





Technische Fakultät
AG Bioinformatik

CELLmicrocosmos
- Interaktive 3D-ZellVisualisierung
Die Dokumentation zur
Masterarbeit von Björn Sommer
Technische Fakultät
AG Bioinformatik
Interdisziplinäre
Medienwissenschaften Bielefeld

Version 1.1 (31.08.2006)

CELL microcosmos

Interaktive 3D-ZellVisualisierung

1. Einleitung

- 2.1. Zielsetzung
- 2.2. Arbeitsschritte

2. Interaktive 3D-ZellVisualisierung

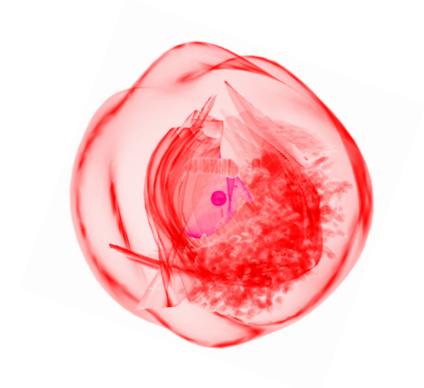
- 2.1. Integration der Vorarbeiten
 - 2.1.1. Bachelor
 - 2.1.2. Cm1
 - 2.1.3. Cm2
- 2.2. Einarbeitung in die Hepatozytenmembran
 - 2.2.1. Erstellung der Texte
 - 2.2.1.1. Recherche
 - 2.2.1.2. Auswahl der Texte
- 2.3. Programmierung und Animation
 - 2.3.1. Das amira®-Package
 - 2.3.2. TCL-Scripting
 - 2.3.2.1. Direkter Zugriff auf amira
 - 2.3.3. C++ Programmierung
 - 2.3.3.1. Integration der TCL-Scripte
 - 2.3.3.2. Integration der Texte
 - 2.3.3.2.1. Textdateien
 - 2.3.3.2.2. Sounds
 - 2.3.3.2.3. Modularer Aufbau
 - 2.3.4. Integration der Membran
 - 2.3.4.1. Membraneditor
 - 2.3.4.2. PDB-Recherche
 - 2.3.4.3. Erstellung der Membran
 - 2.3.4.4. Animation
 - 2.3.4.4.1. Erstellung eines Storyboards
 - 2.3.4.4.2. Animation unter amira®

4. Anhang

- 4.1. Technik
 - 4.2.1. Hardware
 - 4.2.2. Software
 - 4.2.3. Technologien
- 4.2. Abbildungen und Zusatzinformationen
- 4.3. Danksagungen
- 4.4. DisplayCell-Übersicht

5. Quellen

- 5.1. Literatur
- 5.2. Webadressen
- 5.3. Abbildungen
- 5.4. Typogramm
- 5.5. Warenzeichen



2.3.5. DisplayCell

- 2.3.5.1. Kurzbeschreibung
 - 2.3.5.2. Voraussetzungen
 - 2.3.5.3. Installation
 - 2.3.5.4. Erzeugung
 - 2.3.5.5. Bedienung
 - 2.3.5.5.1. Die Zelle
 - 2.3.5.5.2. Die Zellkomponenten
 - 2.3.5.6. Animationen und Texte
 - 2.3.5.6.1. Zellkomponenten
 - 2.3.5.6.2. Hepatozyt
 - 2.3.5.6.3. Plasmamembran
 - 2.3.5.7. Dokumentierung unter amira
- 2.4. Projekt-Dokumentation und die Internetpräsenz
- 2.4.1. Inhalte
 - 2.4.2. PHP-Programmierung
 - 2.4.3. Validierung
 - 2.4.4. Forum

3. Resultate

- 3.1. Neue Problemfelder und Lösungsansätze
- 3.2. Fazit
- 3.3. Ausblick



CELLmicrocosmos

1. Einleitung

1.1. Zielsetzung

Das Ziel der Masterarbeit *CELLmicrocosmos - Interaktive 3D-ZellVisualisierung* besteht in der Aufarbeitung, Weiterverarbeitung und Präsentation der aktuellen Ergebnisse des *CELLmicrocosmos*-Projektes.

