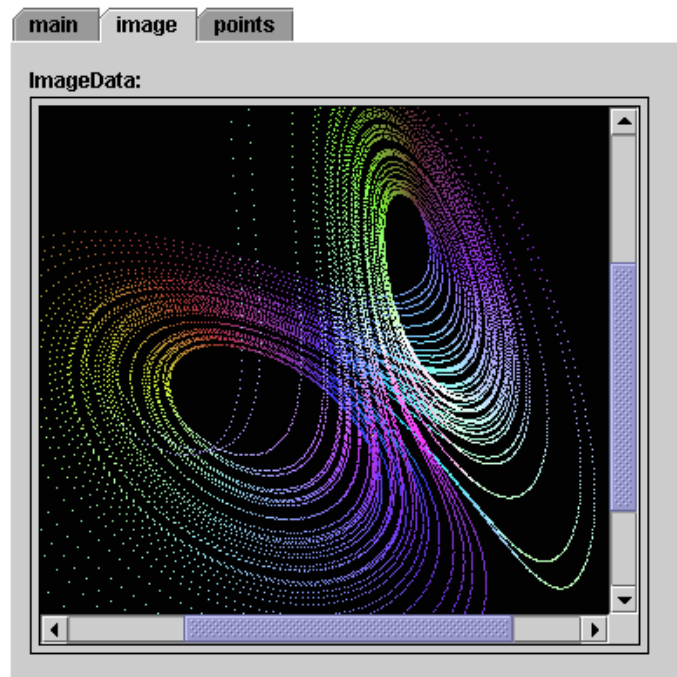


Abbildung 6.2: 2D-Panel des Prototyps

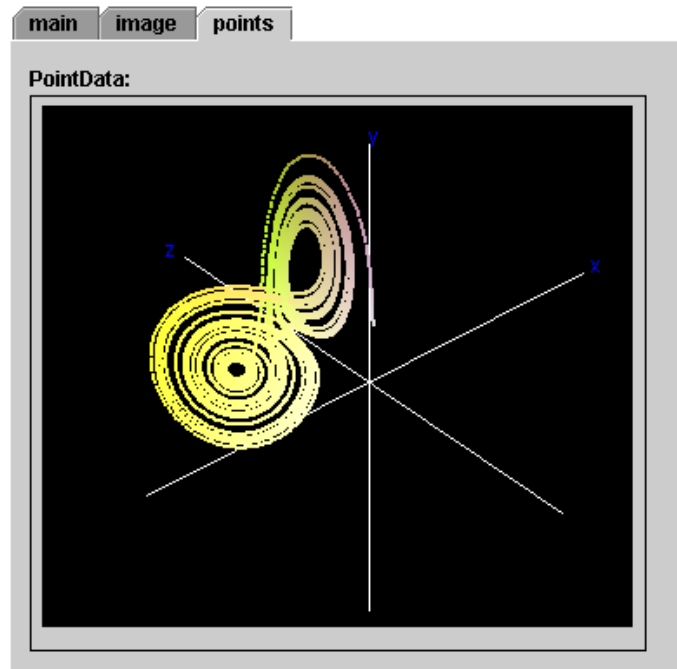


Abbildung 6.3: 3D-Panel des Prototyps

6.2 Vorgehensmodell

Das verwendete Vorgehensmodell ist eine Variante des „Extreme Programming“, wie es in Kapitel 5.2 kurz erläutert wird.

Die wichtigsten Punkte waren hier vor allem:

- möglichst kleine Iterationsschritte
- dauernde Funktions- und Integrationstests

Die Kommunikation innerhalb des Teams entfiel bei einem Ein-Mann-Projekt, aber die Kommunikation mit den Betreuern, die gleichzeitig auch die Rolle des Kunden inne hatten, war während der gesamten Arbeit sehr wichtig. Viele Zwischenpräsentationen der vorläufigen Ergebnisse und eine sehr enge Zusammenarbeit stellten sicher, dass die Software möglichst optimal an die Bedürfnisse der Anwender und Wünsche des Kunden angepasst wurde.

6.3 Anforderungen

Im folgenden Kapitel werden die Anforderungen vorgestellt, die die Grundlage für die endgültige Entwicklung waren.

6.3.1 Nichtfunktionale Anforderungen

Allgemeine Bedingungen

Die Applikation wird in Java implementiert und soll möglichst ab der Java-Version 1.2, mindestens aber ab Version 1.4, lauffähig sein. Die Applikation soll einfach über das World Wide Web gestartet werden können und muss möglichst unabhängig von der Hard- und Softwareausstattung des Benutzers sein (Betriebssystem, Browser, ...).

Leistungsanforderungen

Da die Software ohne besondere Hardwareunterstützung auskommen muss, ist besonders darauf zu achten, dass die verschiedenen Visualisierungen möglichst effizient implementiert und unabhängig voneinander sind. Auch wenn mehrere

Dateiübertragungen im Gange sind, muss die Bedienbarkeit der Anwendung gewährleistet bleiben.

Besondere Qualitäten

- Die Anwendung ist möglichst umfassend und einfach über externe Textdateien konfigurierbar. Hier ist es insbesondere einfach möglich, die Anwendung später an verschiedene Sprachen anzupassen.
- Die Applikation wird umfassend kommentiert, und es wird auch sonst besonders auf eine spätere Erweiterbarkeit der Software geachtet.
- Es wird besonderer Wert auf die Qualität und Einfachheit der Benutzungsoberfläche gelegt. Hierbei ist die ISO Norm 9241 Teil 10 als Grundlage zu verwenden.
- Die Software ist möglichst robust und wenig fehleranfällig. Selbst bei nicht korrigierbaren Fehlern wird der Benutzer über die möglichen Ursachen informiert und die Anwendung geordnet beendet.

6.3.2 Allgemeine funktionale Anforderungen

- In der Anwendung werden AnT-Dateien über eine Baumstruktur ausgewählt. Die gewählte Datei kann dann entweder dreidimensional, dynamisch und interaktiv visualisiert werden, es kann eine statische Abbildung erzeugt werden oder die Datei kann direkt eingesehen werden. Weitere Details zu den Visualisierungen folgen im nächsten Kapitel.
- Es ist möglich, beliebig viele Dateien gleichzeitig zu übertragen, und pro Datei kann es beliebig viele Visualisierungen, egal welcher Art, geben. Die bereits übertragenen Punktdaten sollen auf keinen Fall verloren gehen.
- Die laufenden Dateiübertragungen werden übersichtlich so dargestellt, dass der Übertragungsfortschritt erkennbar ist. Es ist jederzeit möglich, die Übertragung einer einzelnen Datei abubrechen, wobei aber auch hier die bereits übertragenen Punktdaten bis zum Beenden des Applets/der Applikation noch verfügbar sind. Dies gilt allerdings nicht für übertragene Bilder, da diese zu klein sind, als dass eine Anzeige des Übertragungsfortschritts Sinn macht.

6.3.3 Anforderungen an die GUI

Im Folgenden werden die Anforderungen an die grafische Benutzungsoberfläche von AnT-Online kurz in Form von Use-Cases umrissen und dann genauer erläutert.

Alle benötigten Elemente und deren Unterelemente werden kurz erklärt, und abschließend wird eine Abbildung als Beispiel für eine mögliche Implementierung angeführt. Die entsprechenden Abbildungen (6.7,6.8,6.9,6.10,) wurden dabei mit Hilfe eines Grafikprogramms und unter Verwendung von Screenshots des Prototypen erstellt, um das angestrebte Aussehen der GUI schon in diesem Stadium der Arbeiten festlegen zu können.

Use-Cases

In diesem Kapitel werden exemplarisch für die drei wichtigsten Bereiche der Software Use-Cases vorgestellt.

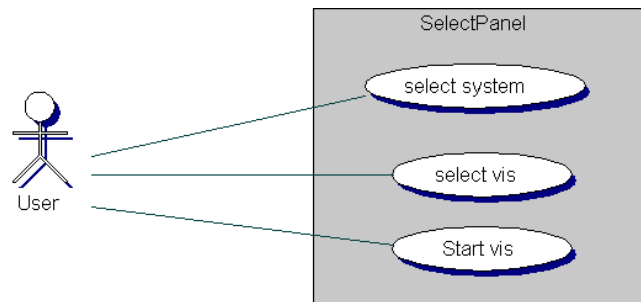


Abbildung 6.4: Select-Use-Case

Abbildung 6.4 zeigt die Aktionen, die dem Benutzer beim ersten Interaktionsschritt mit der Software möglich sind.

Abbildung 6.5 und 6.6 zeigen jeweils mögliche Aktionen in den Bereichen der Software, die der zwei- und dreidimensionalen Visualisierung dienen.

Benötigte Panel

Es können fünf unterschiedliche Klassen von Panels unterschieden werden:

1. Selection-Panel
2. Transfer-Panel

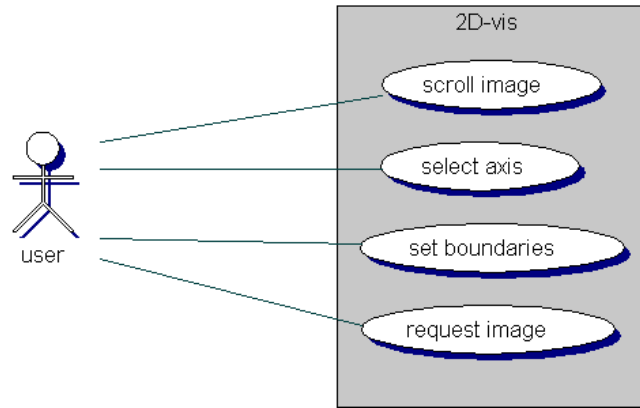


Abbildung 6.5: 2D

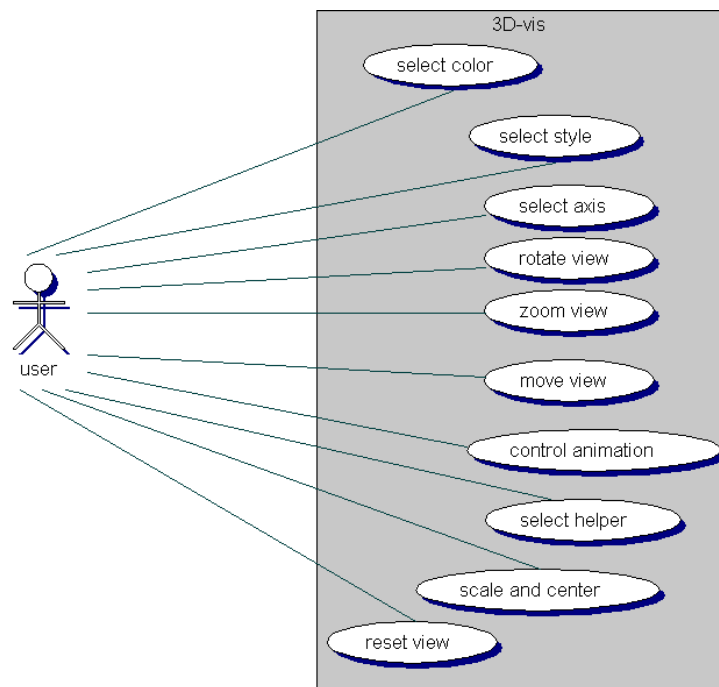


Abbildung 6.6: 3D

3. Image-Panel
4. 3D-Viewer-Panel
5. Help-Panel

Von den Klassen Selection-Panel, Transfer-Panel und Help-Panel kann es jeweils nur eine Instanz geben, während es beliebig viele Image- und 3D-Viewer-Panel-Instanzen geben kann.

Im Folgenden wird jede der Panel-Klassen genauer beschrieben.

Selection-Panel

In dem Selection-Panel (Abbildung 6.7) wird eine Auswahl an verfügbaren Demodatensätzen in einer Baumstruktur dargestellt. Zusätzlich hat der Benutzer die Möglichkeit, die ID eines Systems auszuwählen, das für ihn berechnet wurde. Diese ID bezieht der Benutzer aus einer Email, die der Server versendet, wenn ein Benutzersystem fertig berechnet wurde.

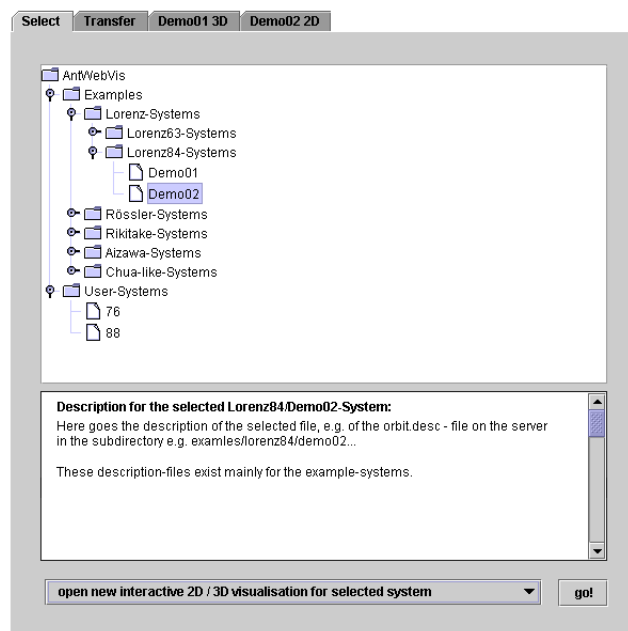


Abbildung 6.7: Spezifikation des Select-Panels

Die ID soll dem Client auch über einen Link verfügbar gemacht werden können, also als Parameter für das Applet. Dieser Parameter könnte über eine Skriptsprache, wie PHP, Perl oder andere aus der URL über einen GET-Parameter extrahiert

werden. Optional könnten auch alle bisher gerechneten Systeme per Cookie auf der Client-Seite oder als Dateien oder in einer Datenbank auf Server-Seite gespeichert werden. Der Client zeigt sie dann alle in einem eigenen Knoten unter den vorher berechneten Beispielsystemen an.

Neben der Baumdarstellung der verfügbaren Systeme und deren verschiedenen Ausprägungen bietet das Selection-Panel Platz, um Beschreibungen, die vom Server zu den einzelnen Systemen und deren Ausprägungen geliefert werden können, anzuzeigen.

Transfer-Panel

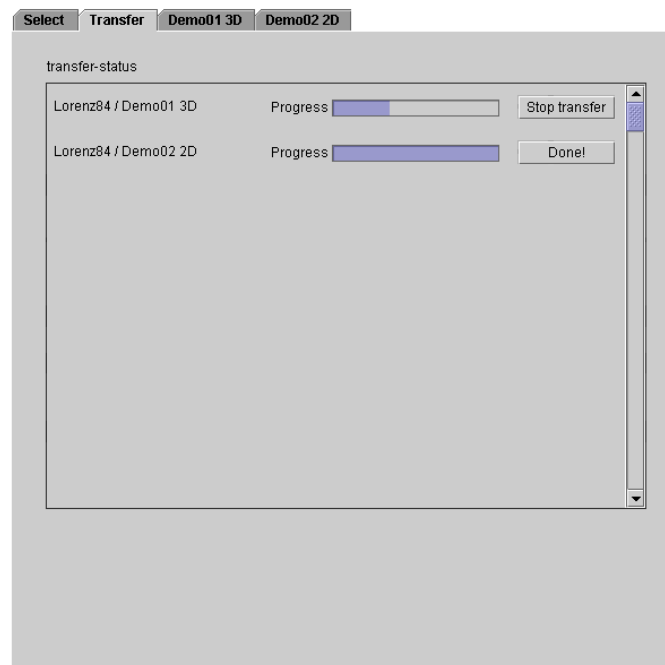


Abbildung 6.8: Spezifikation des Transfer-Panels

Da es theoretisch möglich sein soll, beliebig viele AnT-Dateien parallel zu übertragen, soll das Transfer-Panel einen Überblick über den Status aller Datenübertragungen geben (siehe Abbildung 6.8).

Es gibt für jede AnT-Datei eine kurze Beschreibung, eine Fortschrittsanzeige und einen Button, mit dem die Übertragung abgebrochen werden kann. Dieser Button wechselt nach der Übertragung seinen Zustand, um die beendete Übertragung anzuzeigen.

Image-Panel

Das Image-Panel (siehe Abbildung 6.9) dient dazu, Bilder, die auf dem Server generiert wurden, anzuzeigen. Hierbei soll es dem Benutzer auch möglich sein, die zu visualisierenden Spalten aus der AnT-Datei zu wählen, sowie jeweils den zu visualisierenden Bereich festzulegen.

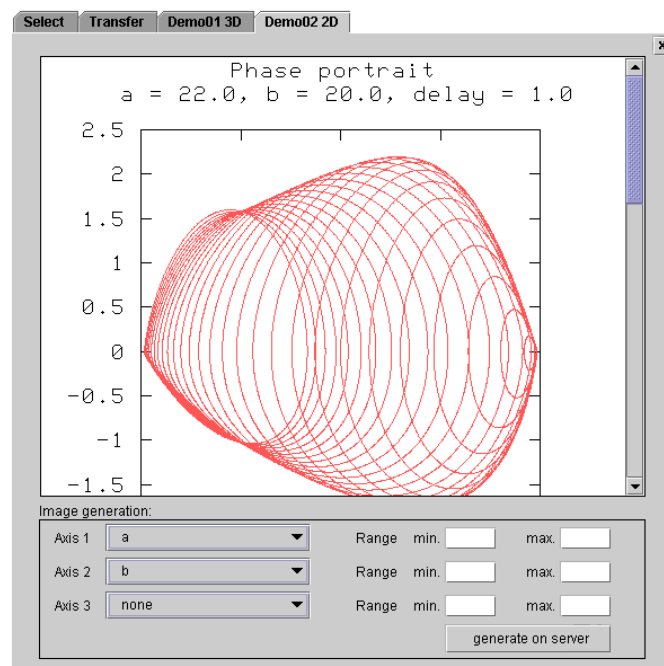


Abbildung 6.9: Spezifikation des Image-Panels

Über einen Button kann dann die Generierung des Bildes auf dem Server angestoßen werden, sobald vom Benutzer alle gewünschten Einstellungen vorgenommen sind.

Sobald der Benutzer seinen Generierungsauftrag abgesetzt hat, wird eine Nachricht eingeblendet, dass die Daten erzeugt und übertragen werden. Diese Nachricht wird erst durch das geladene Bild ersetzt.

3D-Viewer-Panel

Diese Klasse von Panels bietet im Wesentlichen sechs Funktionseinheiten:

Die wichtigste Einheit dient der 3D-Darstellung der Punktwolke. Hier ist es durch direkte Manipulation mit der Maus möglich, die Punktwolke und gegebenenfalls

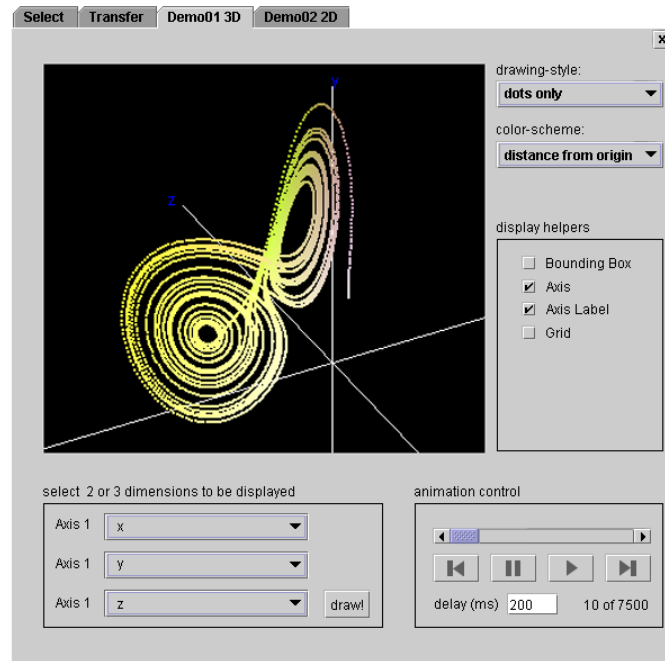


Abbildung 6.10: Spezifikation des 3D-Viewer-Panels

das Koordinatensystem, eine Bounding-Box, Gitter und so weiter zu drehen, zu skalieren und zu verschieben. Es ist auch ein Reset-Button verfügbar, der die Darstellung wieder in den Ursprungszustand zurückversetzt, in der alle Punkte der darzustellenden Punktwolke sichtbar sind. Die Anzeige der einzelnen Punkte ist sofort nach Start der Datenübertragung möglich. Wird nichts anderes angegeben, werden alle Punkte angezeigt, sobald sie verfügbar sind. Es ist aber auch möglich, die übertragenen Punkte wieder auszublenden und mit einer gegebenen Geschwindigkeit hintereinander anzeigen zu lassen.

Über eine Funktionseinheit im jeweiligen 3D-Viewer-Panel ist es möglich, immer jeweils zwei oder drei der n zur Verfügung stehenden Spalten der AnT-Datei zur Darstellung auszuwählen. Dies wird zum Beispiel über eine Reihe von Check- oder Drop-Down-Boxen möglich.

Eine andere Funktionseinheit innerhalb des 3D-Viewer-Panels dient zur Auswahl der Färbungsmethode für die jeweilige Punktwolke. Hier können mindestens die vier folgenden Methoden gewählt werden:

- jeder Punkt wird mit der Standardfarbe eingefärbt
- jeder Punkt wird einzeln und unabhängig von anderen Punkten entsprechend seines Abstandes vom Ursprung, beziehungsweise entsprechend sei-

ner Koordinaten eingefärbt

- jeder Punkt wird abhängig vom Abstand des Punktes zu seinem Vorgänger eingefärbt (Einfärben nach Geschwindigkeit)
- jeder Punkt n wird abhängig vom Abstand zum Punkt $n - 2$ eingefärbt (Einfärben nach Beschleunigung)

Help-Panel

Im Help-Panel soll es dem Benutzer möglich sein, die Online-Hilfe zu der Applikation einzusehen.

Die Seiten werden dabei als HTML-Dokumente auf dem Webserver abgelegt. So ist es auch später jederzeit einfach möglich, die Hilfe zu überarbeiten oder zu erweitern.

6.4 Entwurf

6.4.1 Überblick über die AnT-Online-Architektur

Abbildung 6.11 gibt einen ersten Überblick über den Aufbau des AnT-Online-Systems. Aus dieser Abbildung geht, neben dem internen Aufbau auch hervor, woher AnT-Online die verschiedenen Arten von Daten bezieht.