Dabei soll zum einen die in Ansätzen bereits vorhandene 3D-Zellvisualisierungs-Umgebung unter amira® (*Cm1*) mit den Ergebnissen aus der Zellmembranvisualisierung (*Cm2*) verknüpft werden. Aus dieser Verknüpfung soll eine interaktive Lernumgebung entstehen, die es dem Benutzer ermöglichen soll, Einblick in den Aufbau einer Zelle zu erlangen. Exemplarisch soll dieses Konzept an der Zellmembran erprobt werden. Als Zielgruppe gelten hierbei Biologen, Mediziner und Sekundärstufe-Schüler, denen ein räumlicher Einblick in die Zelle vermittelt werden soll.

Zum anderen sollen die Ergebnisse der unterschiedlichen Projekte im Internet in ansprechender und übersichtlicher Weise dokumentiert bzw. präsentiert werden.

Die Resultate dieser Arbeit sollen eine gute Grundlage für zukünftige Projekte wie Bachelor- oder Masterarbeiten, Themen für Doktorandenprojekte oder Seminare schaffen.

Die medienwissenschaftliche Herausforderung dieser Arbeit besteht in der Integration wissenschaftlich gestalterischer, informationstechnischer und zellbiologischer Elemente in eine interaktive Computer-Umgebung und der Synthese dieser Elemente zu einem ausbaufähigen Gesamtkonzept.

1.2. Arbeitsschritte

Die Bestandteile dieser Masterarbeit lassen sich in folgende Arbeitsschritte untergliedern:

1.2.1. Die *Integration der Vorarbeiten*, bei welcher zunächst die Ergebnisse aus den vorhergehenden Projekten gesichtet werden und auf evtl. notwendige Modifikationen untersucht werden mussten, bildete die Grundlage für diese Arbeit.

1.2.2. Daraufhin musste die *Einarbeitung in die Hepatozytenmembran* erfolgen. Dabei handelte es sich um den größten theoretischen Bestandteil dieser Arbeit, bei welchem bereits vorhandene Texte mit der Literatur und diversen Internetquellen abgeglichen und grundsätzlich überarbeitet werden mussten.

1.2.3. Der größte praktischen Anteil dieser Arbeit folgte anschließend mit der *Programmierung und Animation* unter amira®. Hier mussten unterschiedliche Programmier- und Scriptsprachen angewendet und die Animationsmöglichkeiten von amira® ausgeschöpft werden.

1.2.4. Die *Projekt-Dokumentation und die Internetpräsenz* waren ein weiterer, praxisorientierter Bestandteil dieser Arbeit, bei welchem die Ergebnisse aus mehreren Jahren Arbeit an diesem Projekt auf übersichtliche Weise dargestellt werden sollten.



Abb. 2.1
Zelle aus dem blutbildenden Gewebe der Maus. Elektronenmikroskopische Aufnahme:

ER Endoplasmatisches Retikulum,
M Mitochondrium,
Me Zellmembran,
N Nucleus,
Po Kernporen,
V Vakuole

(Ude/Koch 1982,
S. 27)

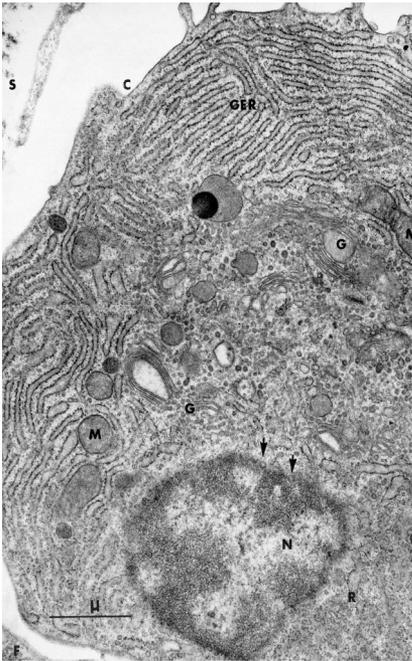


Abb. 2.2
Plasmazelle aus dem menschlichem Gewebe:

C Zellmembran schräg angeschnitten,
F Fibroplast,
G Golgiapparat,
GER Granulares Endoplasmatisches Retikulum,
M Mitochondrium,
N Nucleus,
R Ribosomen,
S Bindegewebe,
Pfeile zeigen auf nukleare Poren aus der Aufsicht

(Peter/Carr 1971,
S. 133)

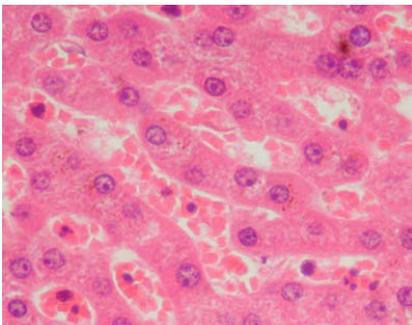


Abb. 2.3
Leber-Zelle einer Maus:

Färbung: Haematoxylin und Eosin
(School of Nursing 2004)

2. Interaktive 3D-ZellVisualisierung

2.1. Integration der Vorarbeiten

2.1.1. Bachelorarbeit: CELLmicrocosmos: ZellenVisualisierung

Die Grundlage für diese Masterarbeit legte im Jahr 2004 die Bachelorarbeit *CELLmicrocosmos: ZellenVisualisierung (Cm)*. Der wichtigste Bestandteil dieser Arbeit waren die Modellierung und Animation von 3D-Objekten mit dem Ziel, einen exemplarischen Einblick in die eukaryontische Zelle bieten zu können. Hierbei kamen die Programme *discreet® 3d Studio MAX®*, ein 3D-Animations-Programm, und *Adobe® Premiere®*, ein Programm für die digitale Filmproduktion, zum Einsatz. Als Grundlage dienten hierbei licht- und elektronenmikroskopische Aufnahmen, an Hand derer die zwei- und drei-dimensionale Struktur analysiert wurde.

Resultat war eine statische, grafisch anspruchsvolle Animation und die Modelle von acht Zellkomponenten, welche auch in dieser Arbeit verwendet wurden (Sommer 2004, S. 18 ff).

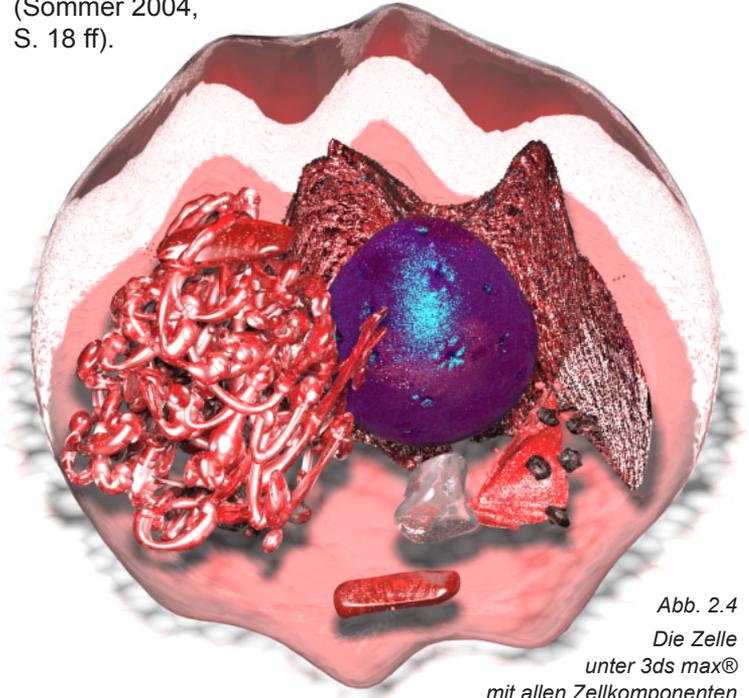


Abb. 2.4
Die Zelle unter 3ds max® mit allen Zellkomponenten

2. Interaktive 3D-ZellVisualisierung

2.1.2. Projekt CELLmicrocosmos I:

Interaktive Visualisierung biologischer Zellen

Bei dem Projekt *CELLmicrocosmos I: Interaktive Visualisierung biologischer Zellen (Cm1)*, welches im Sommersemester 2005 angeboten wurde, erfolgte mit drei Teilnehmern die Integration der o.g. 3D-Modelle in amira®. Zunächst wurde ein Zellvisualisierungs-Modul geschaffen, mit dessen Hilfe die bereits in der Bachelorarbeit (Cm) modellierten Objekte manipuliert werden konnten. Diese damalige Beta-Version des *DisplayCell*-Moduls bildete die Grundlage für die Programmierung im Rahmen dieser Masterarbeit (s. Abb. 2.5, 2.6).