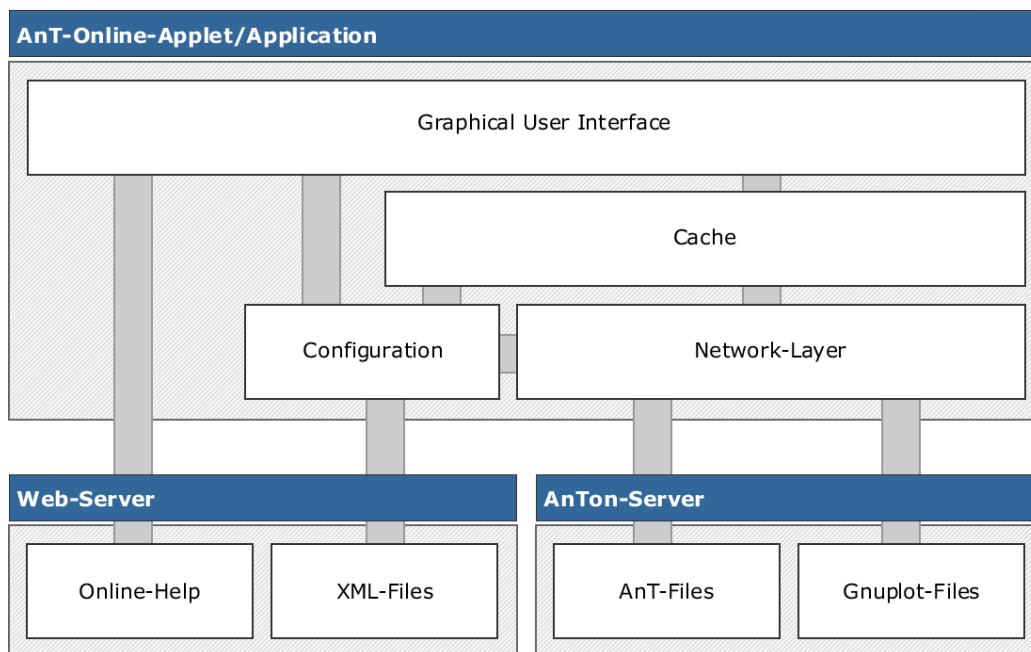


Abbildung 6.11: Grobübersicht

- **Online-Hilfe**

Die Daten für die Online-Hilfe in AnT-Online liegen in Form von HTML-Dateien auf dem Webserver. Hierbei ist zu beachten, dass diese HTML-Seiten für die Anzeige in Java optimiert sind, also nicht alle Möglichkeiten ausgenutzt werden können, die der HTML 3.2-Standard bietet.

Zudem sollten die HTML-Daten nicht unverändert für normale Browser angeboten werden, da die Darstellung (besonders im Internet Explorer 6) sehr von der Darstellung innerhalb von Java abweichen kann.

- **XML-Dateien**

Es können zur Konfiguration des Applets/der Applikation beliebig viele XML-Dateien auf dem Webserver existieren. Welche dieser XML-Dateien zur Konfiguration verwendet werden soll, geht aus einem Parameter hervor,

der in der JNLP-Datei enthalten oder der HTML-Seite enthalten sein muss, in der das Applet eingebunden ist.

- **AnT-Dateien**

Über den AnTon-Server werden die Daten aus den Dateien, die von AnT generiert wurden, direkt nach außen gegeben und über die Netzwerkschicht an den Cache übergeben, um dann in einer oder mehreren Visualisierungen in der GUI zu enden.

- **Gnuplot-Dateien**

Die Bilddateien (PNG) werden über die Gnuplot Schnittstelle des AnTon-Server erzeugt und direkt von diesem zur Verfügung gestellt. Über die Netzwerkschicht und den Cache werden die Bilder an die GUI weitergegeben und dort dann dargestellt.

6.4.2 Überblick über den internen Aufbau von AnT-Online

Package-Struktur

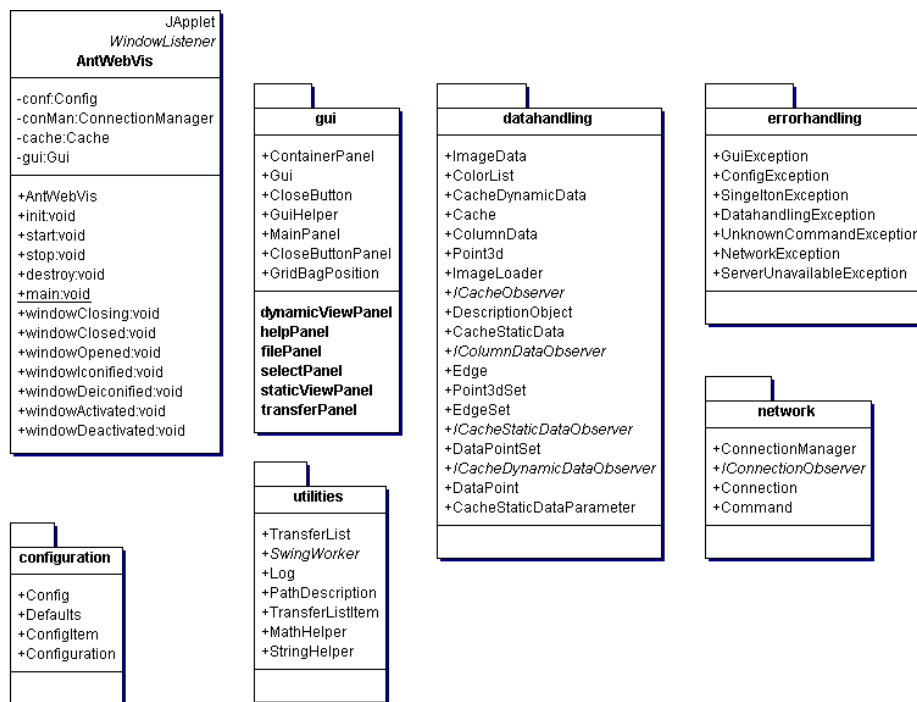


Abbildung 6.12: Paketübersicht zu AnT-Online

Die AnT-Online-Visualisierung besteht im Wesentlichen aus sechs Paketen, wie auch Abbildung 6.12 zeigt.

Die Hauptklasse „AntWebVis“ ist so gestaltet, dass sich die Software als Applet, aber auch als normale Applikation oder Java Web Start-Applikation starten lässt.

Das Package „configuration“

In diesem Package ist die gesamte Funktionalität enthalten, um die Konfigurationsdaten, die als XML-Dateien auf dem Webserver liegen, zu laden, in den entsprechenden Datentyp umzuwandeln, in eine Hashtable einzufügen und für den Zugriff bereitzustellen.

Das Package „datahandling“

Im Package „datahandling“ verbirgt sich der komplette Cache sowie die zugehörige Verwaltungslogik. Wichtigste Funktion ist das Aufnehmen von Punktdaten aus der Netzwerkschicht und deren Umwandlung von Zeichenketten in float-Werte.

Das Package „errorhandling“

Dieses Packet enthält sämtliche eigenen Exception-Typen, die in der gesamten Software Verwendung finden.

Das Package „gui“

Das Package „gui“ ist das umfangreichste der sechs Packages. Es enthält noch eine Reihe von weiteren Packages und umfasst selbst Klassen zu allgemeinen Funktionen der GUI.

- **dynamicViewPanel**
Enthält alle Klassen, die zusammen die GUI-Elemente der dynamischen, interaktiven, dreidimensionalen Visualisierung bilden.
- **helpPanel**
Die Klassen zur Anzeige der Online-Hilfe innerhalb der GUI.
- **filePanel**
Alle Klassen, die für die Dateiansicht nötig sind.

- **selectPanel**
Enthält Klassen, die für die Erzeugung des Auswahl-Tabs in der GUI mit der Baumstruktur verantwortlich sind.
- **staticViewPanel**
Dieses Package enthält die Klassen, die in der GUI die statische Visualisierung bilden.
- **transferPanel**
Ein Package mit Klassen, die für die Anzeige und Steuerung der Datenübertragungen innerhalb der GUI nötig sind.

Das Package „network“

Hier sind alle Klassen zu finden, die mit der Serverkommunikation über TCP zu tun haben.

Das Package „utilities“

Dieses Package dient dazu, alle Klassen zu sammeln, die Funktionen anbieten, welche in der ganzen Anwendung benötigt werden.

Detailliertere Klassendiagramme zu den verschiedenen Packages sind auch in Kapitel A.2.4 zu finden.

6.4.3 Aufbau der Konfiguration

Für jeden Wert in der Konfiguration muss in der entsprechenden XML-Datei ein Item-Knoten angegeben werden, der beispielsweise folgenden Aufbau besitzt:

```
<item
  name="NET_PORT"
  value="4669"
  type="int"
/>
```

Innerhalb der Anwendung kann dann auf diesen Konfigurationseintrag mit `Conf.getInstance().getInt("NET_PORT")` zugegriffen werden.

Es existieren für die Datentypen `int`, `String`, `String[]`, `float`, `double`, `Color` und `Border` eigene Zugriffsmethoden. Wenn auf einen Eintrag mit der falschen Methode

zugegriffen wird (zum Beispiel `getInt` statt `getString`), wird ein Standardwert zurückgeliefert (hier 0). Das gleiche gilt auch, wenn auf einen nicht existierenden Schlüssel zugegriffen wird. Hier wird ebenfalls ein Standardwert zurückgeliefert und eine Warnung an die Konsole ausgegeben.

Es folgt eine kurze Beschreibung der verfügbaren Konfigurationseinträge (komplettes Beispiel siehe Anhang C):

- `LOG_LEVEL` - Gibt an, ab welchem Wert Meldungen über die „Log“-Komponente an die Konsole ausgegeben werden sollen.
- `LOG_CRITICAL_ERROR` - Gibt an, welchen Wert kritische Fehler haben. Legt zusammen mit `LOG_LEVEL` fest, ob kritische Fehler ausgegeben werden oder nicht.
- `LOG_MEDIUM_ERROR` - Gibt an, welchen Wert mittlere Fehler haben. Legt zusammen mit `LOG_LEVEL` fest, ob mittlere Fehler ausgegeben werden oder nicht.
- `LOG_LIGHT_ERROR` - Gibt an, welchen Wert leichte Fehler haben. Legt zusammen mit `LOG_LEVEL` fest, ob leichte Fehler ausgegeben werden oder nicht.
- `LOG_WARNING` - Gibt an, welchen Wert Warnungen haben. Legt zusammen mit `LOG_LEVEL` fest, ob Warnungen ausgegeben werden oder nicht.
- `LOG_MESSAGE` - Gibt an, welchen Wert Nachrichten haben. Legt zusammen mit `LOG_LEVEL` fest, ob Nachrichten ausgegeben werden oder nicht.
- `LOG_DEBUG` - Gibt an, welchen Wert Debug-Meldungen haben. Legt zusammen mit `LOG_LEVEL` fest, ob Debug-Meldungen ausgegeben werden oder nicht. Debug ist der Standardwert, wenn bei der Ausgabe über die „Log“-Klasse keine Priorität angegeben wird.
- `NET_SERVER` - Der Name des AnTon-Servers, von dem die Anwendung ihre Daten beziehen soll.
- `NET_PORT` - Der Port, auf dem der AnTon-Server unter der durch `NET_SERVER` angegebenen Adresse zu erreichen ist.
- `NET_EOT` - „end of transfer“-Symbol. Diesen String sendet der Server am Ende jeder Übertragung.
- `NET_FILESIZE` - Schlüsselwort, hinter dem die Größe der als nächstes zu übertragenden Datei in Bytes steht.

- NET_PATH_SEPERATOR - Trennzeichen, das bei Pfadangaben zwischen den einzelnen Verzeichnissen steht.
- NET_NUMBER_DELIMITER - Trennzeichen, das bei Zeilen aus AnT-Dateien zwischen einzelnen Werten steht.
- NET_COMMENT_DELIMITER - Zeichen, nach dem der Rest einer Zeile aus einer AnT-Datei als Kommentar erkannt wird.
- NET_PROPERTY_DELIMITER - Das Zeichen, das verschiedene Eigenschaften einer Datei, wie etwa Größe, Name, Zugriffsrechte und so weiter trennt.
- NET_ASTERIX - String mit einem Wildcard-Zeichen, das für eine beliebige Zeichenfolge steht, wie zum Beispiel in „*.tna“.
- NET_COLUMN_NO_DELIMETER1 - Trennzeichen, nach dem eine Spaltennummer zu finden ist.
- NET_COLUMN_NO_DELIMETER2 - Abschlusszeichen, vor dem eine Spaltennummer steht.
- NET_COLUMN_TXT_DELIMETER1 - Anfang eines Spaltennamens.
- NET_COLUMN_TXT_DELIMETER2 - Ende eines Spaltennamens.
- NET_COLUMN_NAME - Zeichenfolge, nach der der Name einer Zeile in einer AnT-Datei zu finden ist.
- NET_COLUMN_MAX - Zeichenfolge, nach der der maximale Wert einer Spalte in einer AnT-Datei zu finden ist.
- NET_COLUMN_MIN - Zeichenfolge, nach der der minimale Wert einer Spalte in einer AnT-Datei zu finden ist.
- NET_COLUMN_DESC - Zeichenfolge, nach der eine kurze Beschreibung einer Spalte in einer AnT-Datei zu finden ist.
- NET_HELP_PAGE_URL - URL zur Index-Seite der Online-Hilfe.
- NET_CMD_LISTFILES - Kommando, auf das der Server eine Liste von Dateien zurückgibt.
- NET_CMD_GETPROPERTIES - Kommando, auf das der Server die Eigenschaften einer Datei zurückliefert.
- NET_CMD_GETFILE - Befehl für den Server, eine Datei zu übertragen.

- NET_CMD_GNUPLOT - Befehl, der auf dem Server das Erstellen einer Bild-datei zu einer AnT-Datei zur Folge hat.
- NET_OPT_RECURSIVE - Befehlsoption, die angibt, dass der Befehl re-kursiv ausgeführt werden soll.
- NET_OPT_ASCII - Befehlsoption, die bewirkt, dass die angeforderten Da-ten als ASCII-Text gesendet werden.
- NET_OPT_PROPERITES - Befehlsoption, die den Server veranlasst, die Eigenschaften der jeweiligen Datei zuerst zu senden.
- NET_OPT_BINARY - Option eines Serverbefehls, die festlegt, dass die Da-ten in binärer Form gesendet werden sollen.
- NET_DATAFILES -Zeichenkette mit dem Datei-Suffix von AnT-Dateien.
- NET_CID_GETCOMMENT - Befehls-ID für den internen Befehl, um Kom-mentare zu lesen.
- NET_CID_GETIMAGE - Interne Befehls-ID, um ein Bild vom Server an-zufordern.
- NET_CID_GNUPLOT - Befehls-ID, um ein Bild auf dem Server erstellen zu lassen.
- GUI_WIDTH - Wenn nicht anders in der HTML- oder JNLP-Datei angege-ben, wird die Anwendung beim Start auf diese Breite gebracht.
- GUI_HEIGHT - Wenn nicht anders in der HTML- oder JNLP-Datei ange-geben, wird die Anwendung beim Start auf diese Höhe gebracht.
- GUI_DEF_DYNAMIC_BG - Hintergrundfarbe für den interaktiven 3D- Be-reich, in dem die Punktwolke dargestellt wird.
- GUI_DEF_EDGE_COLOR - Standardfarbe von Linien, wie sie zum Bei-spiel für das Koordinatensystem verwendet werden.
- GUI_DEF_POINT_COLOR - Standardfarbe der Punkte. Gilt nur, wenn die Färbungsart „Einfarbig“ gewählt wird.
- GUI_DEF_FONT_COLOR - Standardfarbe der Schrift innerhalb des 3D-Bereichs, zum Beispiel für die Achsenbeschriftung.
- GUI_DEF_GRIDCOUNT - Standardwert für die Anzahl der Teilungen bei „Helper“-Objekten im 3D-Bereich, zum Beispiel Anzahl der Teilstriche beim Koordinatensystem.

- GUI_DEF_GRIDCOUNT_MIN - Minimum für die Anzahl der Teilungen bei „Helper“-Objekten im 3D-Bereich.
- GUI_DEF_GRIDCOUNT_MAX - Maximum für die Anzahl der Teilungen bei „Helper“-Objekten im 3D-Bereich.
- GUI_POINT_TIME - Standardzeit zwischen dem Erscheinen zweier Punkte bei der Animation in der dynamischen Visualisierung in Millisekunden.
- GUI_DEF_AZIMUTH - Angabe in Grad für den Standardwert des „Azimutwinkels“, unter dem die Punktwolke angezeigt wird.
- GUI_DEF_ELEVATION - Angabe in Grad für den Standardwert des „Höhenwinkels“, unter dem die Punktwolke angezeigt wird.
- GUI_DEF_ZOOM - Standardwert für den Zoom, mit dem die Punktwolke angezeigt wird.
- GUI_DEF_ZOOMMUL - Stufe, um den der Zoom pro Mausbewegung verändert wird.
- GUI_DRAWING_STYLES - Gibt die Bezeichnungen für die verschiedenen Zeichenarten der Punktwolke an.
- GUI_GNUPLOT_STYLES - Zeichenkettenfeld mit den verschiedenen Linienarten, die von der Gnuplot-Schnittstelle verstanden werden.
- GUI_COLOR_SCHEMES - Namen der verschiedenen Färbungsmöglichkeiten für die Punktwolke.
- GUI_DEF_IN_BORDER - Farbe und Linienart für innenliegende Rahmen in der GUI.
- GUI_DEF_OUT_BORDER - Farbe und Linienart für außenliegende Rahmen in der GUI.
- TXT_TAB_SELECT - Name des Auswahl-Tabs in der GUI (erstes Tab mit dem Auswahlbaum).
- TXT_TAB_HELP - Name des Tabs in der GUI, in der die Online-Hilfe dargestellt wird.
- TXT_TAB_TRANSFER - Name des Tabs in der GUI, in dem der Zustand aller Verbindungen angezeigt wird.
- TXT_NO_DESCR - Text, der angezeigt wird, wenn zu einem im Auswahlbaum selektierten Objekt keine Beschreibung verfügbar ist, weil es sich um einen Verzeichnisknoten handelt.

- TXT_NO_IMAGE - Text, der bei einem Fehler beim Laden eines Bildes in der GUI angezeigt wird.
- TXT_NO_DATAFILE - Fehlermeldung, die erscheint, wenn eine Visualisierung gestartet werden soll, aber keine Datei ausgewählt ist.
- TXT_SELECT_COLS - Beschriftung des Buttons zur Bestätigung der Auswahl der in der 3D-Ansicht darzustellenden Spalten einer AnT-Datei.
- TXT_TRANSFER_STOPPED - Beschriftung des Stop-Buttons, wenn ein Datentransfer abgebrochen wurde.
- TXT_NO_FILEDESC - Text, der angezeigt wird, wenn zu einem im Auswahlbaum selektierten Objekt keine Beschreibung auf dem Server verfügbar ist.
- TXT_AXIS - Bezeichnung für eine Achse in der GUI.
- TXT_STYLE - Bezeichnung für einen Stil oder eine Darstellungsart.
- TXT_AUTORANGE - Text zu der Checkbox in der statischen Visualisierung, mit der angegeben wird, dass Gnuplot automatisch die Grenzen für die Achsen zur Visualisierung setzen soll.
- TXT_COL_SELECTION - Überschrift der Bereiche in der GUI, die für die Auswahl der Spalten zuständig sind.
- TXT_ANIMATION_CONTOL - Überschrift des Bereichs in der GUI, der für die Steuerung der Animation verantwortlich ist.
- TXT_COLOR_CONTOL - Überschrift des Bereichs in der GUI, der für die Färbung der Punktwolke zuständig ist.
- TXT_DRAWING_CONTOL - Überschrift des Bereichs in der GUI, der für die Auswahl des Zeichenstils zuständig ist.
- TXT_HELPERS_CONTOL - Überschrift des Bereichs in der GUI, der für die Auswahl der „Helper“ gedacht ist.
- TXT_STEP_CONTOL - Überschrift des Bereichs in der GUI, der für die Verfeinerung der Gitterabstände der „Helper“ zuständig ist.
- TXT_DELAY - Beschriftung des Feldes für die Eingabe der Verzögerungszeit zwischen dem Einblenden zweier Punkte bei der Animation.
- TXT_OF - Zeichen in der Animationskontrolle zwischen aktueller Position und bereits übertragenen Punkten.

- `TXT_INIT_DESCR_TEXT` - Text, der in dem Beschreibungsfeld des Auswahl-Tabs angezeigt wird, wenn noch kein Knoten des Baumes selektiert wurde.
- `TXT_LAUNCHER_BUTTON` - Beschriftung des Buttons im Auswahl-Tab, mit dem ein neuer Visualisierungs-Tab geöffnet wird.
- `TXT_GNUPLOTT_BUTTON` - Beschriftung des Buttons in der statischen Visualisierung, mit der ein neuer Gnuplot-Auftrag an den Server gesendet wird.
- `TXT_LOADING_FILELIST` - Text, der angezeigt wird, bevor eine Liste der verfügbaren AnT-Dateien übertragen wird.
- `TXT_LOADING_DESC` - Text, der angezeigt wird, solange eine Beschreibung zu einem gewählten Objekt im Auswahl-Tab geladen wird.
- `TXT_LOADING_FILELIST_WAIT` - Text, der angezeigt wird, solange die Liste der verfügbaren AnT-Dateien übertragen wird.
- `TXT_LAUNCH_CHOOSE` - Fehlermeldung für den Fall, dass im Auswahl-Tab eine Visualisierung gestartet werden sollte, aber noch keine Visualisierungsart ausgewählt wurde.
- `TXT_LAUNCH_DYNAMIC` - Beschreibung im Auswahlfeld des Auswahl-Tab für den Start einer dynamischen Visualisierung.
- `TXT_LAUNCH_STATIC` - Beschreibung im Auswahlfeld des Auswahl-Tab für den Start einer statischen Visualisierung.
- `TXT_LAUNCH_FILE` - Beschreibung im Auswahlfeld des Auswahl-Tab für den Start einer Dateiansicht.
- `TXT_HELPER_BB` - Name des Helpers „Bounding Box“, also der umschließenden Box.
- `TXT_HELPER_KO` - Name des Helpers „Coordinate System“, also des Koordinatensystems.
- `TXT_HELPER_XZPLANE` - Name des Helpers „Plane XZ“, also der xz-Ebene.
- `TXT_HELPER_XYPLANE` - Name des Helpers „Plane XY“, also der xy-Ebene.
- `TXT_HELPER_YZPLANE` - Name des Helpers „Plane YZ“, also der yz-Ebene.

- `TXT_HELPER_GRIDBOX` - Name des Helpers „Gridbox“, also der Gitterbox.
- `TXT_COLUMN_NAME` - Name einer Spalte für den Fall, dass in der AnT-Datei keine Namen angegeben wurden. Es wird dann immer eine Nummer angehängt.
- `TXT_PLZ_WAIT` - Allgemeiner Text, wenn ein Objekt geladen wird.
- `TXT_NONE_ROW` - Zeichenkette, die angezeigt werden soll, wenn keine Spalte selektiert ist.
- `TXT_NAN` - Allgemeine Fehlermeldung, die angezeigt wird, wenn in ein Feld für numerische Werte keine Nummer eingegeben wurde.
- `TXT_SELECT_SUBMIT` - Text, der in der statischen Visualisierung angezeigt wird, wenn noch kein Bild vom Server geladen wurde.
- `TXT_TRANSFER_STOP` - Beschriftung der Buttons im Transfer-Panel zum Abbruch einer Datenübertragung.
- `TXT_TRANSFER_DONE` - Beschriftung der Buttons im Transfer-Panel, wenn eine Datenübertragung abgeschlossen ist.
- `TXT_PARAMETER_RESET` - Beschriftung des Buttons in der dynamischen Visualisierung, um die ursprüngliche Ansicht auf die Punktwolke wiederherzustellen.
- `TXT_CENTER_ORIGIN` - Beschriftung des Buttons in der dynamischen Darstellung, mit dem die Punktwolke in den Mittelpunkt verschoben und auf maximale Größe skaliert wird.
- `TXT_RESTORE_ORIGIN` - Beschriftung des Buttons in der dynamischen Darstellung, mit dem die Verschiebung und Skalierung der Punktwolke aufgehoben werden kann.
- `TXT_HELPER_SELECT` - Beschriftung für den Bereich in dynamischen Visualisierungen, in dem die „Helper“ ausgewählt werden können.
- `TXT_HELP_UNAVAILABLE` - Text, der angezeigt wird, falls unter der angegebenen URL (`NET_HELP_PAGE_URL`) keine Hilfeseiten verfügbar sind.