Während dieses Seminars erfolgte desweiteren die erste Einarbeitung in den Aufbau einer Leberzelle und ein erster Text hierzu wurde verfasst, der als theoretische Grundlage eingesetzt werden konnte.

Darüberhinaus wurde ein Storyboard für Animationen entwickelt, welches ebenfalls im Rahmen der vorliegenden Arbeit Verwendung gefunden hat (s. 2.3.4.4.).

Abb. 2.5 + 2.6
Screenshots die während der Erstellung der ersten Version von *DisplayCell* entstanden sind:

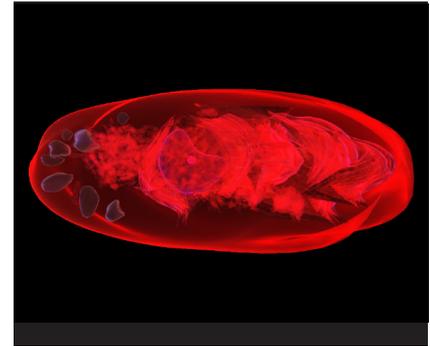


Abb. 2.5
Die Zelle in ihrer Gesamtheit

Abb. 2.6
Der Nucleolus im Nucleus umgeben vom Endoplasmatischen Reticulum

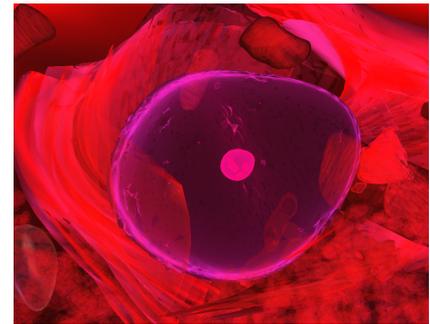
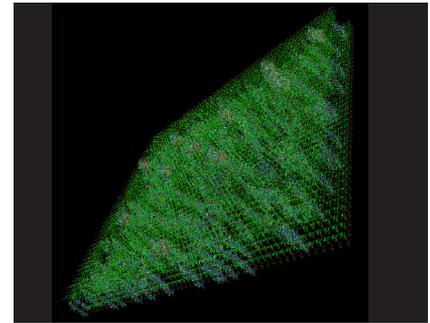


Abb. 2.7
Die Zellmembran aufgebaut aus PDB-Dateien



2.1.3. Projekt CELLmicrocosmos II:

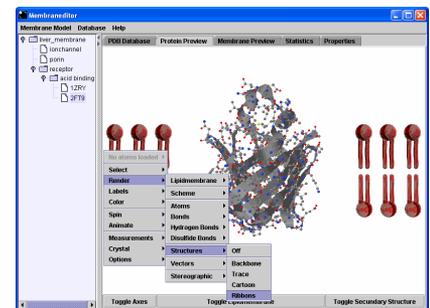
ZellmembranVisualisierung

Im Wintersemester 2005/2006 wurde das Projekt *CELLmicrocosmos II: ZellmembranVisualisierung (Cm2)* angeboten. Insgesamt sechs Teilnehmer erarbeiteten eine erste Version eines Zellmembraneditors (s. Abb. 2.8), der in JAVA™ programmiert wurde und eine zu amira® kompatible Membran im *PDB*-Format erstellt (s. Abb. 2.7). Bei der Protein Database (*PDB*) handelt es sich um eine etablierte, Protein-Modelle frei zur Verfügung stellende Internet-Datenbank (Berman et al. 2000).

Nach leichten Modifizierungen des Membraneditors im Rahmen dieser Masterarbeit konnte mit dessen Hilfe eine *PDB*-Membran erstellt werden. Diese entspricht in ihrer Zusammensetzung in etwa der einer realen Hepatozyten-Plasmamembran.

Ziel war es zunächst, die Membran im Zusammenhang mit den Animationen in amira® zu integrieren und zunächst ein manuelles Anzeigen dieser Membran zu ermöglichen. Somit wurde eine Symbiose aus den beiden Projektseminaren *Cm1* und *Cm2* geschaffen.

Abb. 2.8
Der Zellmembraneditor, welcher die *PDB*-Datei 2FT9 mit ihrer Sekundärstruktur zeigt



2. Interaktive 3D-ZellVisualisierung

2.2. Einarbeitung in die Hepatozytenmembran

2.2.1. Erstellung der Texte

2.2.1.1. Recherche

Die bereits vorhandenen Texte aus *Cm1* bildeten eine gute Basis als Ausgangspunkt für die Recherche der Texte, welche für die einzelnen Komponenten und die Animationen benötigt wurden. Während einige Grundlagen im Internet erarbeitet werden konnten, mussten sie grundsätzlich an Hand einiger Standardwerke der Biochemie überarbeitet und erweitert werden, um den Ansprüchen einer interaktiven Lernumgebung gerecht zu werden.

2.2.1.2. Auswahl der Texte

Auf die Recherche erfolgte die Reduktion auf die relevanten und grundlegenden Informationen. So wurde dabei darauf geachtet, dass die Informationen nicht zu umfangreich wurden und möglichst nur solche Verwendung fanden, die im Rahmen der Arbeit auch nachvollziehbar sein sollten. Ziel war es also, die Aufmerksamkeitsspanne des Benutzers optimal zu nutzen (s. 2.3.5.6.).

2.3. Programmierung und Animation

2.3.1. Das amira®-Package

Bei amira® handelt es sich um ein modulares, objektorientiertes 3D-Visualisierungs- und Modellierungssystem, welches u.a. Echtzeit-Rendering, das zum Visualisierungslabor kompatible Stereoview und Animationen unterstützt. Dabei sind alle Bedienelemente per TCL (s. 2.3.2.1.) und C++ zu großen Teilen scriptfähig und programmierbar.

Einsatzgebiete von amira® sind z.B. die Biologie, Chemie, Medizin, Physik und der Maschinenbau.

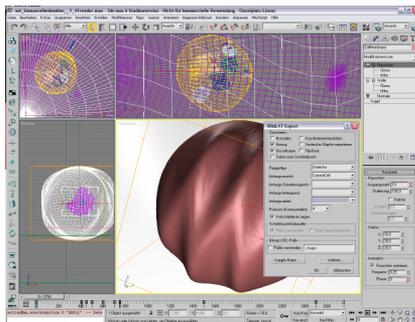


Abb. 2.10

Unter amira® lassen sich die Oberflächen der aus 3ds max® importierten Modelle reduzieren, hier auf ein Viertel der Flächen



Abb. 2.9
VRML97-Export unter 3ds max®: Die modellierten 3D-Modelle werden für die Benutzung unter amira® exportiert

Es basiert auf *Open Inventor from Mercury Release 5.0*, und übernimmt von ihm den 3D-Viewer und den baumstruktur-ähnlichen Aufbau, welcher sich darüberhinaus unter amira® direkt darstellen und manipulieren lässt.

Zum Standard-Paket von amira® wurden im Rahmen dieser Arbeit zwei weitere Pakete genutzt:

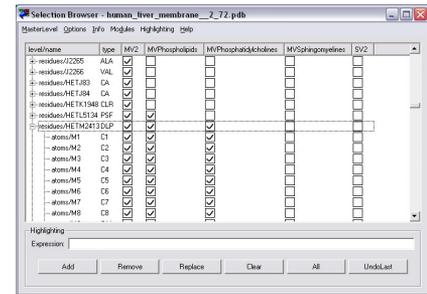
- amiraMol® bietet Visualisierungsmöglichkeiten für Moleküle und molekulare Datensätze (Amira extensions 2002, S. 3).
- amiraDev® ermöglicht es, neue Komponenten und Module zu entwickeln. Hierzu gehören u.a. neue In- und Output-Routinen, Unterklassen und Module für die Visualisierung oder Datensatzanalyse (Amira user's guide 2002, S.445).