6.5 Ergebnisse

6.5.1 Statistiken

Bei der Implementierung von AnT-Online sind insgesamt über 11200 Zeilen an Quellcode in Java-Klassendateien entstanden. Der durchschnittliche Anteil von Kommentaren im Quellcode liegt dabei bei etwas über 30 Prozent.

Bei der Erstellung der Prototypen sowie der Technologie-Studien sind nochmals über 5200 Zeilen in Java-Klassendateien entstanden, wobei hier jedoch kaum kommentiert wurde.

Insgesamt sind damit im Verlauf dieser Diplomarbeit knapp 17000 Zeilen Java-Quellcode entstanden. Hierzu kommt noch verschiedener Quellcode der Prototypen in Java3D, Shockwave, Flash und VRML.

6.5.2 Dokumente

Neben dieser Ausarbeitung sind im Verlauf der Arbeit zusätzlich verschiedene Dokumente entstanden. Unter anderem eine sehr ausführliche JavaDoc- Dokumentation, verschiedene Spezifikations- und Entwurfsdokumente, die Online-Hilfe, die Website mit der Bewertung der Technologien sowie eine umfangreiche Link- und Literatursammlung.

6.5.3 Applet/Java Web Start-Applikation

Der folgende Abschnitt gibt einen kurzen Überblick über die entstandene grafische Benutzungsoberfläche.

Selection-Panel

Abbildung 6.13 zeigt den weiter oben beschriebenen Auswahldialog, in dem in einer Baumstruktur alle verfügbaren AnT-Dateien und, falls vorhanden, Beschreibungen zu den Verzeichnissen und Dateien dargestellt werden.

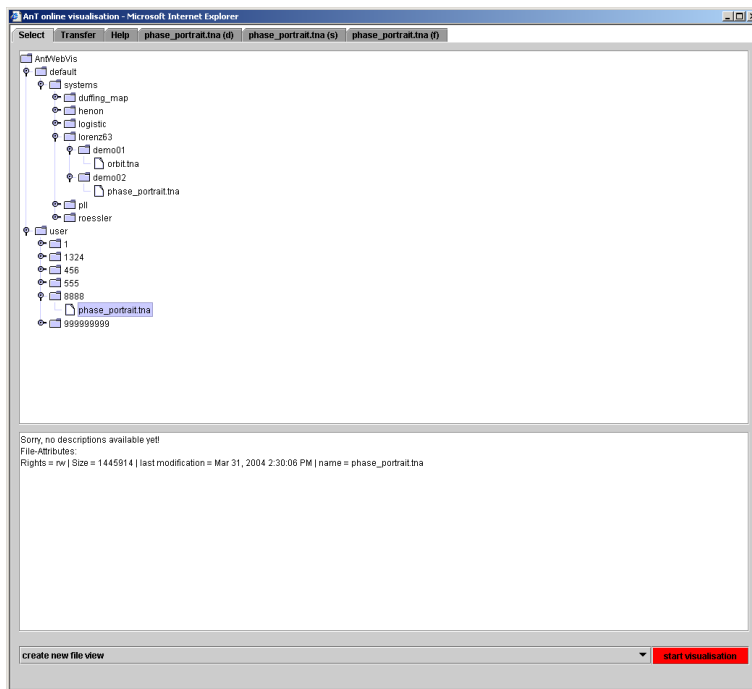


Abbildung 6.13: Screenshot des fertigen Selection-Panels

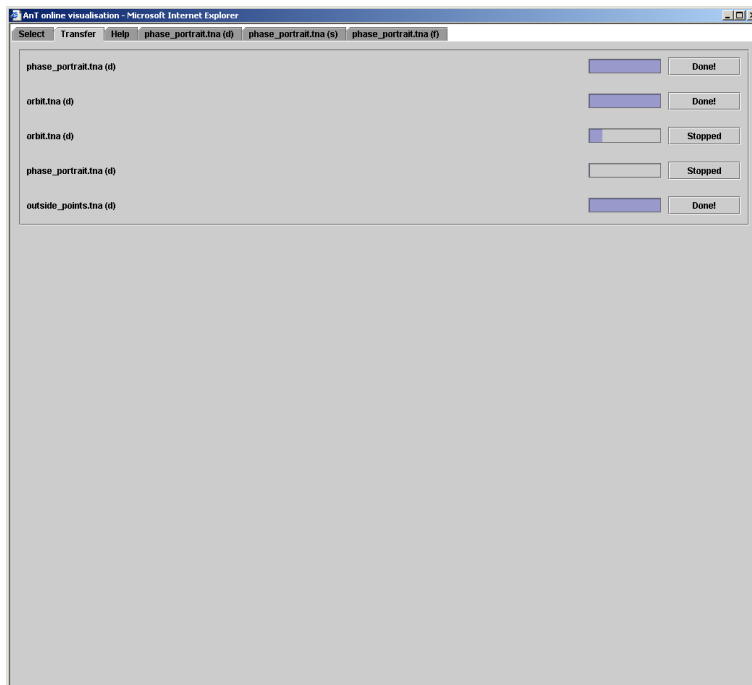


Abbildung 6.14: Screenshot des fertigen Transfer-Panels

Transfer-Panel

Die konkrete Ausprägung des Dialogs zur Übersicht der laufenden Dateiübertragungen zeigt die Abbildung 6.14.

Help-Panel

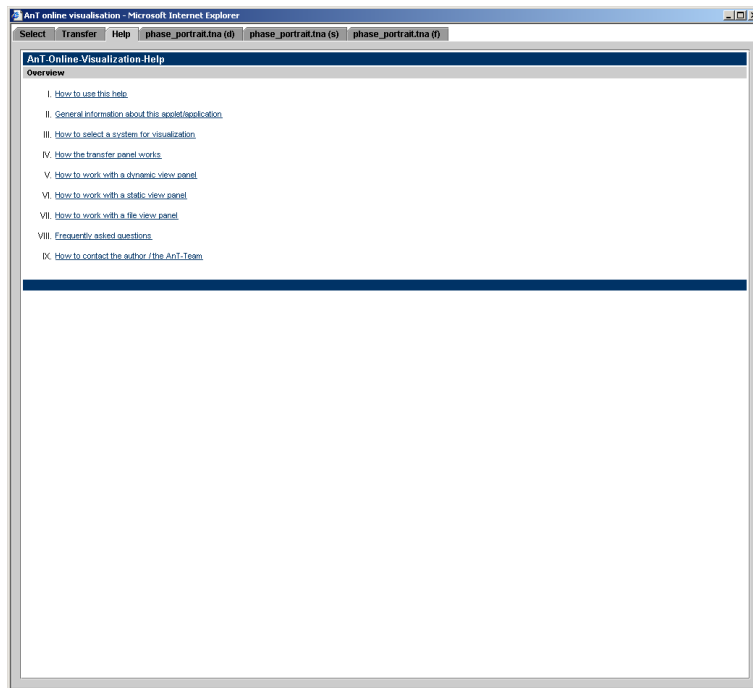


Abbildung 6.15: Screenshot des fertigen Help-Panels

Den Bereich des Applets/der Java Web Start-Applikation, der die Anzeige der Online-Hilfe in Form von HTML-Dokumenten, ähnlich eines üblichen Browsers, übernimmt, zeigt Abbildung 6.15.

3D-Panel

Abbildung 6.16 zeigt den Dialog zur dynamischen, dreidimensionalen Visualisierung.

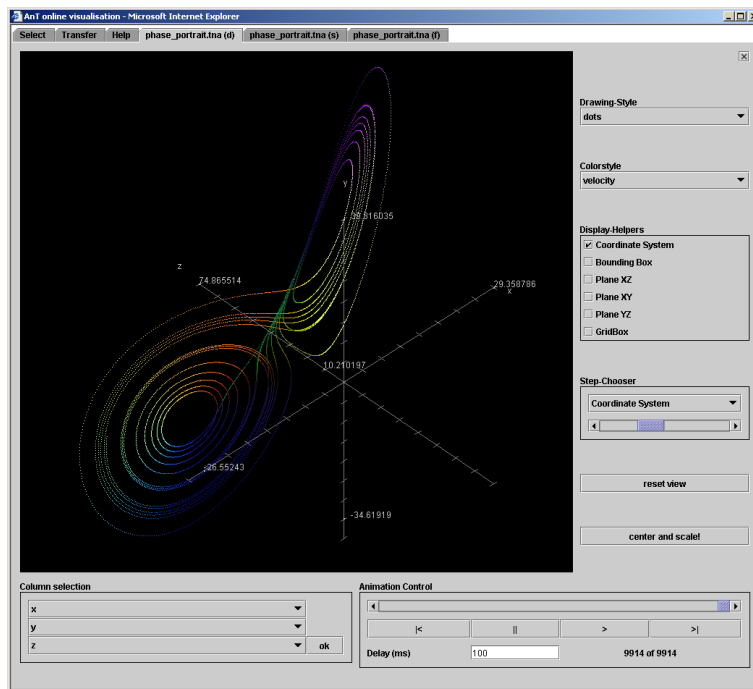


Abbildung 6.16: Screenshot des fertigen 3D-Panels

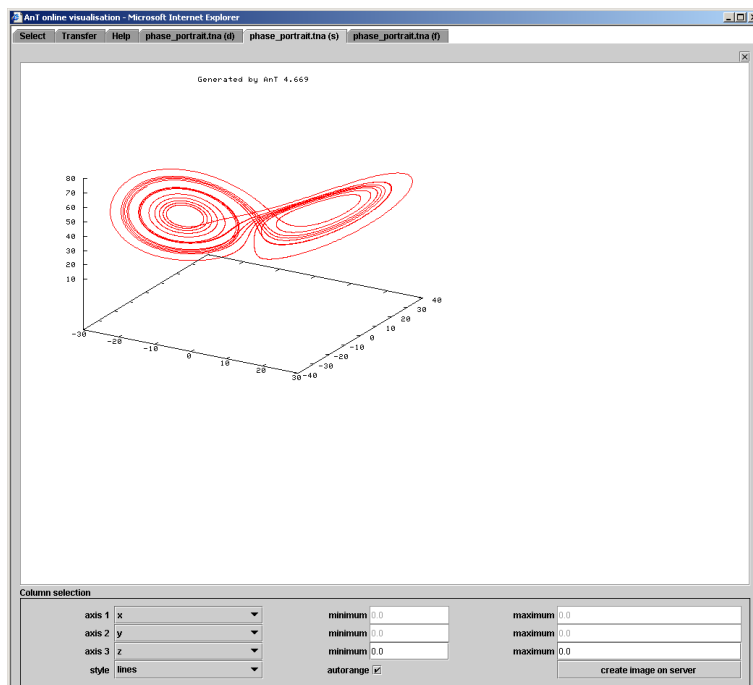
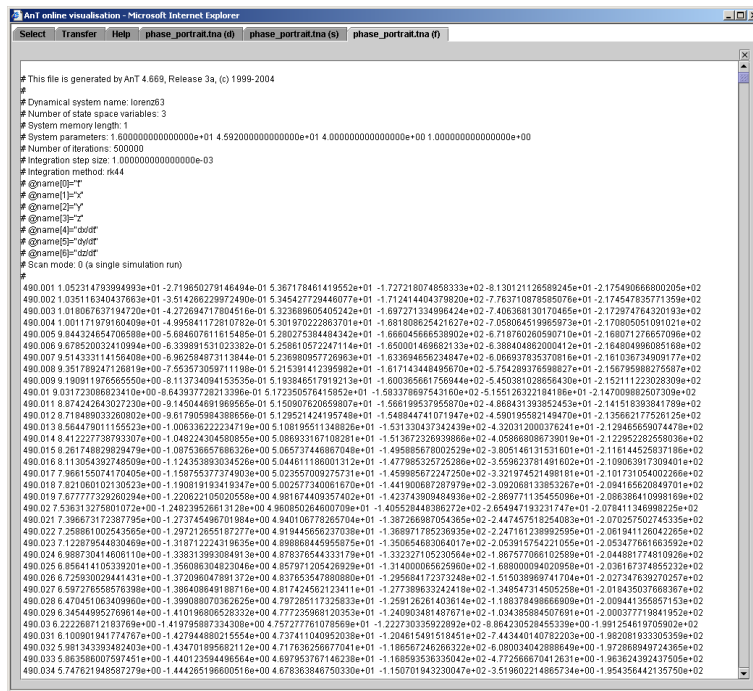


Abbildung 6.17: Screenshot des fertigen Image-Panels

Image-Panel

Den entstandenen Dialog zur statischen, zweidimensionalen Visualisierung zeigt Abbildung 6.17.

File-Panel



```
ANT online visualisation - Microsoft Internet Explorer
Select Transfer Help phase_portrait.tna (d) phase_portrait.tna (s) phase_portrait.tna (f)

# This file is generated by ANt 4.669, Release 3a, (c) 1999-2004
#
# Dynamical system name: lorenzE3
# Number of state space variables: 3
# System memory length: 1
# System parameters: 1.0000000000000000e+01 4.5920000000000000e+01 0.0000000000000000e+00 1.0000000000000000e+00
# Number of iterations: 500000
# Integration step size: 1.0000000000000000e-03
# Integration method: rk44
#
#@name[0]="T"
#@name[1]="x"
#@name[2]="y"
#@name[3]="z"
#@name[4]="dz/dt"
#@name[5]="dy/dt"
#@name[6]="dz/dt"
#
# Scan mode: 0 (a single simulation run)
#
490.001 1.052314793994993e+01 -2.719650279146494e-01 5.367178461419552e+01 -1.727218074858333e+02 -8.130121126589245e+01 -2.175490666800205e+02
490.002 1.035116340437693e+01 -3.514266229972490e-01 5.345427729446077e+01 -1.712414403798200e+02 -7.763710878958076e+01 -2.17454763571359e+02
490.003 1.018007637194720e+01 -4.272694717604516e-01 5.322686905405243e+01 -1.697271334696424e+02 -7.406389130170456e+01 -2.172974745420193e+02
490.004 1.001119719104049e+01 -4.995841172810782e-01 5.301970222863701e+01 -1.681808625421627e+02 -7.058064519965973e+01 -2.170805051091021e+02
490.005 9.844324654708598e+00 -5.684807611615485e-01 5.280275384484324e+01 -1.666045666399020e+02 -6.719760260590710e+01 -2.168071276657098e+02
490.006 9.67952002410994e+00 -6.33991531023382e-01 5.259810527247114e+01 -1.650001469828133e+02 -6.388404620300412e+01 -2.1646049680605109e+02
490.007 9.514333141564038e+00 -6.982594873118044e-01 5.238880867726383e+01 -1.633646562348474e+02 -6.066397935370816e+01 -2.1610387349001177e+02
490.008 9.351789247128191e+00 -7.553573059711198e-01 5.215391412395982e+01 -1.617143448495670e+02 -5.754289376598827e+01 -2.156785988275507e+02
490.009 9.190911976565505e+00 -8.113734094153525e-01 5.192846517919213e+01 -1.600365661756944e+02 -5.450381028656430e+01 -2.152111223028309e+02
490.01 9.03172208622410e+00 -8.64393772312396e-01 5.17230576415052e+01 -1.583378697543100e+02 -5.15513832194186e+01 -2.14700602507309e+02
490.011 8.874242643027230e+00 -9.145044691968556e-01 5.150907620659807e+01 -1.566199537955870e+02 -4.868431393852453e+01 -2.141518393841789e+02
490.012 8.71848903208002e+00 -9.617905984388566e-01 5.129521424195749e+01 -1.548844741071947e+02 -4.58019582148470e+01 -2.135662177526125e+02
490.013 8.56447901155523e+00 -1.00633622234719e+00 5.10819511348826e+01 -1.531330437342439e+02 -4.320312000376241e+01 -2.129465659074478e+02
490.014 8.41222738703307e+00 -1.0492349450856e+00 5.08693167109281e+01 -1.51397239839896e+02 -4.06966069739019e+01 -2.1235229259038e+02
490.015 8.261746829829479e+00 -1.097536657686326e+00 5.065737446887048e+01 -1.496886678002529e+02 -3.805146131531601e+01 -2.116144525837186e+02
490.016 8.113054392748509e+00 -1.124353893034526e+00 5.044611186001312e+01 -1.47798532725288e+02 -3.559623781491602e+01 -2.109639173094010e+02
490.017 7.966155874170405e+00 -1.15978537374903e+00 5.023507009275731e+01 -1.45989567247250e+02 -3.321974521490191e+01 -2.10173154002266e+02
490.018 7.820160102130523e+00 -1.190819193419347e+00 5.002577340061670e+01 -1.441800687287979e+02 -3.092068133653267e+01 -2.094165620849701e+02
490.019 7.67777329280294e+00 -1.220822105020558e+00 4.981674409357402e+01 -1.423742809484938e+02 -2.869771135450066e+01 -2.086386410998169e+02
490.02 7.53631275801072e+00 -1.249239526612128e+00 4.960805264600709e+01 -1.405528448389272e+02 -2.654947192321747e+01 -2.07841346998225e+02
490.021 7.3967312367705e+00 -1.273745498701984e+00 4.94016878265704e+01 -1.387266897054395e+02 -2.447457618244093e+01 -2.07025702745255e+02
490.022 7.25881002543656e+00 -1.297212655187277e+00 4.919445656237038e+01 -1.368971785238935e+02 -2.247161238992959e+01 -2.061941126042265e+02
490.023 7.122879544803408e+00 -1.318712224319635e+00 4.89888644595875e+01 -1.350654683064017e+02 -2.053915754221055e+01 -2.053477661663592e+02
490.024 6.989730414606110e+00 -1.338139830849139e+00 4.878370544333179e+01 -1.332327105239504e+02 -1.867577086102589e+01 -2.044881774810920e+02
490.025 6.85641419539201e+00 -1.356086304923046e+00 4.857871205426928e+01 -1.31400006925389e+02 -1.68800094020959e+01 -2.036167274455232e+02
490.026 6.72590020441431e+00 -1.372086047891372e+00 4.837653547880800e+01 -1.295684172373248e+02 -1.515038969741704e+01 -2.027347639270257e+02
490.027 6.59727658579398e+00 -1.386408849188716e+00 4.817424562123411e+01 -1.27738633242418e+02 -1.348547314605258e+01 -2.018435037688367e+02
490.028 6.47045108349980e+00 -1.39980870362625e+00 4.79729511725323e+01 -1.25912626140314e+02 -1.18037849886809e+01 -2.00944155657152e+02
490.029 6.34544992769914e+00 -1.410198808528332e+00 4.777235966120353e+01 -1.240803481487871e+02 -1.03438594807691e+01 -2.00037719841952e+02
490.03 6.22286712183769e+00 -1.41979589733408e+00 4.75727761078568e+01 -1.222730335922892e+02 -8.86223052845338e+00 -1.991254619705902e+02
490.031 6.100801941774876e+00 -1.427944860215554e+00 4.737411040952038e+01 -1.204615491518451e+02 -7.443440140782203e+00 -1.982081933305359e+02
490.032 5.98134324034037e+00 -1.434701895662112e+00 4.717830256977841e+01 -1.18656724626932e+02 -6.00003402086849e+00 -1.972868849724355e+02
490.033 5.8635686007597451e+00 -1.440123594496564e+00 4.697953767146238e+01 -1.16859353835042e+02 -4.772566870412631e+00 -1.963624392437505e+02
490.034 5.747621948587279e+00 -1.444265196600516e+00 4.678363846750330e+01 -1.150701843230047e+02 -3.519602214865734e+00 -1.954356442135750e+02
```

Abbildung 6.18: Screenshot des fertigen File-Panels

Der letzte Dialog, der lediglich dazu dient, die ANt-Dateien direkt anzuzeigen, wird in Abbildung 6.18 dargestellt.

Kapitel 7

Evaluation

7.1 Einleitung

Die Evaluation der Software soll als Erfolgskontrolle der bei der Erstellung der Anwendung ergriffenen softwareergonomischen Maßnahmen dienen.

Im Rahmen dieser Diplomarbeit kann die Evaluation der entstandenen Software nur einen relativ kleinen Umfang annehmen, da andere Themen im Vordergrund standen.

Die Evaluation steht in dieser Arbeit aus zeitlichen Gründen ganz am Ende, sodass die Ergebnisse der Evaluation nicht mehr in vollem Umfang in eine neue Iteration zur Überarbeitung der Software eingehen können. Dies gilt in besonderem Maße, da die Evaluation des Applets/der Applikation mit Ende der Diplomarbeit nicht abgeschlossen sein wird.

Der erarbeitete Online-Fragebogen mit der entsprechenden Datenbank und den Auswertungswerkzeugen ist auch über die Diplomarbeit hinaus aktiv, damit Benutzer aus dem Internet weiterhin die Möglichkeit haben, konstruktiv an einer möglichen Weiterentwicklung des Applets/der Applikation mitzuwirken.

Aus dieser Situation ergeben sich für die Evaluation einige Einschränkungen und Kompromisse, besonders im Umfang des Fragebogens, da nur eine gewisse Anzahl der Tests durch einen Testleiter betreut werden können, was aber bei anonymen Benutzern aus dem Internet nicht möglich ist.

7.2 Grundlagen der Evaluation

7.2.1 Performance der GUI

Da bei der Erstellung der Software nicht auf hardwarebeschleunigte Visualisierungstechnologien wie OpenGL oder DirectX zurückgegriffen werden konnte, spielte die Performance und Leistungsoptimierung der GUI, besonders des Software-3D-Renderings, eine große Rolle, um die Hardwareanforderungen an den Client-Rechner möglichst gering zu halten und damit das Applet/die Applikation einem möglichst großen Publikum verfügbar zu machen.

Aus diesem Grund wird bei der Evaluation genauer auf die Hard- und Software-Ausstattung des Benutzers eingegangen, um eine Vorstellung über die durchschnittliche Ausstattung der Anwender zu gewinnen (siehe Abbildung E.4).

Andere Fragen zielen dabei auch auf den subjektiven Performance-Eindruck der Benutzer ab, um eine Vorstellung zu gewinnen, ab welcher Ausstattung ein Benutzer üblicherweise die Performance als ausreichend betrachtet (siehe Abbildung E.5).

7.2.2 Performance der Kommunikation mit dem Server

Neben der Performance der GUI spielt bei einer Online-Visualisierung natürlich auch die Geschwindigkeit der Kommunikation mit dem Server, also Übertragen und Anzeigen von Beschreibungen, Übertragen von einer Datei beziehungsweise mehrerer Dateien, Generieren und Transferieren der statischen Ansichten und so weiter, eine große Rolle. Hier soll die Evaluation zeigen, ob eine Kompression des Datenstroms oder eine Veränderung der Übertragungsstrategien erforderlich sein könnte.

7.2.3 ISO 9241: Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten, Teil 10: Grundsätze der Dialoggestaltung

Als allgemein gültige und international anerkannte Grundlage zur Beurteilung der Qualität von Dialogsystemen kann die Norm ISO 9241 herangezogen werden. Besonders ist hierbei der Teil 10: Grundsätze der Dialoggestaltung zu nennen (siehe dazu auch [59], [57] und [58]).

Im Folgenden werden kurz die sieben Kriterien des Standards ISO 9241-10 aufgeführt und kurz erläutert, da die Evaluation neben den beiden Fragenkomplexen zur Performance und Vorwissen des Benutzers im Wesentlichen als Prüfung der Ergonomie anhand dieser Kriterien durchgeführt wird.

Aufgabenangemessenheit

Ein Dialog ist aufgabenangemessen, wenn er den Benutzer bei allen Aufgaben, die der Benutzer über den betreffenden Dialog erledigen will, unterstützt und tatsächlich hilfreich ist. Der Dialog soll die Arbeit nicht unnötig erschweren, sie nicht komplizierter oder umständlicher machen, als sie sein muss (siehe auch Abbildung E.5).