Für dieses Projekt weitere wichtige Eigenschaften von amira® und seiner Zusatzpakete sind:

- Der Import von VRML97-Files, mit dessen Hilfe die Modelle aus 3ds max® weiterverarbeitet werden konnten.
- Die Reduzierung von Oberflächen, um die Performance der Darstellung der aus 3ds max® importierten, teilweise sehr detailreichen VRML97-Modelle zu maximieren.

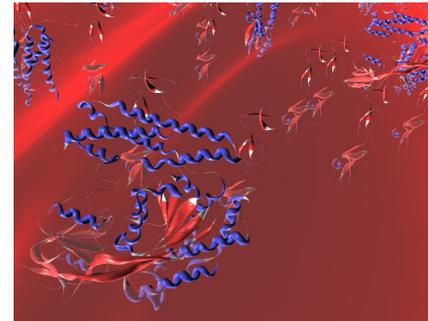
2. Interaktive 3D-ZellVisualisierung

Abb. 2.11
Über den *amira*®
Selection Browser
kann jedes
einzelne Atom
des PDB-Files
ausgewählt
werden



- Die Darstellung von PDB-Files, welche beliebig manipuliert und optimiert werden kann. Darüberhinaus kann jedes Atom eines PDB-Files über den SelectionBrowser (s. Abb. 2.11) angesprochen und die Sekundärstruktur visualisiert werden (s. Abb. 2.12).
- Das Arbeiten mit mehreren Viewern, wodurch die Platzierung von Objekten im drei-dimensionalen Raum deutlich vereinfacht wird (s. Abb. 2.13).

Abb. 2.12
amira® bietet
viele Darstellungsmodi
für Proteine, so
auch die der
Sekundärstruktur



2.3.2. TCL-Scripting

2.3.2.1. Direkter Zugriff auf *amira*®

TCL, die Tool Command Language, bietet eine einfache, C-ähnliche Syntax und ist eine Scriptsprache. Unter *amira*® ist es zum einen möglich, direkt über die *amira console* Befehle einzugeben, oder aber externe Scripte auszuführen. Über diese können die Steuerelemente und Objekte direkt manipuliert werden.

2.3.3. C++ Programmierung

2.3.3.1. Integration der TCL-Scripte

Die eben behandelten TCL-Scripte lassen sich relativ einfach auch mit dem *DisplayCell*-Modul ausführen, wozu sie zunächst mit Hilfe von C++ integriert werden mussten.

2.3.3.2. Integration der Texte

Die Integration der Texte erfolgt audiovisuell. Das heißt, sie werden in Form einer Sounddatei und optional als Textdatei auf dem Monitor wiedergegeben.

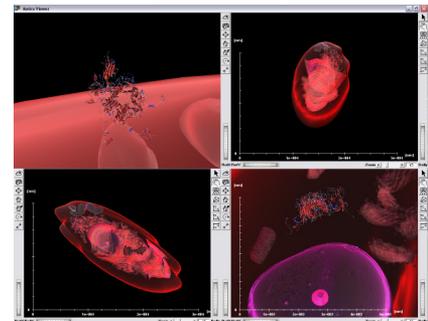
2.3.3.2.1. Textdateien

Die Textdateien werden in einem bestimmten Ordner abgelegt. Der Aufbau der Ordnerstruktur entspricht dabei der Logik, die der C++-Datei zugrunde liegt. Sie ist nach Zellkomponenten aufgeteilt. So gibt es ein Verzeichnis für die Membran, in welchem sich die dem Intro zugehörige Textdatei und sämtliche Dateien befinden, die Bestandteil der Animationen sind. Das Intro bietet dabei eine kurze Einführung zu jeder Komponente (s. 2.3.5.5.2.).

Dabei besteht ein direkter Zusammenhang zwischen der Namensgebung der Demo-Objekte unter *amira*®, der *DisplayCell*-internen Bezeichnungen und den Namen der Verzeichnisse:

Amira-DemoMaker-Objekt:	DemoCellmembranedoublelayer
Verzeichnis der Textdatei:	\$AmiraCellDirectory\sound\Cm_infotext\cellmembrane\doublelayer.info
DisplayCell interner Name:	strAnimationsCellmembrane[1] == „doublelayer“

Abb. 2.13
Die Platzierung
der Modelle wird
in *amira*® durch
die Möglichkeit
erleichtert,
mehrere Viewer
gleichzeitig zu
nutzen



2. Interaktive 3D-ZellVisualisierung

2.3.3.2.2. Sounds

Auch die Ordnerstruktur der Sounddateien entsprechen derselben Logik wie die der Textdateien und werden über ein C++-internen Befehl gestartet.

Verzeichnis der Sounddatei: `$AmiraCellDirectory\sound\Cm_infotext\cellmembrane\doublelayer.WAV`

2.3.3.2.3. Modularer Aufbau

Aufgrund dieses modularen Aufbaues ist es ohne weiteres möglich, ohne eine Neukompilierung der C++-Dateien die Text- und Sounddateien auszutauschen. Änderungen können während amira® ausgeführt wird vorgenommen werden. So können die Textdateien optimal mit Hilfe eines Texteditors auf die Ausgabe am Monitor abgestimmt und die Sounddateien in einem Soundeditor dem zeitlichen Ablauf der Animationen angeglichen werden.

2.3.4. Integration der Membran

2.3.4.1. Membraneditor

Eine weitere wichtige Aufgabe im Rahmen dieser Arbeit war die Integration der *PDB*-Membran in amira®. Dazu mussten zunächst die wichtigsten Bestandteile der Membran und die entsprechenden *PDB*-Dateien recherchiert und anschließend im Rahmen von kurzen Animationen vorgestellt werden.

2.3.4.2. PDB-Recherche

Nachdem die Theorie der Hepatozytenmembran erarbeitet war (s. 2.2.), galt es, die entsprechenden *PDB*-Dateien im Internet zu lokalisieren. Während die Proteine direkt aus der *PDB*-Datenbank unter „<http://www.pdb.org>“ bezogen werden konnten (Berman et al. 2000), erwies sich die Recherche nach den Lipiden als etwas zeitintensiver.

Die meisten Lipide stammen von der Website „<http://xray.bmc.uu.se/hicup>“, deren Entwickler es sich zur Aufgabe gemacht haben, Bestandteile aus *PDB*-Dateien zu extrahieren. So ist z.B. das Lipid Phosphatidylcholin mit dem *PDB*-Komponentennamen „DLP“ in dem *PDB*-File „1LN1“ vorzufinden, ein Phosphatidylcholin-Transferprotein, aus dem lediglich das Lipid zu extrahieren ist (Kleywegt/Jones 1998).

Das Sphingomyelin wiederum war in der Klotho-Datenbank zu finden. Letztere enthält eine Vielzahl biochemischer Verbindungen (Cannon et al. 2006): „<http://www.biocheminfo.org/klotho/>“.

2.3.4.3. Erstellung der Membran

Für die Erstellung einer exemplarischen menschlichen Hepatozyten-Plasmamembran war es zunächst nötig, die Zusammensetzung der Lipide zu recherchieren. Als Grundlage wurden folgende Werte verwendet (Alberts et al. 2002, S. 589):

Tab. 2.1

Lipide = 100% bezieht sich auf die Ausgangswerte aus Alberts et al. 2002

Lipide = 50% bezieht sich auf die Konstellation 50% Lipide zu 50% Proteine, welche in menschlichen Hepatozytenmembranen vorkommt

oben = die zell-äußere Membran

unten = die zell-innere Membran

* s. dazu 3.1.

Lipid	Lipide=100%	Lipide=50%	Position/Anmerkung
Phosphatidylcholin	24%	12%	aktuell 32%*
Sphingomyelin	19%	10%	oben
Cholesterol	17%	8%	4% unten + 4% oben
Phosphatidylethanolamine	7%	4%	unten
Glycolipids	7%	3%	oben
Phosphatidylserine	4%	4%	2% unten + 2% oben
Andere Lipide:	22%	11%	
- Sphingosine	[10%]	[5%]	unten
- Phosphatidylinositol	[12%]	[6%]	unten

2. Interaktive 3D-ZellVisualisierung

Daraufhin mussten Modelle von Proteinen ermittelt werden, die in der Hepatozyten-Membran eines Menschen verankert sein können (s. 2.3.5.2.3.). Dabei wurde sich zunächst aus technischen Gründen auf folgende Proteine beschränkt (s. 3.1.):