Selbstbeschreibungsfähigkeit

Ein Dialog ist selbstbeschreibungsfähig, wenn die Benutzung selbsterklärend ist. Das bedeutet, jeder Dialogschritt ist unmittelbar verständlich. Dem Benutzer ist klar, was die Anwendung tut, was als nächste Eingabe oder Reaktion erwartet wird, und was die Folgen der Interaktion sein werden. Wo dies nicht in vollem Umfang gegeben ist, muss alles, was dem Benutzer unklar ist, auf Anfrage hin erklärt werden, das heißt, es muss eine Hilfe verfügbar sein (siehe auch Abbildung E.6).

Erwartungskonformität

Ein Dialog ist erwartungskonform, wenn er konsistent zu den bisher vom Benutzer in seinem Arbeitsbereich durch Ausbildung und Erfahrung kennengelernten Dialogen ist. Es trägt ebenfalls zur Erwartungskonformität bei, wenn sich ein Dialog auch an allgemein gültige und anerkannte Konventionen hält (siehe auch Abbildung E.7).

Fehlerrobustheit

Ein Dialog ist fehlerrobust, wenn das angestrebte Arbeitsergebnis trotz falscher Benutzereingaben entweder leicht und mit minimalem Aufwand korrigierbar ist oder sogar durch den Dialog selbst korrigiert wird, falls dies möglich ist. Der Dialog muss den Benutzer auch von falschen Eingaben unterrichten und soll möglichst hilfreiche Korrekturvorschläge machen (siehe auch Abbildung E.8).

Lernförderlichkeit

Ein Dialog ist lernförderlich, wenn er den Benutzer beim Erlernen des Dialogsystems unterstützt und anleitet (siehe auch Abbildung E.9).

Steuerbarkeit

Ein Dialog ist steuerbar, wenn es dem Benutzer möglich ist, auf den Dialogablauf Einfluss zu nehmen. Der Benutzer soll Reihenfolge und Geschwindigkeit beeinflussen können. Es muss zudem möglich sein, dass der Benutzer in der Abarbeitung eines Teildialogs pausiert, um den Teildialog zu einem späteren Zeitpunkt wieder aufnehmen zu können, ohne die bisherigen Daten zu verlieren (siehe auch Abbildung E.10).

Individualisierbarkeit

Ein Computerprogramm ist individualisierbar, wenn die Dialoge des Systems individuell an die Bedürfnisse des Benutzers und an die Erfordernisse der Arbeitsaufgabe angepasst werden können. Die Individualisierung soll allerdings begrenzt sein, sodass der Benutzer keine schädlichen Einstellungen vornehmen kann. Die Individualisierung kann auch kein Ersatz für ergonomisch gestaltete Dialoge sein (siehe auch Abbildung E.11).

7.3 Vorbereitung der Evaluation

Bei der Vorbereitung einer Software-Evaluation müssen verschiedene grundsätzliche Fragestellungen beachtet werden. Im Allgemeinen müssen die folgenden Punkte geklärt werden.

7.3.1 Was soll mit der Evaluation erreicht werden?

Da die Software als Applet oder Applikation theoretisch von jedem Internet-Benutzer verwendet werden kann, ist bei der Erstellung besonderer Wert auf einfache und komfortable Handhabung gelegt worden.

Es sind auch bekannte Konzepte von in der Bedienung vergleichbaren Programmen wie Maple, Mathematica, 3D Studio Max, Maya und so weiter eingeflossen. Die Evaluation soll zeigen, wie erfolgreich diese Bemühungen waren.

Besonders soll auch nach Ressel [59] aus der Evaluation hervorgehen,

- ob noch zusätzliche Funktionen von den Benutzern gewünscht werden,
- ob es besondere Problemstellen in den Dialogen gibt,
- ob die Struktur ausreichend klar und verständlich ist,
- ob die Dialoge klar und verständlich sind und sich der Funktionsumfang und die Bedienung automatisch erschließen.

7.3.2 Wann und wo findet der Test statt?

Der Test soll unter möglichst realistischen Bedingungen vorgenommen werden. Deswegen erfolgt auch der betreute Teil der Tests an den jeweiligen Arbeitsplatzrechnern der Teilnehmer.

7.3.3 Wie lange dauert eine Testsitzung?

Eine Testsitzung sollte nicht länger als etwa 15 Minuten im betreuten Teil (Einführung, Bearbeitung der drei Aufgaben wie in Kapitel 7.4.2 beschrieben) und 5 bis 10 Minuten im selbstständig zu bearbeitenden Teil (Online-Fragebogen) dauern.

7.3.4 Welche Hard- und Software wird benötigt?

Da der Test nur einen Browser mit lauffähiger Java 1.2-Umgebung voraussetzt und am Arbeitsplatz der Testteilnehmer stattfinden soll, wird kein Testlabor und dementsprechend auch keine weitere Hard- und Software benötigt.

7.3.5 Von welchem Anfangszustand des Systems soll ausgegangen werden?

Ausgangszustand ist in diesem Fall ein Browser, in dem die Startseite für das Applet/die Applikation aufgerufen wurde, sowie ein laufender Web- und AnTon-Server.

7.3.6 Wie stark soll das System im Test ausgelastet sein, und wie ist dies zu erreichen?

Da zum Zeitpunkt des Tests leider noch nicht die endgültige Konfiguration (eigener Server an der Uni Stuttgart) verfügbar ist, kann auch die Auslastung der beteiligten Server nicht simuliert werden. Da der endgültige Server und dessen Anbindung um Größenordnungen besser sein werden (Sun-Server mit mehreren Prozessoren, statt Celeron 300, 10Mbit/s statt 128kbit/s, und so weiter), sollte der Test unter den gegebenen Bedingungen etwa einem mittelstark ausgelasteten endgültigen System entsprechen.

7.3.7 Wer ist der Testleiter?

Testleiter ist, zumindest in den betreuten Tests, der Autor dieser Diplomarbeit. Da die Evaluation auch nach der Beendigung der Diplomarbeit mittels eines Online-Fragebogens fortgesetzt werden soll, ist bei diesen freiwilligen Tests von Benutzern natürlich eine Betreuung oder Leitung hier nicht mehr möglich.

7.3.8 Wer sind die Testbenutzer?

Als Testbenutzer dienen einige Mitarbeiter der Abteilung sowie einige weitere Studenten der Softwaretechnik, Informatik oder Medieninformatik. Dabei wird immer unterschieden, ob der Testkandidat bereits mit AnT und/oder der bestehenden Visualisierung vertraut ist oder nicht. Dies ist nötig, um die Ergebnisse nicht zu verfälschen, da der überwiegenden Mehrheit der Nutzer des Applets/der Applikation AnT noch nicht bekannt sein wird.

Andererseits ist auch die Sicht der Benutzer wichtig, die bereits mit AnT gearbeitet haben, da diese eventuell Vor- und Nachteile zur bestehenden AnT-Visualisierung aufzeigen kann.

7.3.9 Wie viele Testbenutzer werden benötigt?

In der Literatur (siehe auch [57] oder [59]) wird von mindestens 5-15 Testbenutzern ausgegangen, wobei andere Untersuchungen ergeben haben, dass bereits 5 Testbenutzer ausreichend sind [72]. Durch diese Studie wurde festgestellt, dass

sich folgender Zusammenhang ergibt:

$$N(1 - (1 - L)^n)$$

Dabei ist N die Summe der Probleme im Design der Benutzungsoberfläche, n die Anzahl der Tester und L die durchschnittliche Menge von Fehlern ($L = 31\%$ als Durchschnittswert vieler untersuchter Fälle), die bei Tests mit nur einem Benutzer aufgedeckt werden können.

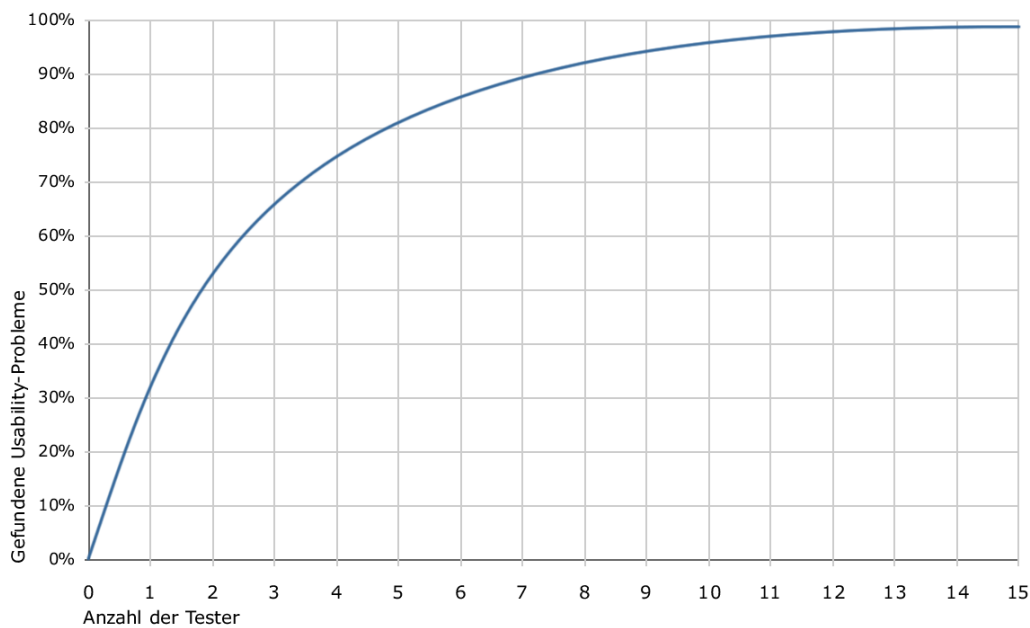


Abbildung 7.1: Gefundene Probleme im Verhältnis zur Testeranzahl

Dieser Zusammenhang wird auch in Abbildung 7.1 dargestellt.

Natürlich werden die erhobenen Daten aussagekräftiger, je mehr Teilnehmer an der Evaluation teilnehmen. Wie Abbildung 7.1 aber zeigt, bringt der Aufwand von 10 oder mehr Teilnehmern nur noch geringe Vorteile bei der Aufdeckung von Problemen, weswegen die Zahl der Testpersonen im Rahmen dieser Diplomarbeit aus Zeit- und Aufwandsgründen zumindest im betreuten Teil des Tests auf etwa 3-6 Teilnehmer pro Gruppe beschränkt bleibt.

7.3.10 Welche Aufgaben werden gestellt?

Aufgaben können natürlich nur in den betreuten Tests gestellt werden. Details zu den Aufgaben für die Bereiche

- Dynamische Visualisierung
- Statische Visualisierung
- Hilfe

finden Sie in Kapitel 7.4.2

7.3.11 Welche Hilfsmittel haben die Benutzer, und in welchem Umfang ist Hilfestellung erlaubt?

Um den Test möglichst realistisch durchzuführen, steht dem Testteilnehmer nur die Online-Hilfe des Applets/der Applikation zur Verfügung. Der Testleiter darf keine Hilfestellungen geben, da dies beim Anwendungsfall über das Internet auch nicht oder nur sehr eingeschränkt, zum Beispiel per Email, möglich wäre.

7.3.12 Welche Daten werden gesammelt, und wie werden diese analysiert?

Daten werden hauptsächlich über den Online-Fragebogen in Form von Multiple-Choice-Fragen und Freitextfeldern gesammelt und in einer Datenbank gespeichert. Die Analyse erfolgt dann über Scripte, die auf diese Datenbank zugreifen. Nur so ist gewährleistet, dass auch nach Ende der Diplomarbeit weiter Daten gesammelt und ausgewertet werden können.

7.3.13 Technische Umsetzung

Die Erfassung der Daten aus dem in HTML umgesetzten Fragebogen geschieht auf dem Server unter <http://www.kunzelnick.de/da/evaluation> mittels ASP.NET.

Abbildung 7.2 zeigt das sehr einfache Datenbankschema der Tabellen, welche die Daten des Online-Fragebogens aufnehmen.

Für jeden ausgefüllten Fragebogen wird eine neue Spalte in der Tabelle DA_Form aufgenommen, die eine eindeutige Fragebogen-ID sowie das Erfassungsdatum enthält. Die n erfassten Datensätze werden in einer Key-Value-Relation in n neuen Spalten in der Tabelle DA_FormQuestion gespeichert und über einen Fremdschlüssel dem konkreten Fragebogen zugeordnet. In gleicher Weise werden noch einige Informationen zum Absender des Fragebogens gespeichert, zum Beispiel

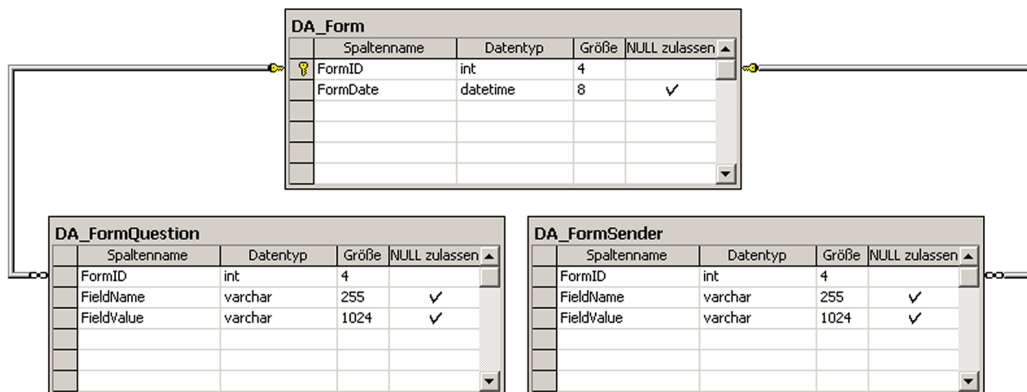


Abbildung 7.2: Datenbankschema

die IP-Adresse, um Mehrfachbenutzung des Fragebogens erkennen zu können und gegebenenfalls auch genauere Daten zu den verwendeten Plattformen und Browsern, um so weitere Daten gewinnen zu können.

7.4 Durchführung

Die Durchführung der betreuten Tests erfolgt im Wesentlichen in fünf Schritten:

1. Einleitung:

In der Einleitung wird je nach Vorwissen auf den Einsatzbereich des Applets eingegangen, und es werden kurz die Aufgaben des Applets umrissen. Es wird zusätzlich betont, dass es bei dem Test darum geht, Schwächen und Probleme innerhalb des Applets aufzudecken, und dass nicht der Benutzer bewertet werden soll. Es wird auch verdeutlicht, dass die Auswertung anonym erfolgt, dass der Benutzer also völlig frei in seiner Bewertung der Software ist, unter keinerlei Zeitdruck steht, und der Testleiter beim Test vollkommen unbeachtet bleiben soll, er damit aber auch keinerlei Hilfen geben kann.

Zuletzt wird noch auf die Think-Aloud-Methode (siehe 7.4.1) eingegangen und diese dem Tester näher gebracht.

2. Aufgabenstellung:

Es wird eine der insgesamt drei Aufgaben gestellt.

3. Aufgabendurchführung:

Der Tester wird bei der Ausführung der Aufgabe durch den Testleiter be-

obachtet. Wenn die Aufgabe erfüllt wurde oder die Aufgabe abgebrochen werden musste, wird der zweite Schritt wiederholt und eine neue Aufgabe gestellt. Sind alle Aufgaben abgearbeitet, wird mit dem vierten Schritt begonnen.

4. **Interview:**

Der Tester erhält die Möglichkeit, frei über seinen Gesamteindruck zu Performance, Bedienfreundlichkeit und so weiter zu reden.

5. **Fragebogen:**

Zum Abschluss füllt der Tester den Online-Fragebogen aus. Hierbei bleibt er unbeobachtet vom Testleiter. Damit hat der Testleiter keine Möglichkeit, nachzuvollziehen, welcher der Tester welchen Fragebogen ausgefüllt hat, um die Anonymität zu gewährleisten.

7.4.1 Think-Aloud-Methode

Die Think-Aloud-Methode wird in der Evaluation von Software oft zusammen mit anderen Maßnahmen wie etwa Fragebögen eingesetzt.

Bei dieser Methode soll der Tester einer Software gegenüber dem Leiter des Tests verbalisieren,

- was er sieht, beziehungsweise seine aktuellen Eindrücke und Empfindungen zu der vorliegenden Software;
- was er als Nächstes zu tun beabsichtigt;
- was er sich von der nächsten Aktion erwartet;
- wenn er eine Aktion durchführt, wie er die Reaktion der Software wahrnimmt;
- ob und in wie weit das wahrgenommene Ergebnis vom erwarteten Ergebnis abweicht;
- ob und was ihn verwirrt, abgelenkt oder sonst in irgendeiner Weise bei der Umsetzung seiner Ziele behindert hat.

Hierbei darf der Testleiter nicht oder nur sehr bedingt Hilfestellung leisten. Während des Tests soll sich aber keinesfalls ein Gespräch entwickeln, da der Benutzer möglichst unbeeinflusst bleiben soll.

Durch diese Vorgehensweise soll der Testleiter ein besseres Verständnis und eine bessere Vorstellung vom Gedankenmodell des Anwenders erlangen. Grundsätz-

liche Probleme der Anwender bei der Bedienung der Software können so besser erfasst und korrigiert werden.

Ein zusätzlicher Vorteil dieser Methode ist, dass die Beurteilung der Software freier ist. Das bedeutet, sie ist nicht auf die vom Testleiter festgelegten Richtungen (zum Beispiel durch Fragebögen) beschränkt, sodass auch Probleme in nicht oder unzureichend beachteten Bereichen der Software aufgedeckt werden können.

Beachtet werden muss bei dieser Methode allerdings auch, dass es, bedingt durch die ständige Verbalisierung und die Beobachtung durch den Testleiter, zu unerwünschten Effekten kommen kann: Wie verschiedene Studien zeigten, wurden von Testern, die die Think-Aloud-Methode anwendeten, bis zu 80 Prozent weniger Fehler gemacht als von anderen Testern. Dies ist zum einen dadurch bedingt, dass durch die Verbalisierung alle Aktionen langsamer durchgeführt werden, und die Tester gezwungen sind, mehr über beabsichtigte Aktionen nachzudenken, zum anderen aber auch dadurch, dass Tester durch die Verbalisierung schneller lernen und sogar Fehler in den eigenen Gedankengängen schon vor der Interaktion mit der Software erkennen und korrigieren können.

7.4.2 Aufgabenstellungen

Die Aufgabenstellungen sollen möglichst schnell die Benutzung aller wichtigen Funktionalitäten der Software abdecken. Aus diesem Grunde werden zwei Aufgaben zu den eigentlichen Kernbereichen der Software - der dynamischen 3D- und der statischen 2D-Visualisierung gestellt. Hinzu kommt noch eine Aufgabe zur Online-Hilfe, da auf deren einfache Bedienbarkeit bei der Erstellung besonderer Wert gelegt wurde.

Ausgangssituation für alle Aufgaben ist die Startseite des Applets beziehungsweise der Java Web Start-Applikation.

Dynamische Visualisierung

Hier sieht die Aufgabe vor, dass eine konkrete AnT-Datei zur dynamischen Visualisierung ausgewählt werden soll. Dabei handelt es sich um die Datei „/user/8888/phase_portrait.tna“.

In der Visualisierung soll folgendes Ziel erreicht werden:

- Visualisierung der x, y und z-Achse

- Darstellung von etwa 4000 Punkten in Liniendarstellung, wobei die Linien nach Geschwindigkeit gefärbt werden sollen.
- Als Hilfsmittel soll ein Gitter in der xz-Ebene eingeblendet werden, dessen Linienabstände möglichst klein werden sollen.
- Die 3D-Visualisierung soll so gestaltet werden, dass der Bereich, der über dem Gitter liegt (mit $z \geq 0$), möglichst groß gezoomt in der Mitte des 3D-Bereiches liegt.

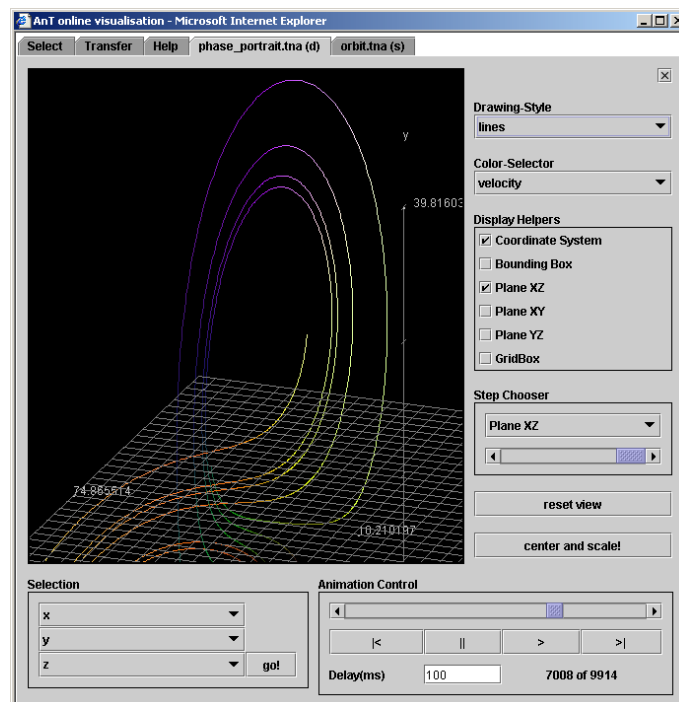


Abbildung 7.3: Angestrebte dynamische Visualisierung

Die Aufgabe gilt als erledigt, wenn sich eine dynamische Visualisierung vergleichbar mit Abbildung 7.3 ergibt.

Statische Visualisierung

Die Aufgabe lautet hier: Erzeugen einer statischen Visualisierung der Datei „/default/systems/logistik/demo01/orbit.tna“. Dabei sollen als Achsen „col0“ und „col1“ in einem Bereich von $2500 \leq col0 \leq 2505$ und $0.0 \leq col1 \leq 1.0$ gewählt werden.

Die Aufgabe gilt als erledigt, wenn sich eine Visualisierung ergibt, die vergleichbar zu Abbildung 7.4 ist.

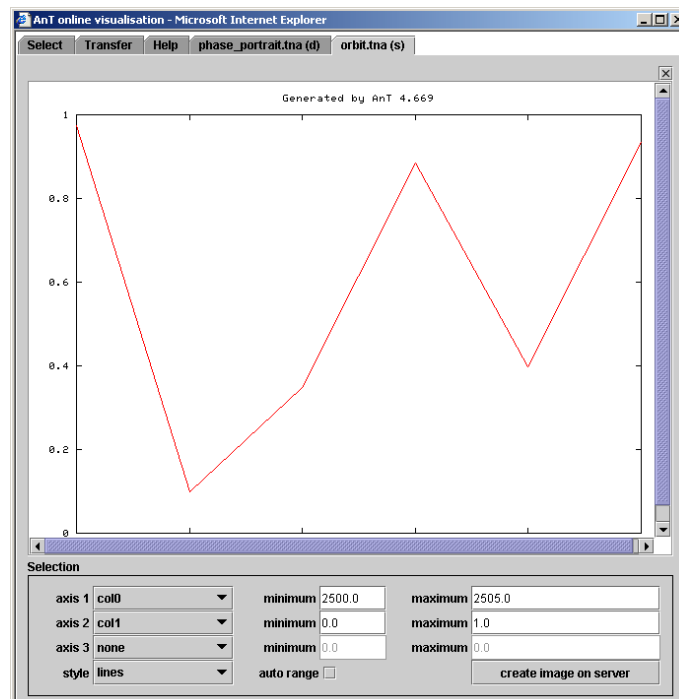


Abbildung 7.4: Angestrebte statische Visualisierung

Online-Hilfe

Um die Online-Hilfe zu evaluieren, soll angenommen werden, dass dem Benutzer nicht klar ist, was der „Step-Chooser“ in der dynamischen Visualisierung ist.

Aufgabe ist es dann, die entsprechende Stelle in der Online-Hilfe zu finden und zu bewerten, ob die Dokumentation hilfreich ist. Es soll angenommen werden, dass dies nicht der Fall ist und eine Frage an den Autor des Applets oder das AnT-Team gestellt werden muss.

Die Aufgabe ist dann als erledigt anzusehen, wenn eine Testmail an den Autor des Applets oder das AnT-Team abgesetzt hätte werden können.

7.4.3 Online-Fragebogen

Nach den Tests oder nachdem unbekannte Besucher das Applet benutzt haben, ergibt sich über einen Online-Fragebogen die Möglichkeit einer Bewertung.

Die Daten des Fragebogens werden dabei von einer ASP.NET-Anwendung auf der Website zur Diplomarbeit (siehe Kapitel 1.3.2) entgegengenommen und in

eine Datenbank auf einem Microsoft SQL Server 2000 gesichert.

7.5 Ergebnisse

Die Daten der in der Datenbank gespeicherten Fragebögen werden automatisch über eine ASP.NET-Anwendung aufbereitet und können über die Website zu dieser Diplomarbeit (siehe 1.3.2) eingesehen werden.

Die bis zum Abschluss der Arbeiten an dieser Ausarbeitung verfügbaren Daten, sind im Anhang (siehe E.2) zu finden.

Soweit die Evaluation Probleme aufdeckt, die mit geringem Aufwand korrigiert werden können, werden diese in einem Wartungszyklus innerhalb dieser Arbeit behoben. Ansonsten gehen die Ergebnisse in Kapitel 8.2 ein und bleiben weiteren Wartungszyklen, die unabhängig von dieser Diplomarbeit durchgeführt werden sollten, vorbehalten.

Kapitel 8

Fazit und Ausblick

8.1 Fazit

In dieser Arbeit wurden 45 Plugins auf ihre Tauglichkeit zur einfachen, komfortablen, performanten und plattformunabhängigen Darstellung von Punktwolken auf verschiedene Weisen untersucht. Die dynamische, dreidimensionale und die statische zweidimensionale Visualisierung dieser durch AnT erzeugten Punktwolken wurde dann in Java umgesetzt. Zum Abschluss der Arbeit wurde die entstandene Software durch eine Evaluation, unter anderem mit Hilfe der Norm ISO9241-10, auf ihre Gebrauchstauglichkeit untersucht.

Hierbei ist nicht nur eine voll funktionsfähige, zuverlässige und frei konfigurierbare Software für eine Online-Visualisierung zur Vervollständigung der AnT-Software für Forschung und Lehre entstanden, sondern es ist auch eine der bis dahin größten Untersuchungen von Plugins zur Darstellung von 3D-Inhalten zusammengetragen worden, die im World Wide Web zu finden ist.

Während dieser Arbeit musste sich der Autor mit vielen anspruchsvollen und interessanten Bereichen aus der Softwaretechnik intensiv beschäftigen, wie zum Beispiel:

- Untersuchung, Bewertung und Auswahl geeigneter Technologien
- GUI-Programmierung mit Swing
- Thread-Programmierung
- 3D-Programmierung
- Netzwerkprogrammierung

- Anwenden von Design-Patterns
- Datenbankprogrammierung
- Erstellung von statischen und dynamischen Seiten für das WWW
- Evaluation von Software auf Basis von ISO-Normen

8.2 Ausblick

Folgende Verbesserungen von AnT-Online sollten in nächsten Versionen ins Auge gefasst werden:

- **Wiederverwendbare Netzwerkverbindungen**
Die Netzwerkschicht ist von Anfang an so angelegt, dass einmal benutzte Netzwerkverbindungen nach dem Erfüllen ihrer Aufgabe nicht weiter benutzt werden und irgendwann automatisch geschlossen werden. Wünschenswert wäre natürlich, unter anderem aus Performance-Gründen, dass verwendete Verbindungen weiter benutzt werden, zum Beispiel über ein Pooling durch den ConnectionManager.
- **Wiederaufnehmbare Datenübertragung**
Die Übertragung von AnT-Dateien war so spezifiziert, dass sie zwar abgebrochen, nicht aber fortgesetzt oder neu aufgenommen werden kann. Das Fortsetzen oder erneute Starten von abgebrochenen Datenübertragungen wäre eine sinnvolle Verbesserung, auch wenn dazu das Datenübertragungsprotokoll geändert werden muss.
- **Verbesserte 3D-Darstellung**
Bei der 3D-Darstellung sind noch einige Verbesserungen denkbar, zum Beispiel könnte die Art der Projektion wählbar sein. Weniger wichtig, aber dennoch interessant könnte auch eine Überarbeitung sein, sodass es auch möglich wäre, Ebenen darzustellen, eventuell sogar Ebenen, die Shader zulassen. Hier kämen dann aber einige Probleme (Verdeckung, Clipping, Füllen, usw.) hinzu, welche sich wieder negativ auf die Performance auswirken dürften.
- **Verbessertes Caching**
Bis jetzt werden hauptsächlich die Beschreibungsdaten und die eigentlichen AnT-Daten im Cache abgelegt. Es könnte aber auch darüber nachgedacht werden, zum Beispiel die von Gnuplot erzeugten Bilder zwischenspeichern, um diese dann nochmals durchblättern zu können oder ähnliches.

- **Kompression**

Die Übertragung aller Daten erfolgt bis jetzt ohne jegliche Kompression. Dies könnte noch verbessert werden, um die Ladezeiten deutlich zu verringern. Auch wenn es gegebenenfalls bei den durch Gnuplot generierten Bildern keinen großen Vorteil bringen dürfte, sollte eine Verbesserung der Übertragung der AnT-Daten einen deutlichen Gewinn an Geschwindigkeit bringen. Hier sind zwei Optimierungen denkbar:

- Die Punkt-Daten sollten nicht mehr als Zeichenketten übertragen und erst auf der Client-Seite ausgewertet werden, sondern dies sollte bereits auf dem Server geschehen, um dann die Werte direkt als float-Werte, eingebunden in eine eigene Struktur übertragen zu können
- Die Streams sollten vor der Übertragung komprimiert werden. Hier muss gegebenenfalls geprüft werden, ob der auf Server und Client anfallende Rechenaufwand durch die Verkleinerung des zu übertragenden Datenvolumens gerechtfertigt ist.

Anhang A

Technologieanalyse

A.1 Möglichkeiten zur 3D-Visualisierung

Das folgende Kapitel enthält eine komplette Übersicht der untersuchten Plugins zur 3D-Visualisierung. Die genauere Untersuchung ergab bei allen folgenden Plugins, dass sie nicht für den Einsatz im Rahmen dieser Arbeit geeignet sind. Details über die geeigneten Plugins finden Sie in Kapitel 2.3.

A.1.1 3D Anywhere [1]

Vorteile

- + Komplet in Java implementiert

Nachteile

- Wird nicht mehr weiterentwickelt

A.1.2 3D Groove [2]

Vorteile

- + sehr gute Performance
- + multiuserfähig, damit auch netzwerkfähig (Nachladen/Streamen möglich)

Nachteile

- wurde hauptsächlich für Spiele entwickelt
- lediglich ein Director Xtra (siehe Kapitel 2.3.4), deswegen ergeben sich auch alle Nachteile, die Shockwave aufweist.

A.1.3 3DMeNow / XPlayer [3]

Vorteile

- + keine gefunden

Nachteile

- nur für die Visualisierung von Charakteren gedacht
- nur für Windows verfügbar

A.1.4 3DX Player [4]

Vorteile

- + keine gefunden

Nachteile

- nur für Windows und MacOS verfügbar
- nur ein Plugin um 3DX-Dateien, die mit dem 3D Atelier erstellt wurden, im Web verfügbar zu machen
- geringe Verbreitung
- keine Sprachunterstützung

A.1.5 3Space [7]

Vorteile

- + XML-basiert
- + eigene Scriptsprache
- + Physik-Engine
- + für nichtkommerzielle Anwendungen kostenlos

Nachteile

- nur für Windows und MacOS verfügbar

- wenig verbreitet

A.1.6 Activate!3D [8]

Vorteile

- + Java-basiert und damit als Applet auf vielen Plattformen lauffähig
- + alle Vorteile von Java

Nachteile

- schlechte Performance
- schlechte Bildqualität
- alle Nachteile von Java

A.1.7 Adobe Atmosphere [9]

Vorteile

- + streamingfähig
- + programmierbare API (Unterstützung von JS, Java, SOAP, ...)
- + integrierte Physik-Engine
- + sehr hohe Bildqualität durch Unterstützung von Real-Time-Schatten, Nebel, dynamischer Beleuchtung und weiteren Effekten

Nachteile

- nur Unterstützung für IE unter Windows
- wenig verbreitet

A.1.8 Anark [10]

Vorteile

- + sehr performant
- + Renderengine unterstützt OpenGL und DirectX

Nachteile

- nur für Windows und MacOS verfügbar
- kann nur mit kostenpflichtigem Anark-Studio erstellt werden

- keine richtige Scriptsprache oder API; Steuerung nur über vorhandene behaviours möglich
- mehr für Multimedia mit 3D-Effekten gedacht
- wenig verbreitet

A.1.9 Anfy3D [11]

Vorteile

- + basiert auf Java und damit voll programmierbar
- + alle Vorteile von Java

Nachteile

- zum Download werden nur eine Windows- und MacOS-Version angeboten
- benötigen installierte Java RE, damit alle Probleme von Java
- ist kostenpflichtig

A.1.10 Axel [12]

Vorteile

- + dynamische Inhalte mit DB-Interaktion möglich
- + gute Rendering-Qualität (Texturen, dynamisches Licht und weitere Features)

Nachteile

- nur für Windows und MacOS verfügbar
- erfordert kostenpflichtiges Programm zur Erstellung
- keine richtige Scriptsprache

A.1.11 B3D [13]

Vorteile

- + sehr performant durch Nutzung von 3D-Hardware
- + skalierbar (unter anderem Qualitätsstufen der Schatten) und damit gut an Hardwareausstattung des Benutzers anpassbar

Nachteile

- nur für Windows verfügbar
- nur mit kostenpflichtiger Software erstellbar
- schlechte Dokumentation

A.1.12 Blaxxun Virtual Worlds Platform [14]

Vorteile

- + basiert auf dem VRML-Standard
- + verfügt über eine Java-API

Nachteile

- Plugin und Software nur für Windows verfügbar
- hält sich nicht an den VRML-Standard und ist damit nicht kompatibel zu anderen VRML-Playern und Plugins

A.1.13 CharActor [15]

Vorteile

- + keine gefunden

Nachteile

- nur für Windows verfügbar
- nur zur 3D-Visualisierung von 3D-Charakteren

A.1.14 Cortona [16]

Vorteile

- + siehe VRML

Nachteile

- nur für Windows und MacOS verfügbar
- weiteres siehe VRML

A.1.15 Cosmo Player [17]

Vorteile

- + siehe VRML

Nachteile

- nur für Windows und MacOS verfügbar
- weiteres siehe VRML

A.1.16 Critical Reach [18]

Vorteile

- + keine gefunden

Nachteile

- geeignet, um technische Informationen zu Maschinen per Web, zum Beispiel für Servicetechniker, bereitzustellen und diese mit 3D-Visualisierungen zu unterstützen
- zu wenig flexibel und kaum für Anwendungen geeignet

A.1.17 Cult 3D [19]

Vorteile

- + für verschiedene Plattformen verfügbar (Windows, MacOS, Linux, Solaris)
- + relativ weit verbreitet
- + Plugins für PDF und Office-Anwendungen verfügbar

Nachteile

- kein offenes Format, kann damit nicht auf dem Server erzeugt werden

A.1.18 Cybercore Entrance [20]

Vorteile

- + keine gefunden

Nachteile

- wird nicht mehr weiterentwickelt

A.1.19 EON [21]

Vorteile

- + Ausnutzung vorhandener 3D-Hardware (OpenGL, DirectX)
- + sehr performant
- + unterstützt Streaming

Nachteile

- nur für Windows-Plattform verfügbar, keine Unterstützung für Unix, MacOS und weitere Betriebssysteme.

A.1.20 eViewer [22]

Vorteile

- + Plugin für verschiedene Browser verfügbar (IE, NS, Opera)

Nachteile

- nur für Windows-Plattform verfügbar
- eher gedacht, um bestehende CAD-Daten zu visualisieren und zu wenig flexibel, um Anwendungen zu erstellen
- wenig verbreitet

A.1.21 Flash [62]

siehe Kapitel 2.3.3

A.1.22 Flatland Rover / 3DML [23]

Vorteile

- + eigene XML-Dateistruktur (3D Markup Language)
- + einfache Textdateien, die einfach auf dem Webserver erzeugt werden können

- + eigene Scriptsprache (Roverscript)
- + gute Dokumentation, Tutorials, Foren und Communities.
- + einfach zu lernen

Nachteile

- erfordert DirectX 8.0 oder höher - damit nur unter Windows verfügbar
- proprietäres Format von Flatland
- Sprache wenig mächtig
- nur mäßig verbreitet

A.1.23 Hypercosm 3D Player [24]

Vorteile

- + sehr kleines Plugin (1MB)
- + performant, da bestehende 3D-Hardware genutzt wird

Nachteile

- nur für Windows-Plattform verfügbar
- kaum Dokumentation verfügbar
- wenig verbreitet

A.1.24 iPix [25]

Vorteile

- + relativ weit verbreitet
- + als Java-Applet verfügbar (damit alle Vorteile von Java)
- + für verschiedene Browser verfügbar
- + unterstützt Streaming

Nachteile

- nur für Windows und MacOS verfügbar
- eher zum Anzeigen von 360°-Bildern gedacht
- alle Nachteile von Java (in der Applet-Version)

A.1.25 Miner3D [27]

Vorteile

- + XML-basiert, damit einfach zu erstellen
- + dynamische Anzeige geänderter Daten

Nachteile

- wenig verbreitet
- Plugin nur für Windows verfügbar

A.1.26 Objects to See [28]

Vorteile

- + verfügbar für IE 4+ und NS 4+
- + kann einfach auch in nicht web-basierte Anwendungen eingebunden werden

Nachteile

- nur für Windows verfügbar

A.1.27 Openworlds [29]

Vorteile

- + programmierbare API (in C++)
- + komplettes Framework

Nachteile

- nur für Windows und SGI verfügbar
- kaum verbreitet

A.1.28 plan_b media [30]

Vorteile

- + keine gefunden

Nachteile

- zwar 3D, aber nur um Charaktere für Benutzeransprache zu realisieren

A.1.29 Pulse 3D [31]

Vorteile

- + basiert auf Java-Applets und ist damit überall lauffähig, Java RE vorausgesetzt
- + alle Vorteile von Java

Nachteile

- dient eher zur Visualisierung von 3D-Modellen von Personen und Tieren um sie sprechen zu lassen
- alle Nachteile von Java

A.1.30 Quest3D [32]

Vorteile

- + sehr performant

Nachteile

- nur für Windows verfügbar
- nur gedacht, um Quest3D-Projekte im Web publizieren zu können

A.1.31 Quicktime VR [33]

Vorteile

- + weit verbreitet
- + Plugin unterstützt schon Funktionen wie Zoom, Rotation und andere
- + dank Integration in Apples Quicktime relativ weit verbreitet und einfach zu installieren
- + für viele Plattformen verfügbar

Nachteile

- vorgerenderte Darstellungen. Es können keine nachträglichen Änderungen an der Szene vorgenommen werden

- eher gedacht, um 360°-Panoramen darzustellen
- kein Nachladen und Streamen möglich
- zur Programmierung wäre Einbindung in Java-Applet nötig

A.1.32 RichFX [34]

Vorteile

- + keine gefunden

Nachteile

- Entwicklung eingestellt, teilweise zugunsten von Implementierungen in Flash

A.1.33 SCOL [35]

Vorteile

- + relativ kleines, einfach zu installierendes Plugin (1.2MB)

Nachteile

- Dokumentation nur auf Französisch
- nur für Windows und MacOS verfügbar
- wenig verbreitet
- relativ alt (2000/2002)
- es ist fraglich, ob es noch weiterentwickelt wird

A.1.34 Shockwave [62]

siehe Kapitel 2.3.4

A.1.35 Shout 3D [36]

Vorteile

- + auf OpenGL basierendes Java-Render-Applet, damit auch voll über Java programmierbar
- + läuft auf jedem Browser, solange Java 1.1 verfügbar ist

- + alle Vorteile von Java

Nachteile

- Voraussetzung ist installiertes Java
- nicht frei verfügbar
- gedacht, um Modelle aus 3D-Programmen (3DMax) im Browser anzeigen zu können
- alle Nachteile von Java

A.1.36 SPX/Open SPX [37]

Vorteile

- + Unterstützung für verschiedene Browser (NS und IE)
- + streamingfähig
- + unter GPL-Lizenz verfügbar

Nachteile

- Plugin nur für Windows gefunden
- wenig verbreitet
- wenig brauchbare Dokumentation im Web
- hauptsächlich für Scannerdatenvisualisierung gedacht

A.1.37 Viewpoint Media Player [38]

Vorteile

- + sehr weit verbreitet (nach Ipsos-Studie, siehe Abbildung 2.1)
- + einfach zu installieren, relativ klein (1.3MB)
- + scheint relativ mächtig und ausgereift
- + XML-basierte Szenenbeschreibung die on-the-fly geändert werden kann
- + integriertes Update
- + kann andere Medien integrieren (hochauflösende Bilder, Flash, Sounds, ...)
- + nutzt 3D-Hardware
- + Real-Time-Antialiasing
- + Unterstützung von Licht und Schatten
- + Texturen möglich
- + Kompression der übertragenen Daten

- + verwendet Scenegraph wie Java, ist aber komplett unabhängig von Java

Nachteile

- nur für Windows und MacOS verfügbar

A.1.38 Virtools [39]

Vorteile

- + verfügbar für verschiedene Browser (IE, NS, AOL, Compuserve)
- + geringe Leistungsanforderungen (Pentium 166 oder PowerPC 604)
- + einfache Installation des kleinen Plugins (750kB)
- + unterstützt DirectX und OpenGL
- + von außen steuerbar per JavaScript
- + intern steuerbar über Behaviours, in die man auch C++-Code integrieren kann
- + Kompression
- + einigermaßen weit verbreitet

Nachteile

- Plugin nur für Windows und MacOS verfügbar
- Streaming nur für Sound möglich (mp3, wma)
- nur Plugin ist frei verfügbar, Entwicklungsumgebung ist kostenpflichtig
- serverseitig schwer zu generieren

A.1.39 Virtue3D [40]

Vorteile

- + geringe Anforderungen an Bandbreite (min. 28.8kb/s durch Kompression) und Rechenleistung (min. 166MHz Pentium)
- + Plugin nur 400kB groß und einfach zu installieren
- + API, die zum Beispiel über Java, C++, VB und andere Sprachen angesprochen werden kann
- + unterstützt viele Dateiformate (VRML 2.0, 3DS, OBJ, PLY, STL, DXF)
- + verfügbar für IE und NS

Nachteile

- nur Plugin kostenlos verfügbar, sonst kostenpflichtig, funktioniert aber auch ohne Lizenz (Vermerk „Sample“ im 3D-Modell)
- wenig verbreitet
- eher zum Export bestehender CAD-Daten in eine statische, proprietäre „VTU“-Datei gedacht
- kein Streaming
- nur unter Windows verfügbar

A.1.40 VizStream [41]

Vorteile

- + das Plugin unterstützt schon Aktionen wie Rotieren, Zoomen, Detailbetrachtungen und weitere
- + unterstützt Streaming
- + skalierbar
- + selektives Streaming, um Volumen zu sparen
- + benutzt Standard-Dateiformate, unter anderem XGL
- + erlaubt Anpassung und Animation durch verschiedene Sprachen, u.a. JS

Nachteile

- nicht kostenlos, nur Trial-Version verfügbar
- eher zum Export von bestehenden CAD-Daten ins Web gedacht
- unterstützt nur IE ab Version 5.0, da das Plugin als ActiveX-control implementiert ist und damit nur unter Windows nutzbar

A.1.41 VRML [45]

Vorteile

- + siehe Kapitel 2.3.5

Nachteile

- siehe Kapitel 2.3.5

A.1.42 WF-3D [42]

Vorteile

- + sehr kleiner 3D-Renderer für Java. Es ist außer Java 1.1 keine weiteres Plugins nötig
- + verfügbar für Windows, MacOS und in einer Java-Version

Nachteile

- Installer nur für Windows verfügbar (baut auf DirectX 5.0 oder höher auf)

A.1.43 WildTangent [43]

Vorteile

- + Zugriff auf die komplette API über COM und damit Programmierung in C++, C# und weiteren Sprachen möglich
- + nur ca. 1 MB und einfache Installation
- + durch Streaming und Kompression auch für Internet-Anbindung mit geringer Bandbreite geeignet
- + relativ gute Verbreitung
- + Ausnutzung von vorhandener 3D-Hardware
- + unterstützt verschiedenste Eingabegeräte
- + sehr performant
- + unterstützt verschiedenste Beleuchtungsarten (Ambient, point, spot and directional lighting)
- + Unterstützt verschiedenste Standards und Best Practices (JavaScript, VBScript, Java, Software rendering, polygon and bounding box collision detection, mouse and keyboard event capture, weighted vertex skeletal animation)
- + unter verschiedenen Browsern einsetzbar (IE ab 5.5, NS ab 4.7x)
- + das Plugin ist kostenlos, genauso ist das SDK zur Entwicklung von nicht-kommerziellen Anwendungen frei verfügbar

Nachteile

- baut auf Microsoft DirectX 5.0 oder höher auf und ist damit nur für verschiedene Windows-Versionen verfügbar

A.1.44 X3D [45]

Vorteile

- + Mitte 2004 sollte Standardisierung abgeschlossen sein
- + als Sprachen stehen je nach Version Java, Javascript, VRMLScript oder ECMAScript zur Verfügung
- + Weiterentwicklung von VRML
- + ein XML-Format

Nachteile

- noch nicht verbreitet
- noch kaum Plugins
- Nachladen und Streamen nur bedingt möglich
- Entwicklung nach Mitte 2004 noch nicht abschätzbar
- erscheint zu spät für die Implementierungsphase

A.1.45 ZAP [44]

Vorteile

- + keine gefunden

Nachteile

- wird nicht mehr weiterentwickelt

A.2 Möglichkeiten für die Datenhaltung

A.2.1 MySQL (4.1)

Vorteile

- + unterstützt Replikation (SSL)
- + Volltextindexierung
- + einfach zu bedienen
- + schnell
- + Subqueries
- + abgeleitete Tabellen

- + Unterstützung für UTF8 und UCS2
- + multiple Ausdrücke und Resultsets
- + Fremdschlüssel mit kaskadierendem Update/Delete
- + Quasistandard für Webanwendungen
- + MySQL Control Center - grafische Oberfläche zur Verwaltung der DB (plattformunabhängig)
- + sehr gute Dokumentation
- + sehr große Zahl von Benutzern (viel Literatur, Tutorials, Foren, ...)

Nachteile

- nicht konform zum SQL-Standard
- eingeschränkter Funktionsumfang
- keine Stored Procedures verfügbar (geplant ab Version 5.1)
- keine Views (geplant ab Version 5.1)
- keine Trigger (geplant ab Version 5.1)
- Leistungseinbußen bei größeren Projekten - damit nur schlecht skalierbar.

Fazit geeignet

A.2.2 PostgreSQL (7.3.4)

Vorteile

- + volle Unterstützung für ACID
- + gewährleistet referenzielle Integrität
- + volle Unterstützung von Fremdschlüsseln mit kaskadierenem Update/Delete
- + unterstützt Trigger und Stored Procedures
- + Rules
- + Subselects
- + Sichten (Views)
- + für 34 Plattformen verfügbar
- + sehr stabil
- + umfassendere SQL-Standard-Konformität als zum Beispiel MySQL
- + insgesamt sehr viele Features
- + native Schnittstellen für ODBC, JDBC, C, C++, PHP, Perl, TCL, ECPG, Python und Ruby
- + unterstützt Sequences, Vererbung, Outer Joins

- + ermöglicht benutzerdefinierte Datentypen
- + gut skalierbar
- + verhältnismäßig schnell, gerade bei großen Projekten

Nachteile

- noch keine Replikationsunterstützung
- noch keine Volltextindexierung
- Tradeoff zwischen großer Funktionsvielfalt und Geschwindigkeit
- verhältnismäßig langsam bei sehr kleinen Projekten

Fazit geeignet

A.2.3 Hypersonic SQL (1.7.1)

Vorteile

- + Stored Procedures
- + komplett in Java implementiert
- + plattformunabhängig

Nachteile

- nicht multithreaded
- wenig performant
- geringe Verbreitung, dementsprechend weniger Informationen, Dokumentation, Foren und Communities

A.2.4 Interbase (7.1)

Vorteile

- + keine besonderen Vorteile gefunden

Nachteile

- geringe Verbreitung, dementsprechend weniger Informationen, Dokumentation, Foren und Communities

Anhang B

Klassendiagramme

B.1 Die Package-Struktur

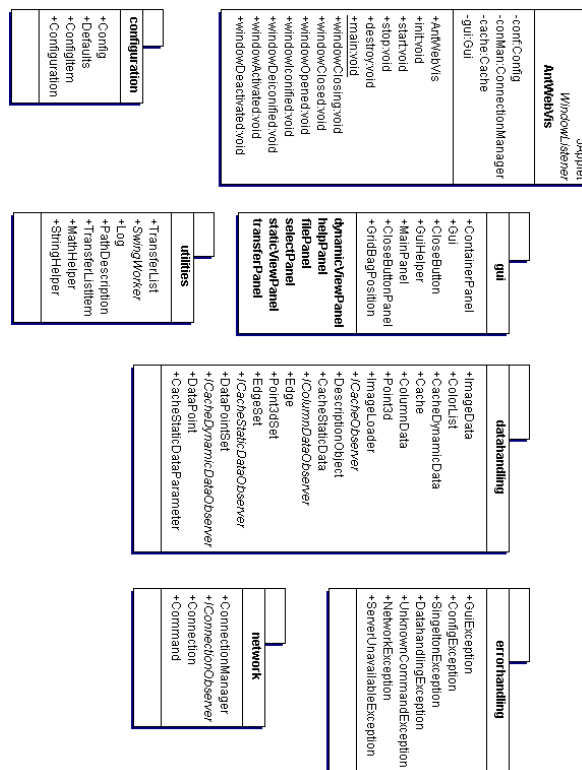


Abbildung B.1: Package-Übersicht

B.4 Das Package „errorhandling“

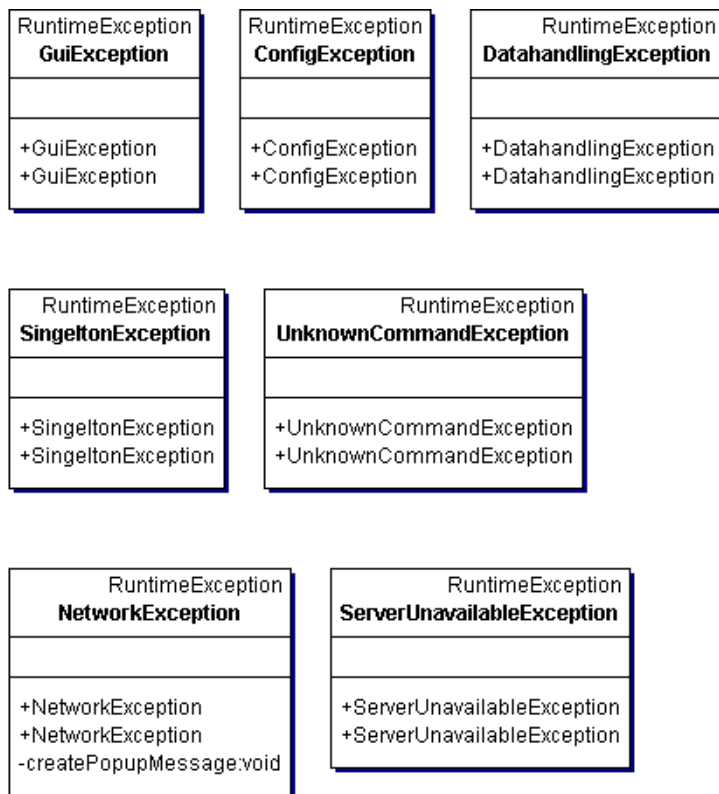


Abbildung B.4: errorhandling

Das Package „filePanel“

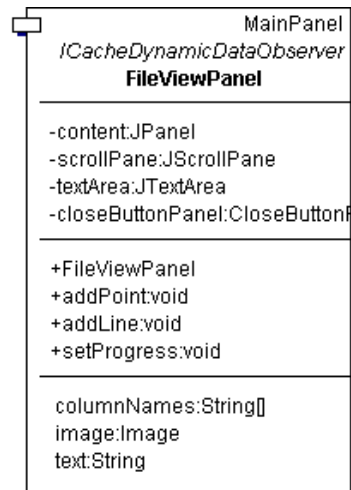


Abbildung B.6: filePanel

Das Package „helpPanel“

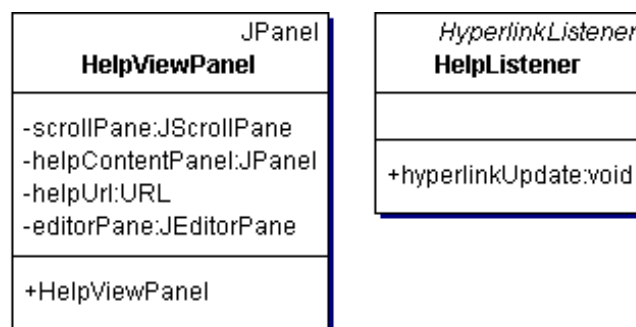


Abbildung B.7: helpPanel

Das Package „selectPanel“

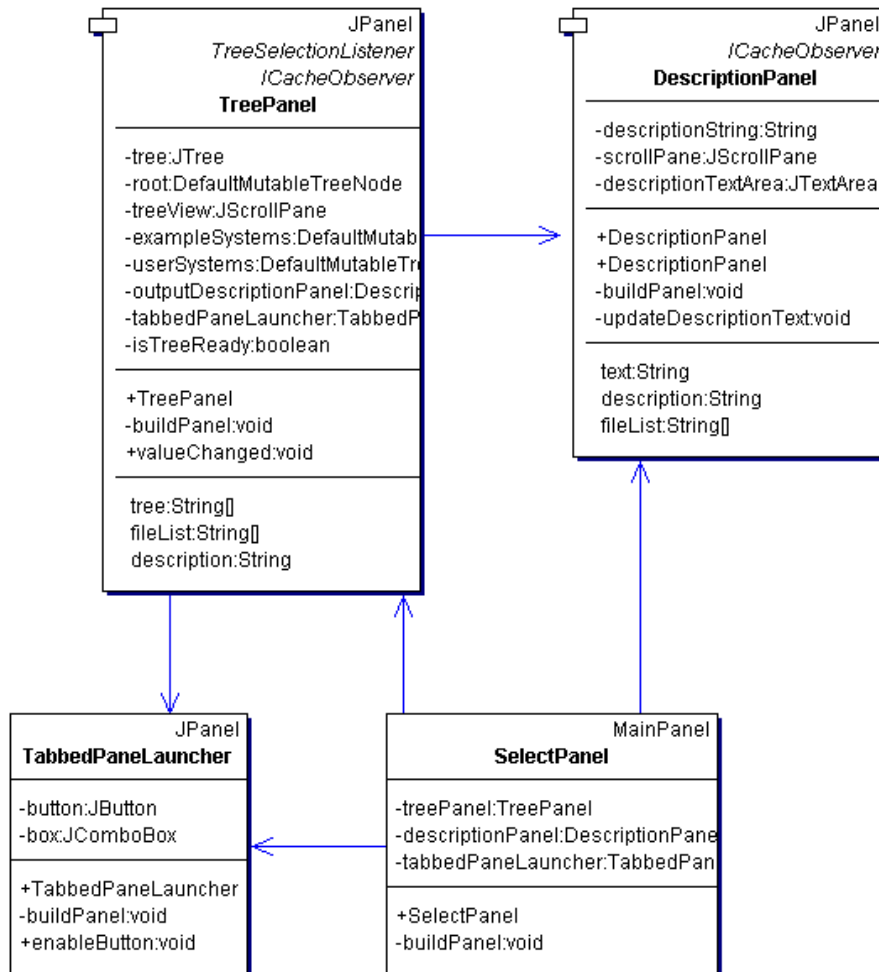


Abbildung B.8: selectPanel

Das Package „staticViewPanel“

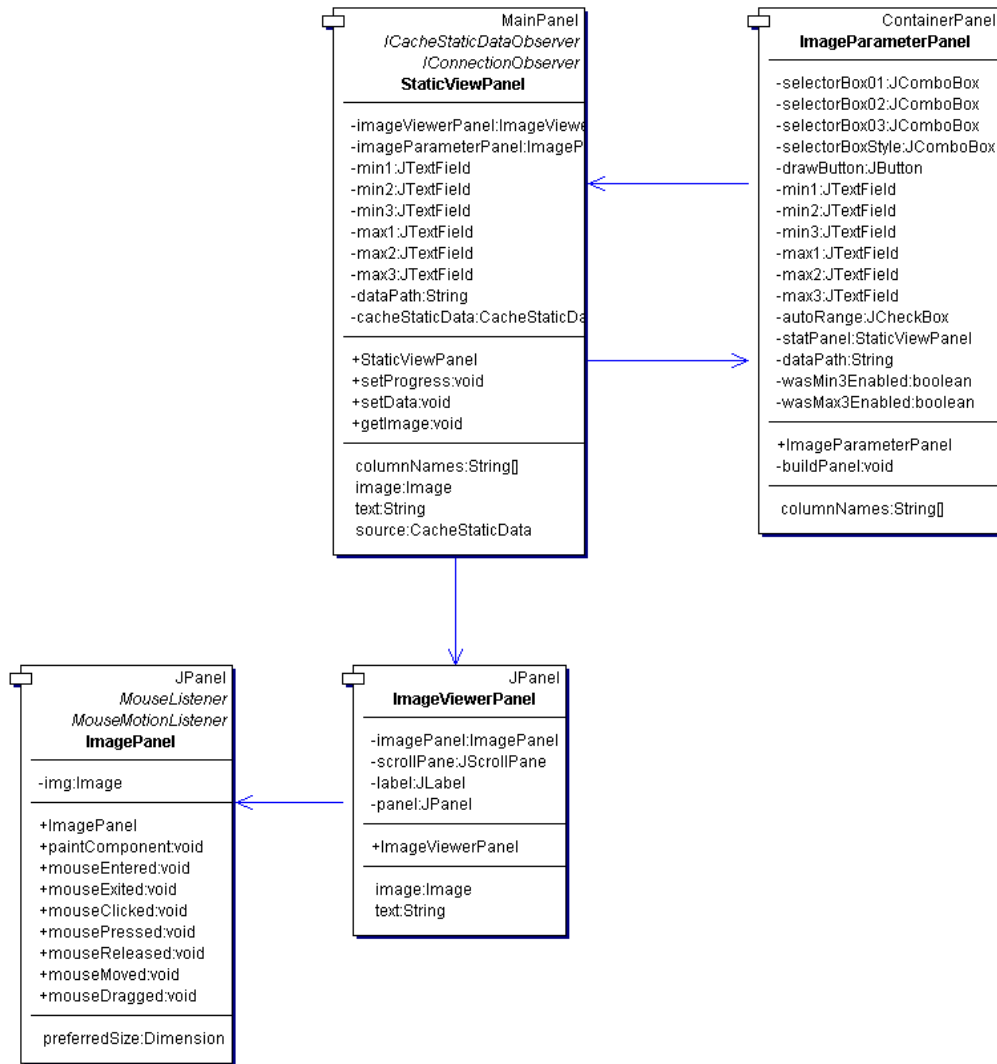


Abbildung B.9: staticViewPanel

Das Package „transferPanel“

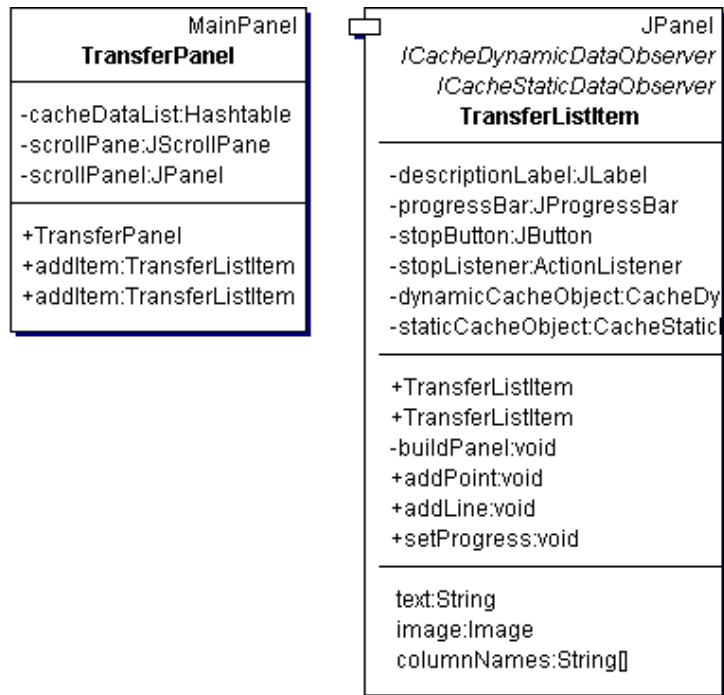


Abbildung B.10: transferPanel

B.6 Das Package „network“

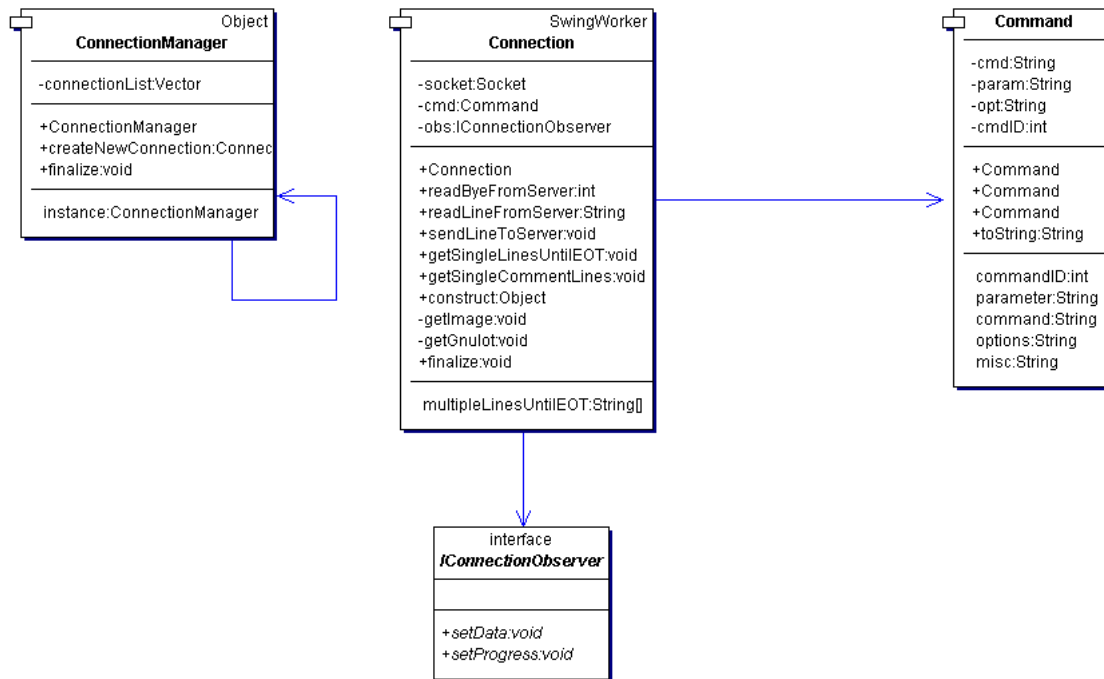


Abbildung B.11: network

B.7 Das Package „utilities“

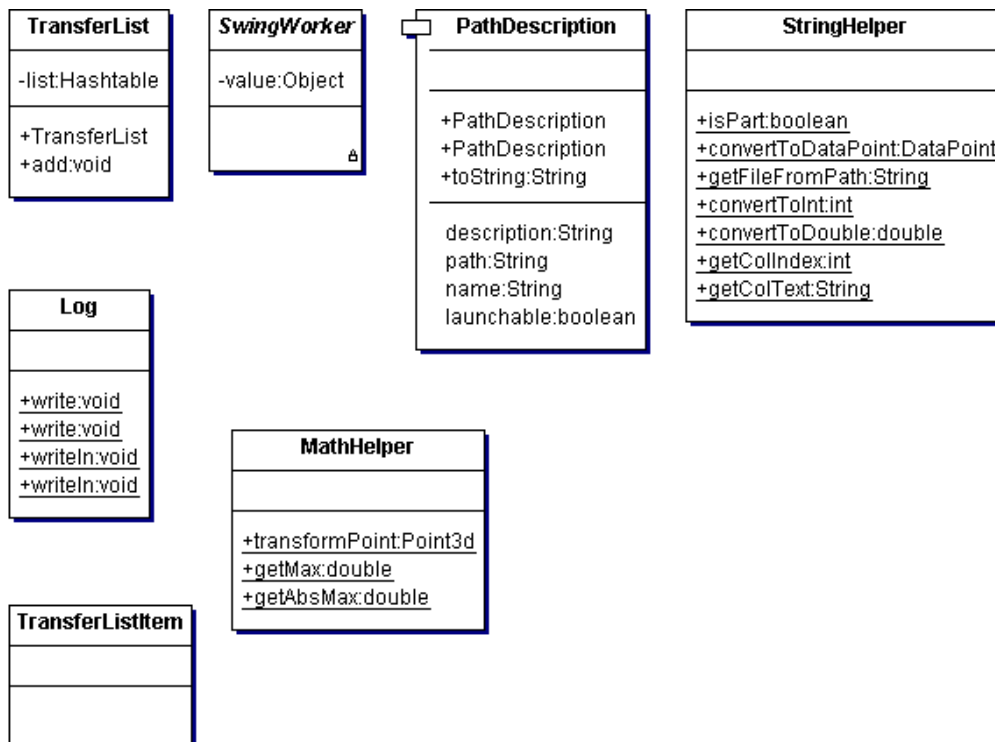


Abbildung B.12: utilities

Anhang C

XML-Konfigurationsdatei

Beispiel für eine Konfigurationsdatei für AnT-Online:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!--
***** config file for AnTWebVis applet/application *****
** <item name="<key>" value="<value>" type="<javatype>"
** <key> = key to reference within the applet
** <value> = value to assign with the key
** <javatype> = type that is known in Java.
** Supported types up to now: int, String, Double, Color, String[]
** Attention:
** - for Color-Items, the <value> must look like this: "r#g#b"
**   with int-values for r, g and b and 0 < r, g, b < 255
** - for String[]-Items, the <value> must look like this: "v1#v2#v3#...#vn"
** - for Border-items, the <value> must look like this: "<borderstyle>#r#g#b"
*****
-->
<configdata>
  <!-- ***** start logging-config ***** -->
  <section>
    <item
      name="LOG_LEVEL"
      value="1000"
      type="int"
    />
    <item
      name="LOG_CRITICAL_ERROR"
      value="1"
      type="int"
    />
    <item
      name="LOG_MEDIUM_ERROR"
      value="2"
      type="int"
    />
    <item
      name="LOG_LIGHT_ERROR"
      value="3"
      type="int"
    />
  />
</section>
</configdata>
```

```

<item
  name="LOG_WARNING"
  value="100"
  type="int"
/>
<item
  name="LOG_MESSAGE"
  value="200"
  type="int"
/>
<item
  name="LOG_DEBUG"
  value="300"
  type="int"
/>
</section>
<!-- ***** end logging-config ***** -->

<!-- ***** start networking-config ***** -->
<section>
  <item
    name="NET_SERVER"
    value="pamaiva.dyndns.org"
    type="String"
  />
  <item
    name="NET_PORT"
    value="4444"
    type="int"
  />
  <item
    name="NET_EOT"
    value="[EOT]"
    type="String"
  />

  <item
    name="NET_FILESIZE"
    value="Size"
    type="String"
  />
  <item
    name="NET_PATH_SEPERATOR"
    value="/"
    type="String"
  />
  <item
    name="NET_NUMBER_DELIMITER"
    value="_"
    type="String"
  />
  <item
    name="NET_COMMENT_DELIMITER"
    value="#"
    type="String"
  />
  <item
    name="NET_PROPERTY_DELIMITER"
    value="_|_"
    type="String"
  />

```

```

<item
  name="NET_ASTERIX"
  value="*"
  type="String"
/>
<item
  name="NET_COLUMN_NO_DELIMETER1"
  value="["
  type="String"
/>
<item
  name="NET_COLUMN_NO_DELIMETER2"
  value="]"
  type="String"
/>
<item
  name="NET_COLUMN_TXT_DELIMETER1"
  value="""
  type="String"
/>
<item
  name="NET_COLUMN_TXT_DELIMETER2"
  value="""
  type="String"
/>
<item
  name="NET_COLUMN_NAME"
  value="@name["
  type="String"
/>
<item
  name="NET_COLUMN_MAX"
  value="@max["
  type="String"
/>
<item
  name="NET_COLUMN_MIN"
  value="@min["
  type="String"
/>
<item
  name="NET_COLUMN_MIN"
  value="@min["
  type="String"
/>
<item
  name="NET_COLUMN_DESC"
  value="@tooltip["
  type="String"
/>
<item
  name="NET_HELP_PAGE_URL"
  value="http://www.informatik.uni-stuttgart.de/ipvs/bv/projekte/nld/
software/da/antwebvis/help/help_en/"
  type="String"
/>
<!-- commands -->
<item
  name="NET_CMD_LISTFILES"
  value="listfiles"
  type="String"
/>

```

```

<item
  name="NET_CMD_GETPROPERTIES"
  value="getfileproperties"
  type="String"
/>
<item
  name="NET_CMD_GETFILE"
  value="getfile"
  type="String"
/>
<item
  name="NET_CMD_GNUPLOT"
  value="gnuplot"
  type="String"
/>
<item
  name="NET_OPT_RECURSIVE"
  value="-r"
  type="String"
/>
<item
  name="NET_OPT_ASCII"
  value="/a"
  type="String"
/>
<item
  name="NET_OPT_PROPERITES"
  value="/p"
  type="String"
/>
<item
  name="NET_OPT_BINARY"
  value="/b"
  type="String"
/>
<item
  name="NET_DATAFILES"
  value=".tna"
  type="String"
/>
<item
  name="NET_CID_GETCOMMENT"
  value="10"
  type="int"
/>
<item
  name="NET_CID_GETIMAGE"
  value="11"
  type="int"
/>
<item
  name="NET_CID_GNUPLOT"
  value="12"
  type="int"
/>
</section>
<!-- ***** end networking-config ***** -->

<!-- ***** start gui-config ***** -->
<section>
  <item
    name="GUI_WIDTH"

```

```

        value=" 600 "
        type=" int "
    />
<item
    name=" GUI_HEIGHT "
    value=" 700 "
    type=" int "
/>
<item
    name=" GUI_DEF_DYNAMIC_BG "
    value=" 0#0#0 "
    type=" Color "
/>
<item
    name=" GUI_DEF_EDGE_COLOR "
    value=" 127#127#127 "
    type=" Color "
/>
    <item
        name=" GUI_DEF_POINT_COLOR "
        value=" 255#0#0 "
        type=" Color "
    />
    <item
        name=" GUI_DEF_FONT_COLOR "
        value=" 200#200#200 "
        type=" Color "
    />
<item
    name=" GUI_DEF_GRIDCOUNT "
    value=" 5 "
    type=" int "
/>
<item
    name=" GUI_DEF_GRIDCOUNT_MIN "
    value=" 2 "
    type=" int "
/>
<item
    name=" GUI_DEF_GRIDCOUNT_MAX "
    value=" 50 "
    type=" int "
/>
<item
    name=" GUI_POINT_TIME "
    value=" 100 "
    type=" int "
/>
<item
    name=" GUI_DEF_AZIMUTH "
    value=" 35 "
    type=" int "
/>
<item
    name=" GUI_DEF_ELEVATION "
    value=" 30 "
    type=" int "
/>
<item
    name=" GUI_DEF_ZOOM "
    value=" 3 "
    type=" float "

```

```

/>
  <item
    name="GUI_DEF_ZOOMMUL"
    value="0.01"
    type="float"
  />
  <item
    name="GUI_DRAWING_STYLES"
    value="dots#lines#dots_and_lines#smooth_lines_and_dots#dotted"
    type="String[]"
  />
  <item
    name="GUI_GNUPLOT_STYLES"
    value="lines#points#linespoints#impulses#dots#steps#errorbars#xerrorbars#
      yerrorbars#xyerrorbars#boxes#boxerrorbars#boxxyerrorbars"
    type="String[]"
  />
  <item
    name="GUI_COLOR_SCHEMES"
    value="same_color#position#velocity#acceleration"
    type="String[]"
  />
  <item
    name="GUI_DEF_IN_BORDER"
    value="LineBorder#0#0#0"
    type="Border"
  />
  <item
    name="GUI_DEF_OUT_BORDER"
    value="EtchedBorder#0#0#0"
    type="Border"
  />
</section>
<!-- ***** end gui-config ***** -->

<!-- ***** start text-config ***** -->
<section>
  <item
    name="TXT_TAB_SELECT"
    value="Select"
    type="String"
  />
  <item
    name="TXT_TAB_HELP"
    value="Help"
    type="String"
  />
  <item
    name="TXT_TAB_TRANSFER"
    value="Transfer"
    type="String"
  />
  <item
    name="TXT_NO_DESCR"
    value="sorry ,_no_description_available._try_a_child_node"
    type="String"
  />
  <item
    name="TXT_NO_IMAGE"
    value="sorry ,_cannot_display_image!"
    type="String"
  />

```

```

/>
<item
  name="TXT_NO_DATAFILE"
  value="sorry ,_no_datafile_selected_to_view!"
  type="String"
/>
<item
  name="TXT_SELECT_COLS"
  value="ok"
  type="String"
/>
<item
  name="TXT_TRANSFER_STOPPED"
  value="Stopped"
  type="String"
/>
<item
  name="TXT_NO_FILEDESC"
  value="Sorry ,_no_descriptions_available_yet!"
  type="String"
/>
<item
  name="TXT_AXIS"
  value="axis"
  type="String"
/>
<item
  name="TXT_STYLE"
  value="style"
  type="String"
/>
<item
  name="TXT_AUTORANGE"
  value="autorange"
  type="String"
/>
<item
  name="TXT_COL_SELECTION"
  value="Column_selection"
  type="String"
/>
<item
  name="TXT_ANIMATION_CONTOL"
  value="Animation_Control"
  type="String"
/>
<item
  name="TXT_COLOR_CONTOL"
  value="Colorstyle"
  type="String"
/>
<item
  name="TXT_DRAWING_CONTOL"
  value="Drawing-Style"
  type="String"
/>
<item
  name="TXT_HELPERS_CONTOL"
  value="Display-Helpers"
  type="String"
/>
<item

```

```

        name="TXT_STEP_CONTOL"
        value="Step-Chooser"
        type="String"
    />
<item
    name="TXT_DELAY"
    value="Delay_(ms)"
    type="String"
/>
<item
    name="TXT_OF"
    value="_of_"
    type="String"
/>
<item
    name="TXT_INIT_DESCR_TEXT"
    value="please_select_a_system_in_the_tree_above"
    type="String"
/>
<item
    name="TXT_LAUNCHER_BUTTON"
    value="start_visualisation"
    type="String"
/>
<item
    name="TXT_GNUPLOTT_BUTTON"
    value="create_image_on_server"
    type="String"
/>
<item
    name="TXT_LOADING_FILELIST"
    value="please_wait,_transferring_file-list"
    type="String"
/>
<item
    name="TXT_LOADING_DESC"
    value="please_wait,_transferring_description"
    type="String"
/>
<item
    name="TXT_LOADING_FILELIST_WAIT"
    value="please_wait_while_file-list_is_transferred"
    type="String"
/>
<item
    name="TXT_LAUNCH_CHOOSE"
    value="please_choose_a_view"
    type="String"
/>
<item
    name="TXT_LAUNCH_DYNAMIC"
    value="create_new_dynamic_view"
    type="String"
/>
<item
    name="TXT_LAUNCH_STATIC"
    value="create_new_static_view"
    type="String"
/>
<item
    name="TXT_LAUNCH_FILE"
    value="create_new_file_view"

```

```

        type=" String "
    />
</item>
    name="TXT_HELPER_BB"
    value=" Bounding_Box"
    type=" String "
    />
</item>
    name="TXT_HELPER_KO"
    value=" Coordinate_System"
    type=" String "
    />
</item>
    name="TXT_HELPER_XZPLANE"
    value=" Plane_XZ"
    type=" String "
    />
</item>
    name="TXT_HELPER_XYPLANE"
    value=" Plane_XY"
    type=" String "
    />
</item>
    name="TXT_HELPER_YZPLANE"
    value=" Plane_YZ"
    type=" String "
    />
</item>
    name="TXT_HELPER_GRIDBOX"
    value=" GridBox"
    type=" String "
    />
</item>
    name="TXT_COLUMN_NAME"
    value=" column_"
    type=" String "
    />
</item>
    name="TXT_PLZ_WAIT"
    value=" loading_-_please_wait!"
    type=" String "
    />
</item>
    name="TXT_NONE_ROW"
    value=" none"
    type=" String "
    />
</item>
    name="TXT_NAN"
    value=" sorry ,_cannot_convert_to_a_number"
    type=" String "
    />
</item>
    name="TXT_SELECT_SUBMIT"
    value=" please_select_which_rows_to_plot_and_enter_ranges"
    type=" String "
    />
</item>
    name="TXT_TRANSFER_STOP"
    value=" _Stop!"
    type=" String "
    />
</item>

```

```

<item
  name="TXT_TRANSFER_DONE"
  value="Done!"
  type="String"
/>
<item
  name="TXT_PARAMETER_RESET"
  value="reset_view"
  type="String"
/>
<item
  name="TXT_CENTER_ORIGIN"
  value="center_and_scale!"
  type="String"
/>
<item
  name="TXT_RESTORE_ORIGIN"
  value="restore_view"
  type="String"
/>
<item
  name="TXT_HELPER_SELECT"
  value="select_helper"
  type="String"
/>
<item
  name="TXT_HELP_UNAVAILABLE"
  value="Help-Page_is_not_available!\r\nPlease_inform_martin@kunzelnick.
    de"
  type="String"
/>
</section>
<!-- ***** end text-config ***** -->
</configdata>

```

Anhang D

Java Web Start

Beispiel für eine Java Web Start-Datei (JNLP) wie Sie für diese Diplomarbeit verwendet wird:

```
<jnlp codebase="http://www.kunzelnick.de/da/antwebvis/"
      href="http://www.kunzelnick.de/da/antwebvis/webstart_en.jnlp">
  <information>
    <title>AnT online visualisation</title>
    <vendor>University of Stuttgart / Martin Kunzelnick</vendor>
    <homepage href="http://www.kunzelnick.de/da/" />
    <description>Visualisation for AnT-files</description>
  </information>
  <security>
    <j2ee-application-client-permissions />
  </security>
  <resources>
    <j2se version="1.2+" />
    <jar href="antvis.jar" />
  </resources>
  <application-desc main-class="AntWebVis">
    <param name="configURI" value="http://www.kunzelnick.de/da/antwebvis/config/
      config_en.xml" />
  </application-desc>
</jnlp>
```

Anhang E

Evaluation

E.1 Fragebogen

Die folgenden Screenshots (Abbildungen E.1 bis E.12) zeigen die verschiedenen Bereiche des Online-Fragebogens.

Die Auswertung dieser Online-Umfrage erfolgt anonym. Bitte beantworten Sie die Fragen so präzise wie möglich. Getestet werden nicht Sie, sondern die Software - Ihr Feedback hilft uns, Schwierigkeiten aufzudecken, und gibt uns die Chance, die Software zu verbessern.

Die Bearbeitung des Fragebogens erfordert ca. 10 Minuten.
Vielen Dank!

Abbildung E.1: Einleitung

Abschnitt 1 von 5: Persönliche Angaben

Geschlecht: männlich weiblich

Alter: Jahre

Tätigkeitsbereich:

Entwickeln Sie selbst Software? ja nein

Abbildung E.2: Persönliche Angaben

Abschnitt 2 von 5: Vorwissen

Arbeiten Sie mit dynamischen Systemen? ja nein

Haben Sie AnT bereits verwendet? ja nein

Haben Sie schon die Visualisierung von AnT benutzt? ja nein

Abbildung E.3: Vorwissen

Abschnitt 3 von 5: Angaben zum verwendeten System / Performance

Wie viele Prozessoren hat Ihr Computer?

Mit welcher Taktfrequenz arbeitet der Computer?

Welches Betriebssystem verwenden Sie?

Welche Java-Version verwenden Sie?

Verwenden Sie JavaWebstart ja nein

Über welche Internet-Anbindung verfügen Sie?

trifft nicht zu trifft voll zu

Der Start des Applets war relativ schnell für ein Java-Applet.

Die Bedienung der Oberfläche war durchgehend flüssig und ohne größere Wartezeiten möglich.

Die Animation (Einblenden der Punkte nacheinander) war ausreichend schnell.

Die Übertragung der dynamischen Daten war ausreichend schnell.

Die Generierung und Übertragung der Bilder in der statischen Visualisierung war ausreichend schnell.

Abbildung E.4: System

Abschnitt 4 von 5: Bewertung des Applets

Aufgabenangemessenheit

Dieser Abschnitt dient dazu, festzustellen, ob das Applet Sie bei der Erledigung Ihrer Aufgaben unterstützt, oder ob die Bedienung unnötig, umständlich oder zu schwer ist :

trifft nicht zu trifft voll zu

Das Applet verfügt über alle von Ihnen benötigten Funktionen und Sie mussten sich keiner Tricks und Umwege bedienen, um Ihre Ziele zu erreichen.

Die erzielten Arbeitsergebnisse rechtfertigen den Aufwand, der zu ihrer Erreichung nötig war.

Die Dialoge verlangten keine überflüssigen Einstellungen oder Eingaben und waren nicht unnötig kompliziert.

Platz für Anmerkungen und Anregungen.
Wie denken Sie, könnte man die Aufgabenangemessenheit verbessern?

Abbildung E.5: Subjektive Bewertung

Selbstbeschreibungsfähigkeit

Ein Dialog ist selbstbeschreibungsfähig, wenn jeder Dialogschritt unmittelbar verständlich ist und Ihnen klar ist, was die Anwendung tut, was als nächste Eingabe erwartet wird und was die Folgen sein werden. Wo dies nicht der Fall ist, muss Hilfe verfügbar sein.

trifft nicht zu trifft voll zu

Die Dialoge sind übersichtlich und nicht unnötig kompliziert aufgebaut, und alle Elemente sind unmissverständlich benannt.

Sie mussten entweder die Hilfe nie konsultieren oder deren Inhalt war hilfreich und ausreichend.

Die Meldungen und Reaktionen des Systems waren für Sie immer verständlich.

Platz für Anmerkungen und Anregungen.
Wie denken Sie, könnte man die Selbstbeschreibungsfähigkeit verbessern?

Abbildung E.6: Selbstbeschreibungsfähigkeit

Erwartungskonformität

Dialoge innerhalb eines Systems müssen konsistent sein und sollen sich an an allgemeingültige und anerkannte Konventionen halten.

trifft nicht zu trifft voll zu

Das Verhalten diese Applets entsprach Ihren Erwartungen (bezogen auf bisherige Erfahrungen mit anderen Anwendungen in vergleichbaren Aufgabenstellungen).

Die Dialoge innerhalb dieses Applets sind konsistent zueinander.

Die Antwortzeiten des Applets waren nicht größer als Sie erwartet hätten (Sie hatten nie den Eindruck, dass das Applet abgestürzt ist oder nicht mehr reagiert).

Platz für Anmerkungen und Anregungen.
Wie denken Sie, könnte man die Erwartungskonformität verbessern?

Abbildung E.7: Erwartungskonformität

Fehlerrobustheit

Ein Dialog ist fehlerrobust, wenn das angestrebte Arbeitsergebnis trotz falscher Benutzereingaben mit minimalem Korrekturaufwand erreichbar ist. Der jeweilige Dialog soll den Benutzer auch von falschen Eingaben unterrichten und möglichst verständliche und hilfreiche Korrekturvorschläge machen.

Wie viele Fehler traten bei der Bedienung auf.

trifft nicht zu trifft voll zu

Der Grund für die Fehler wurde gut beschrieben, und die Ursache war dannach klar erkennbar.

Fehler ließen sich schnell und einfach korrigieren.

Platz für Anmerkungen und Anregungen.
Wie denken Sie, könnte man die Fehlerrobustheit verbessern?

Abbildung E.8: Fehlerrobustheit

Lernförderlichkeit

Ein System ist lernförderlich, wenn es den Benutzer beim Erlernen des Dialogsystems unterstützt und anleitet.

trifft nicht zu trifft voll zu

Das Experimentieren und Herumprobieren im Applet ist einfach und Sie haben keine Bedenken, Schaden anzurichten.

Nach kurzer Zeit war Ihnen die Bedienung des Applets klar und Sie konnten ihre Ziele effizient umsetzen.

Platz für Anmerkungen und Anregungen.
Wie denken Sie, könnte man die Lernförderlichkeit verbessern?

Abbildung E.9: Lernförderlichkeit

Steuerbarkeit

Ein Dialog ist steuerbar, wenn es dem Benutzer möglich ist, auf den Dialogablauf Einfluss zu nehmen. Der Benutzer soll Reihenfolge und Geschwindigkeit beeinflussen können.

trifft nicht zu trifft voll zu

Sie können die Dialoge steuern, d. h. die Geschwindigkeit des Ablaufs, die Reihenfolge usw. beeinflussen.

Das Wiederaufnehmen unterbrochener Aktionen ist möglich und ausgeführte Aktionen können rückgängig gemacht werden.

Platz für Anmerkungen und Anregungen.
Wie denken Sie, könnte man die Steuerbarkeit verbessern?

Abbildung E.10: Steuerbarkeit

Individualisierbarkeit

Ein Dialog ist individualisierbar, wenn er an die Bedürfnisse des Benutzers und an die Erfordernisse der Arbeitsaufgabe angepasst werden kann.

trifft nicht zu trifft voll zu

Die verschiedenen Einstellungsmöglichkeiten (z. B. Systemwahl, Fenstergröße, Helper, Animationsgeschwindigkeit usw.) sind ausreichend.

Die verschiedenen Einstellungsmöglichkeiten (siehe oben) sind sinnvoll.

Platz für Anmerkungen und Anregungen.
Wie denken Sie, könnte man die Individualisierbarkeit verbessern?

Abbildung E.11: Individualisierbarkeit

Abschnitt 5 von 5: Abschließende Bemerkungen

Würden Sie AnT-Online zukünftig wieder verwenden? ja nein

Konnte AnT-Online Ihr Interesse für AnT wecken? ja nein

Platz für Wünsche und Verbesserungsvorschläge für AnT-Online?

Fragebogen fertiggestellt

Wenn Sie alle Fragen beantwortet haben, senden Sie bitte die Daten über diesen Button. Nach dem Senden gibt es keine Korrekturmöglichkeit mehr!

Daten senden!

Abbildung E.12: Abschluss

E.2 Ergebnisse

Die Ergebnisse der Evaluation, die mittels des oben gezeigten Fragebogens gewonnen wurden, finden Sie auf der Website zu dieser Diplomarbeit. Bis zum 28. Juni 2004 lagen die folgende Daten vor:

Anzahl der Tester	7
Geschlecht (0=männlich, 1=weiblich)	0,14
Durchschnittsalter der Tester (in Jahren)	25,7
Anzahl der AnT-Benutzer	0
Softwareentwickler (0=nein, 1=ja)	0,86
Arbeiten Sie mit dynamischen Systemen? (0=nein, 1=ja)	0,14
AnT-Benutzer (0=nein, 1=ja)	0,0
AnTVis-Benutzer (0=nein, 1=ja)	0,0
Durchschnittliche Prozessoranzahl	1,0
Meistgenannter Taktfrequenzbereich	2,5-3,0GHz (43%)
Meistgenanntes Betriebssystem	Windows (100%)
Meistgenannte Javaversion	1.4.0 (71%)
Java Web Start-Benutzer (0=nein, 1=ja)	0,14
Meistgenannter Bandbreitenbereich	128kBit/s- 1,5MBit/s (100%)
Interesse für AnT geweckt (0=nein, 1=ja)	0,42
Weitere Verwendung von AnT-online (0=nein, 1=ja)	0,57
Durchschnittliche Anzahl von Fehlern	0,86
Frage (siehe Fragebogen)	Ø
Der Start des Applets war relativ schnell für ein Java-Applet.	1,14
Die Bedienung der Oberfläche war durchgehend flüssig und ohne größere Wartezeiten möglich.	1,43
Die Animation (Einblenden der Punkte nacheinander) war ausreichend schnell.	0,71
Die Übertragung der dynamischen Daten war ausreichend schnell.	0,71
Die Generierung und Übertragung der Bilder in der statischen Visualisierung war ausreichend schnell.	1,14

Frage (siehe Fragebogen)	Ø
Das Applet verfügt über alle von Ihnen benötigten Funktionen und Sie mussten sich keiner Tricks und Umwege bedienen, um Ihre Ziele zu erreichen	1,57
Die erzielten Arbeitsergebnisse rechtfertigen den Aufwand, der zu ihrer Erreichung nötig war.	1,29
Die Dialoge verlangten keine überflüssigen Einstellungen oder Eingaben und waren nicht unnötig kompliziert.	1,43
Die Dialoge sind übersichtlich und nicht unnötig kompliziert aufgebaut, und alle Elemente sind unmissverständlich benannt.	0,43
Sie mussten entweder die Hilfe nie konsultieren oder deren Inhalt war hilfreich und ausreichend.	1,28
Die Meldungen und Reaktionen des Systems waren für Sie immer verständlich.	1,71
Das Verhalten diese Applets entsprach Ihren Erwartungen (bezogen auf bisherige Erfahrungen mit anderen Anwendungen in vergleichbaren Aufgabenstellungen).	1,57
Die Dialoge innerhalb dieses Applets sind konsistent zueinander.	1,29
Die Antwortzeiten des Applets waren nicht größer als Sie erwartet hätten (Sie hatten nie den Eindruck, dass das Applet abgestürzt ist oder nicht mehr reagiert).	1,86
Der Grund für die Fehler wurde gut beschrieben, und die Ursache war dannach klar erkennbar.	0,71
Fehler ließen sich schnell und einfach korrigieren.	1,42
Das Experimentieren und Herumprobieren im Applet ist einfach und Sie haben keine Bedenken, Schaden anzurichten.	2,0
Nach kurzer Zeit war Ihnen die Bedienung des Applets klar und Sie konnten ihre Ziele effizient umsetzen.	1,57
Sie können die Dialoge steuern, d. h. die Geschwindigkeit des Ablaufs, die Reihenfolge usw. beeinflussen.	1,71
Das Wiederaufnehmen unterbrochener Aktionen ist möglich und ausgeführte Aktionen können rückgängig gemacht werden.	0,86
Die verschiedenen Einstellungsmöglichkeiten (z. B. Systemwahl, Fenstergröße, Helper, Animationsgeschwindigkeit usw.) sind ausreichend.	1,42
Die verschiedenen Einstellungsmöglichkeiten (siehe oben) sind sinnvoll.	1,0

In dieser Auswertung repräsentieren, wenn nicht anders angegeben, die ganzzahligen Werte von -2 bis 2 die Antworten „trifft nicht zu“ bis „trifft voll zu“.

Index

- 3D Anywhere, 96
- 3D Groove, 96
- 3DMeNow / XPlayer, 97
- 3DX Player, 97
- 3Space, 97

- Activate3D, 98
- Adobe Atmosphere, 98
- Anark, 98
- Anfy3D, 99
- ASP.NET, 47
- AWT, 31
- Axel, 99

- B3D, 99
- Bildschirm-Transformation, 46
- Blaxxun Virtual Worlds Platform, 100

- CA, 27
- Callback-Methoden, 33
- CharActor, 100
- configuration, 65
- Cortona, 100
- Cosmo Player, 101
- Critical Reach, 101
- Cult 3D, 101
- Cybercore Entrance, 101

- datahandling, 65
- DataOutputStream, 30
- dynamicViewPanel, 65

- Einheitswürfel, 46

- EON, 102
- errorhandling, 65
- event dispatcher thread, 31
- eViewer, 102
- Extreme Programming, 48
- Extreme Programming, Grundwerte, 49

- filePanel, 65
- Flash, 102
- Flash, detaillierte Bewertung, 16
- Flatland Rover / 3DML, 102
- Fluchtpunkt, 40

- GridBagLayout, 32
- gui, 65

- helpPanel, 65
- Hypercosm 3D Player, 103
- Hypersonic SQL, 113

- InputStream, 30
- Interbase, 113
- iPix, 103

- jarsigner, 27
- Java, detaillierte Bewertung, 14
- Java3D, detaillierte Bewertung, 15

- keytool, 27
- Koordinaten, homogene, 35
- Koordinatensysteme, globales, 38
- Koordinatensysteme, lokales, 38
- Koordinatensysteme, transformation, 38

Miner3D, 104
 Modellierung, 41
 MySQL (4.1), 111

 network, 66
 Netzwerkprogrammierung, 28
 NPD online survey, 14

 Objects to See, 104
 Openworlds, 104

 Parallelprojektion, 38
 PHP, 47
 planb media, 104
 Port, 28
 PostgreSQL, 112
 Projektion, perspektivische, 38
 Projektionen, 38
 Projektionstransformation, 45
 Pulse 3D, 105

 Quest3D, 105
 Quicktime VR, 105

 RichFX, 106
 Rotation, 36
 Runnable, 28

 SAX, 32
 SCOL, 106
 selectPanel, 66
 ServerSocket, 29
 Shockwave, 106
 Shockwave, detaillierte Bewertung, 18
 Shout3D, 106
 Sicht-Transformation, 41
 Sichtvolumen, 43
 Sichtvolumen, kanonisches, 44
 Sichtvolumen-Transformation, 43
 Skalierung, 37
 Skalierung, uniforme, 37
 SPX/Open SPX, 107
 staticViewPanel, 66

 Stream, 30
 SVG, detaillierte Bewertung, 19
 Swing, 31
 SwingWorker, 31
 synthetisches Kameramodell, 42

 Thread, 27
 transferPanel, 66
 Transformationen, 36
 Transformationen, Kombination von, 40
 Translation, 37

 utilities, 66

 View-Reference-Coordinate-System, 42
 Viewing-Pipeline, 40
 Viewpoint Media Player, 107
 Virtools, 108
 Virtue3D, 108
 VizStream, 109
 VRML, 109
 VRML, detaillierte Bewertung, 19

 Web of Trust, 27
 WF-3D, 110
 WildTangent, 110

 X3D, 111

 ZAP, 111

Abbildungsverzeichnis

2.1	NPD online survey vom März 2004	13
2.2	Resultate der Evaluation	21
4.1	Übersicht der Projektionsarten	39
4.2	Anzahl der Fluchtpunkte bei der perspektivischen Projektion	39
4.3	Weg von 3D-Welt- zu 2D-Bildkoordinaten	40
4.4	Aufbau des Sichtkoordinatensystems	41
4.5	Sichtvolumen bei perspektivischer Projektion	43
4.6	Transformation der Mitte des Sichtvolumens auf die z-Achse und in das kanonische Sichtvolumen	45
4.7	Überführen des Pyramidenstumpfes in einen Quader	45
5.1	Abhängigkeiten bei der Softwareentwicklung	50
6.1	Steuerungsoberfläche des Prototyps	52
6.2	2D-Panel des Prototyps	53
6.3	3D-Panel des Prototyps	53
6.4	Select-Use-Case	56
6.5	2D	57
6.6	3D	57
6.7	Spezifikation des Select-Panels	58
6.8	Spezifikation des Transfer-Panels	59
6.9	Spezifikation des Image-Panels	60
6.10	Spezifikation des 3D-Viewer-Panels	61
6.11	Grobübersicht	63
6.12	Paketübersicht zu AnT-Online	64
6.13	Screenshot des fertigen Selection-Panels	75
6.14	Screenshot des fertigen Transfer-Panels	75
6.15	Screenshot des fertigen Help-Panels	76
6.16	Screenshot des fertigen 3D-Panels	77
6.17	Screenshot des fertigen Image-Panels	77

6.18	Screenshot des fertigen File-Panels	78
7.1	Gefundene Probleme im Verhältnis zur Testeranzahl	85
7.2	Datenbankschema	87
7.3	Angestrebte dynamische Visualisierung	90
7.4	Angestrebte statische Visualisierung	91
B.1	Package-Übersicht	114
B.2	configuration	115
B.3	datahandling	116
B.4	errorhandling	117
B.5	dynamicViewPanel	118
B.6	filePanel	119
B.7	helpPanel	119
B.8	selectPanel	120
B.9	staticViewPanel	121
B.10	transferPanel	122
B.11	network	123
B.12	utilities	124
E.1	Einleitung	136
E.2	Persönliche Angaben	136
E.3	Vorwissen	137
E.4	System	137
E.5	Subjektive Bewertung	138
E.6	Selbstbeschreibungsfähigkeit	138
E.7	Erwartungskonformität	139
E.8	Fehlerrobustheit	139
E.9	Lernförderlichkeit	140
E.10	Steuerbarkeit	140
E.11	Individualisierbarkeit	141
E.12	Abschluss	141

Abkürzungsverzeichnis

ACID = Atomicity, Consistency, Isolation, Durability

ADO = Active Data Objects

AnT = A simulation and Analysis Tool for Dynamical Systems

API = Application Programming Interface

ASCII = American Standard Code for Information Interchange

ASP = Active Server Pages

AWT = Abstract Window Toolkit

BP = Back Plane

CA = Certification Authority

CAD = Computer Aided Design

COM = Component Object Model

CW = Center of View-Window

DC = Device Coordinate System / Gerätekkordinatensystem

DHTML = Dynamic Hypertext Markup Language

DOM = Document Object Model

ECMA = European Computer Manufacturers Association

ECPG = Embedded SQL C Preprocessor

FP = Front Plane

GUI = Graphical User Interface

HTML = Hypertext Markup Language

HTTP = Hypertext Transfer Protocol
IE = Internet Explorer
JAR = Java Archive
JDBC = Java Database Connectivity
JFC = Java Foundation Classes
JNLP = Java Network Launching Protocol
JS = JavaScript
MC = Model Coordinate System / Model-Koordinatensystem
NPC = Normalized Projection Coordinate System
NRP = Normal Reference Point / Eyepoint / Betrachterstandpunkt
NS = Netscape
ODBC = Open Database Connectivity
OpenGL = Open Graphics Language
PC = Personal Computer
PDF = Portable Document Format
PHP = PHP: Hypertext Preprocessor
PNG = Portable Network Graphics
PRP = Projection Reference Point / Augenpunkt
RE / RTE = Runtime Environment
RUP = Rational Unified Process
SAX = Simple API for XML-Parsing
SDK = Software Development Kit
SMIL = Synchronized Multimedia Integration Language
SOAP = Simple Object Access Protocol
SQL = Structured Query Language
SSL = Secure Sockets Layer
SVG = Scalable Vector Graphics

TCL = Tool Command Language
UCS = Universal Character Set
URI = Universal Resource Identifier
URL = Universal Resource Locator
UTF-8 = 8-bit Unicode Transformation Format
VB = Visual Basic
VP = View Plane
VPN = View-Plane-Normal-Vector
VRC = View Reference Coordinate-System
VRML = Virtual Reality Modeling Language
VRP = View Reference Point / Referenzpunkt / Blickrichtung
VM = Virtual Machine
VUV = View Up Vector / vertikale Orientierung
W3C = World Wide Web Consortium
WCS = World Coordinate System / Weltkoordinatensystem
WWW = World Wide Web
X3D = Extensible 3D
XGL = Window Graphics Library
XML = Extensible Markup Language

Glossar

ActiveX

Von Microsoft entwickelte Technologie, mit der sich, vergleichbar zu Java oder der Plugin-Schnittstelle von Netscape, der Internet-Explorer nahezu beliebig um neue Komponenten erweitern lässt. Anders als bei Java werden ActiveX- Komponenten nicht in einer Art Sandbox ausgeführt, sondern laufen direkt auf dem (Windows-) System des Benutzers, was neben relativ hoher Performance natürlich auch erhebliche Sicherheits- und Datenschutzprobleme mit sich bringt, da somit der volle Zugriff auf das System des Benutzers möglich ist.

Advanced Window Toolkit

Standard-API innerhalb der Java Foundation Classes zur Implementierung plattformunabhängiger grafischer Benutzerschnittstellen (GUI).

Affine Transformation

Eine Transformation, bei der die Längen- und Winkelverhältnisse gewahrt bleiben.

API

Das Application Programming Interface ist eine Schnittstelle, über die ein Softwaresystem anderen Programmen eigene Funktionalität zur Verfügung stellt. Eine API stellt die Mittel bereit, um Software zu erstellen.

Certification Authority

Eine vertrauenswürdige Instanz innerhalb einer Zertifizierungshierarchie, die digitale Zertifikate ausstellt, um damit digitale Unterschriften und Paare von öffentlichen und privaten Schlüsseln zu erstellen.

Community

Bezeichnet im Rahmen dieser Arbeit eine Gemeinschaft von Menschen mit gemeinsamen fachlichen Interessen. Da die Mitglieder ähnliche Informationen benötigen, werden Erfahrungen, Anleitungen, Ideen, Meinungen und so weiter über eine Plattform im Internet (Kombinationen von Websites, Foren, Chats, Groups, ...) ausgetauscht.

Dialog

Im Rahmen dieser Diplomarbeit bezeichnet Dialog einen oder mehrere Bestandteile der GUI.

DirectX

Microsofts DirectX stellt unter Windows-Betriebssystemen eine Sammlung von Multimedia-APIs dar. Vergleichbar zu OpenGL können damit in Echtzeit komplexe 3D-Darstellungen (zum Beispiel für Spiele) erzeugt werden. DirectX umfasst aber nicht nur APIs für Grafikdarstellungen (DirectX Graphics), sondern auch für die Wiedergabe, das Aufzeichnen und Verändern von Tönen (DirectSound, DirectMusic), zum Ansprechen von Eingabegeräten (DirectInput), zur Interaktion über ein Netzwerk (DirectPlay) oder zur Wiedergabe oder Verarbeitung von Video-Streams (DirectShow).

Freeware

Software, die zwar über ein Copyright geschützt sein kann, aber kostenlos verwendet werden darf.

Gnuplot

Gnuplot ist ein kommandozeilenorientiertes Programm für UNIX, IBM OS/2, MS Windows, DOS, Apple MacOS, VMS, Atari und viele andere Plattformen. Mit Gnuplot können Funktionen gezeichnet oder Daten aus Dateien geplottet werden. Gnuplot ist Freeware. Für mehr Informationen und Gnuplot zum Herunterladen siehe [73].

GridBagLayout-Manager

Sehr mächtiger und vielseitiger, aber auch komplexer Layout-Manager von Java (siehe auch [82]).

GUI

Grafische Benutzungsoberfläche.

Helper

Bezeichnet in dieser Arbeit die dreidimensionalen Objekte, die sich in die interaktive Visualisierung der Punktwolken einblenden lassen. Typische Helper sind das Koordinatensystem, eine Bounding-Box, Gitterebenen und vieles mehr.

Kanonisches Sichtvolumen

Bei der Parallelprojektion ist das kanonische Sichtvolumen der Einheitswürfel, bei der perspektivischen Projektion ist es der Einheitspyramidenstumpf. Alle Teile einer 3D-Szene, die im Sichtvolumen liegen, werden am Ende der Viewing-Pipeline auf das Gerätekoordinatensystem (DC) abgebildet.

Native Bindungen

Native Bindungen (englisch: native bindings) bezeichnet in diesem Dokument

die ab Java-Version 1.4 verfügbare Möglichkeit, Swing Benutzerschnittstellen-Elemente direkt auf Elemente des jeweiligen Systems, auf dem das Applet/die Applikation ausgeführt wird, abzubilden.

OpenGL

OpenGL ist die Spezifikation einer plattformunabhängigen API zur Entwicklung von 3D-Applikationen. Der OpenGL-Standard umfasst etwa 250 Befehle, die auf der entsprechenden Hardware das Real-Time-Rendern von komplexen 3D-Szenen erlauben. Üblicherweise wird OpenGL in der konkreten Implementierung vom Hersteller noch mit proprietären Erweiterungen versehen.

Plugin

Im Kontext dieser Arbeit eine Softwarekomponente, die in einen Browser integriert wird, um die Fähigkeiten des Browsers zu erweitern und um anderen Content anzuzeigen, als nur HTML, Text usw.

PNG

Grafikformat, das langfristig das GIF-Format ersetzen soll, da es nicht durch Patente belastet ist. PNG bietet alle Vorteile des GIF-Formats, verfügt aber auch über einige zusätzliche Vorzüge, wie etwa einen Alphakanal, Farbcodierung von 2-48 Bit, Farbkorrekturmechanismen, CRC- Prüfsummen, Tags und vieles mehr.

Proprietär

Bezeichnet in der Informatik traditionell Dateiformate, Protokolle, und so weiter, die nicht allgemein anerkannten Standards entsprechen und von einer Firma oder einem Konsortium selbst entwickelt wurden. Meist sind diese Dateiformate, Protokolle usw. auch nicht offen gelegt und durch Patente geschützt.

Rendering-Pipeline

siehe Viewing-Pipeline.

Runtime Environment (Java)

Laufzeitumgebung für Java-Software. Stellt die Funktionen und Komponenten des darunter liegenden Java-Frameworks zur Verfügung.

SAX

SAX ist eine programmiersprachenunabhängige Spezifikation einer API zum Lesen und zur Verarbeitung von XML-Daten. Die Spezifikation sieht verschiedene Methoden vor, wie zum Beispiel „startDocument“, „startElement“, „endElement“, „endDokument“ und andere. In Java existiert eine leere, eventbasierte SAX-Implementierung in der Klasse „org.xml.sax.helpers.DefaultHandler“. Von dieser Klasse kann einfach abgeleitet werden, dann müssen nur noch die entsprechenden Methoden selbst implementiert werden.

SMIL

Synchronized Multimedia Integration Language (ausgesprochen engl. SMILE) ist eine einfache, an HTML angelehnte Skriptsprache, die vom W3C spezifiziert wurde.

SVG

Scalable Vector Graphics ist ein vom W3C im September 2001 verabschiedeter Standard zur Speicherung von 2D-Vektorgrafiken in einem XML-Format. Siehe auch [66] und [67].

Synthetisches Kameramodell

Ein einfaches Modell, das Lage und Orientierung der Projektionsebene, den Standpunkt der Kamera und den sichtbaren Ausschnitt der Szene definiert.

View-Plane-Normal-Vector

Auf Länge 1 normierter Vektor, der senkrecht zur Projektionsebene steht. Wird auch als w-Achse des Sichtkoordinatensystems bezeichnet.

View-Reference-Point

Punkt innerhalb der Projektionsebene (VP), auf den die synthetische Kamera zeigt.

View-Up-Vector

Er definiert zusammen mit seinem Ursprung, dem Referenzpunkt, was auf der Projektionsebene oben liegt.

Viewing-Pipeline

Zusammenfassende Bezeichnung aller Schritte zur Umrechnung von den Modell-Koordinaten bis hin zu den Geräte-Koordinaten.

VRML

Die Virtual Reality Modeling Language ist eine Beschreibungssprache für 3D-Szenen. Es ist damit möglich, Geometrien, Beleuchtung, Bewegungen, Kameras und Möglichkeiten der Interaktion zu beschreiben. Die nötigen Daten sind als Plaintext (ASCII) in .wrl-Dateien abzulegen. Der Nachfolger von VRML soll X3D werden, dessen Datenformat auf XML basiert. Zum Betrachten der Szenen wird ein entsprechendes Browser-Plugin benötigt, das zur Darstellung idealerweise mittels OpenGL oder DirectX auf eventuell vorhandene 3D-Hardware zurückgreift. VRML ist durch verschiedene ISO-Normen ab 1997 standardisiert.

W3C

Siehe World Wide Web Consortium.

Web of Trust

Das Web of Trust wird dadurch ermöglicht, dass Personen oder Organisationen öffentliche Schlüssel signieren, deren Echtheit sie überprüft und deren Inhaber sie

als vertrauenswürdig eingestuft haben. Vertraut man einer Person oder Organisation A, dann vertraut man über das Web of Trust implizit auch allen Personen oder Organisationen B, denen A vertraut. Somit muss man B nicht kennen und kann trotzdem sicher sein, dass B vertrauenswürdig ist.

World Wide Web Consortium

Ein international anerkanntes Gremium zur Standardisierung von Technologien, die das Internet betreffen. Beispiele hierfür, die auch in dieser Arbeit vorkommen, sind unter anderem:

- HTML/XHTML/XML
- CSS
- SMIL
- PNG
- SVG
- VRML

Gegründet wurde das W3C von Tim Berners-Lee („Erfinder“ des WWW), der es momentan auch leitet. [80].

X3D

Zukünftige, auf XML basierende Weiterentwicklung von VRML.

Literaturverzeichnis

- [1] 3D Ltd. Website, <http://www.3danywhere.com>
- [2] The Groove Alliance Website. <http://www.3dgroove.com>
- [3] biovirtual Website. <http://www.biovirtual.com>
- [4] Micronet Group Website. <http://www.micronet-japan.com>
- [5] Biovirtual Website. <http://www.biovirtual.com>
- [6] THE MICRONET GROUP. <http://www.micronet-japan.com/3dxmain.asp>
- [7] TGS Website. <http://www.tgs.com/>
- [8] Kaon Interactive Website, <http://www.kaon.com/>
- [9] Adobe Inc. Website, <http://www.adobe.com/products/atmosphere/>
- [10] Anark Website, <http://www.anark.com/>
- [11] Anfy team Website, <http://anfyteam.com/panfy3d.html>
- [12] Mindavenue Website, <http://www.mindavenue.com/en/products/>
- [13] brilliant digital Website, <http://www.brilliantdigital.com/>
- [14] blaxxun technologies Website, <http://www.blaxxun.com>
- [15] Charamel Software GmbH Website, <http://www.charamel.com/>
- [16] ParallelGraphics Website, <http://www.parallelgraphics.com/>
- [17] Cosmo Software Website, <http://ca.com/cosmo/>
- [18] Critical Reach Website, <http://www.criticalreach.com/>

- [19] Cycore Website, <http://www.cult3d.com/>
- [20] Cynosys Website, <http://www.cynosys.com/>
- [21] EON Reality, Inc. Website, <http://www.eonreality.com/>
- [22] Open CASCADE Website, <http://www.opencascade.org>
- [23] Flatland Online, Inc. Website, <http://www.flatland.com/>
- [24] Hypercosm LLC. Website, <http://www.hypercosm.com/>
- [25] IPIX Corp. Website, <http://www.ipix.com/>
- [26] Sun Microsystems, Inc. Website, <http://java.sun.com/>
- [27] J3D Website, <http://www.j3d.org/>
- [28] Eleco Software GmbH Website, <http://www.o2c.de>
- [29] OpenWorlds Inc. Website, <http://www.openworlds.com/>
- [30] plan_b media ag Website, <http://www.planb-media.de>
- [31] Pulse Website, <http://www.pulse3d.com/>
- [32] Act-3D Website, <http://www.quest3d.com>
- [33] Apple Computer, Inc. Website, <http://www.apple.com/quicktime/>
- [34] RichFX Inc. Website, <http://www.richfx.com/>
- [35] CryoNetworks Website, <http://www.cryonetworks.com/>
- [36] Eyematic Website, <http://www.eyematic.com/>
- [37] Scanbull Software GmbH Website, <http://www.scanbull.net>
- [38] Viewpoint Corp. Website, <http://www.viewpoint.com>
- [39] Virtools SA Website, <http://www.virtools.com/>
- [40] Virtue3d Website, <http://www.virtue3d.com/>
- [41] RealityWave Inc. Website, <http://www.realitywave.com/>
- [42] Demicron AB Website, <http://www.demicron.com/>
- [43] WildTangent Inc. Website, <http://www.wildtangent.com/>
- [44] Pointcloud Website, <http://www.pointcloud.com/>
- [45] Web3D Website, <http://www.web3d.org/>

- [46] Google Groups Website, <http://groups.google.de/>
- [47] Frank Lamprecht. 3D für das Web. Grundlagen und Beispiele für Flash, Director, DHTML und Java. Addison-Wesley, 2002
- [48] Klaus Zeppenfeld. Lehrbuch der Grafikprogrammierung. Grundlagen, Programmierung, Anwendung. Spektrum Akademischer Verlag, 2004
- [49] Christian Bunse, Antje von Knethen. Vorgehensmodelle Kompakt. Spektrum Akademischer Verlag, 2002
- [50] Richard S. Schifman, Günther Heinrich, Yvonne Heinrich. Animation und Interaktion im WWW mit Shockwave und Flash. Springer-Verlag, 1998
- [51] Christof Marte. Flash MX und PHP, MySQL und XML. Dynamische Webseiten generieren mit Flash. Galileo Press, 2002
- [52] Jon Barrilleaux. 3D user interfaces with java3d. A guide to computer-human interaction in three dimensions. Manning Publications, 2000
- [53] Dirk Louis, Svend Nissen. Flash MX und Actionscript. Webdesign und dynamisches Programmieren Markt und Technik, 2002
- [54] Daniel Selman. Java3D Programming. A guide to key concepts and effective techniques Manning Publications, 2000
- [55] Andrew Davison. Games Programming with Java and Java3D. <http://fivedots.coe.psu.ac.th/~ad/jg/intro/>, 2003
- [56] DATech - Deutsche Akkreditierungsstelle Technik e.V.. DATech-Prüfhandbuch Gebrauchstauglichkeit. Leitfaden für die softwareergonomische Evaluierung von Software auf Grundlage von DIN EN ISO 9241, Teile 10 und 11. Version 3.2 <http://www.datech.de>, 2004
- [57] Michael Herczeg. Software-Ergonomie. Grundlagen der Mensch-Computer-Kommunikation. Addison-Wesley, 1994
- [58] Europäisches Komitee für Normung. EN ISO 9241-10. Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten. Teil 10: Grundsätze der Dialoggestaltung. 1995
- [59] Dr. rer. nat. Matthias Ressel, UBS AG. Gastdozent am VIS Skript zur Vorlesung: Visualisierung und Interaktive Systeme (VIS). Uni Stuttgart, Wintersemester 2003/2004
- [60] Java 2 Platform, Standard Edition, v 1.4.2 API Specification. <http://java.sun.com/j2se/1.4.2/docs/api/index.html>, 2004

- [61] Wikipedia Website, <http://de.wikipedia.org>
- [62] Macromedia Inc. Website, <http://www.macromedia.com>
- [63] Macromedia Shockwave Website, <http://www.macromedia.com/software/shockwaveplayer/>
- [64] NPD Studie zur Verbreitung von Plugins, http://www.macromedia.com/software/player_census/flashplayer/
- [65] OpenSWF-Projekt Website, <http://www.openswf.org/>
- [66] Adobe svg Zone Website, <http://www.adobe.com/svg/>
- [67] W3C SMIL Website, <http://www.w3c.org/AudioVideo/>
- [68] Webbarometer Website, <http://www.webhits.de/deutsch/webstats.html>
- [69] webreference.com, <http://www.webreference.com/stats/browser.html>
- [70] PCWelt Website, <http://www.pcwelt.de/news/internet/29132/>
- [71] browserwatch von w3b.org, <http://www.w3b.org/trends/browserwatch.html>
- [72] Jakob Nielsen. Why You Only Need to Test With 5 Users, <http://www.useit.com/alertbox/20000319.html>
- [73] gnuplot Website, <http://www.gnuplot.info>
- [74] Kent Beck, Martin Fowler. Extreme Programming planen Addison-Wesley, 2001
- [75] Java ist auch eine Insel, <http://www.galileocomputing.de/openbook/javainsel3/>
- [76] Sun. Working with XML Website, <http://java.sun.com/xml/jaxp/dist/1.1/docs/tutorial/TOC.html>
- [77] Java API Documentation Website, <http://java.sun.com/j2se/1.4.2/docs/api/org/xml/sax/helpers/DefaultHandler.html>
- [78] Sun. How to Sign Applets Using RSA-Signed Certificates, http://java.sun.com/j2se/1.4.2/docs/guide/plugin/developer_guide/rsa_signing.html

- [79] Sun. Thread Tutorial, <http://java.sun.com/docs/books/tutorial/essential/threads/customizing.html>
- [80] World Wide Web Consortium Website, <http://www.w3c.org/>
- [81] Sun. Using the Swingworker-class, <http://java.sun.com/docs/books/tutorial/uiswing/misc/threads.html>
- [82] Sun. How to Use GridBagLayout, <http://java.sun.com/docs/books/tutorial/uiswing/layout/gridbag.html>

Erklärung

Hiermit versichere ich, diese Arbeit
selbständig verfasst und nur die
angegebenen Quellen benutzt zu haben.

(Martin S. Kunzelnick)