PDB-Datei	Lipide=50%	Funktion	Quelle
2AHE.pdb	10%	Ionenkanal	Li/Weinman 2002
1HZ8.pdb	14%	LDL-Rezeptor	Lodish et al. 1996, S. 753
1ZX8.pdb	6%	Transferrinrezeptor	Lodish et al. 1996, S. 757f

Tab. 2.2

*Lipide = 50% bezieht sich auf die Konstellation 50% Lipide zu 50% Proteine, welche in menschlichen Hepatozytenmembranen vorkommt**

Storyboard Membran 3				
take	scene-actions	camera-actions	direct-sound	off-sound
3.1 #	phospholipids 00:18-00:27	nur Phospholipid zeigt für Membranmodell, welches die basis Lipid-schicht zu verfeinern muss	Kamera von 2.2	phospholipids
3.2 #	phospholipids 00:27-00:37	Maßangabe 3 mm einbleiben	Kamera fährt auf Membrananschnitt Halbtotale	phospholipids
3.3 #	phospholipids II 00:39-00:48	Membranmodell transparent auf Halbtotale durch Aufsicht des Lipide 3 mm aus	Kamera Stop	phospholipids
3.4 #	phospholipids IV 00:49-00:50		Kameraabwärtung vertikal um Membranmodell	phospholipids

Abb. 2.14

Das Storyboard zur dritten Szene der Zellmembran-Animation, welche die Phospholipide behandelt

2.3.4.4. Animation

2.3.4.4.1. Erstellung eines Storyboards

Bevor die Animation der einzelnen Membrankomponenten unter amira® vorgenommen werden konnte, galt es, zunächst ein Storyboard zu verfassen (Kahrmann/Reise 1999, S. 74). Auf diese Weise können Abfolge-Fehler vermieden werden, deren Korrektur während des Animations-Scripting unter amira® deutlich mehr Zeit in Anspruch genommen hätte.

2. Interaktive 3D-ZellVisualisierung

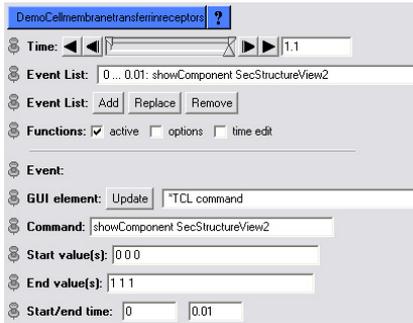


Abb. 2.15

Das amira® DemoMaker-Modul ermöglicht das zeitbezogene Scripting von Animationen

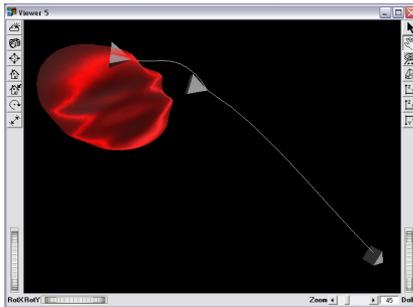


Abb. 2.16

Unter amira® lassen sich die Kamerapfade per Editor animieren

2.3.4.4.2. Animation unter amira®

Unter amira® werden zwei wichtige Objekte zur Verfügung gestellt, mit welchen Animationen erstellt werden können. Zum einen ist das der DemoMaker, welcher scriptfähig ist, in seiner Abspieldauer variierbar und Aktionen ausführen kann. Letztere können wiederum zu einem bestimmten Zeitpunkt oder über einen längeren Zeitraum ausgeführt werden. Es kann sowohl auf die einzelnen Objekte wie die VRML97- und PDB-Modelle direkt zugegriffen werden, als auch indirekt über das Display-Cell-Modul oder das ihm zugrunde liegende TCL-Script.

Das zweite Objekt ist der CameraPath, der direkt mit dem DemoMaker als Zeitgeber verknüpft wird. Ihm können unterschiedliche Kameraperspektiven des aktuellen Viewers unter amira® direkt zugewiesen und anschließend die so entstandenen Kamerapfade verändert werden. In einem separaten Viewer lassen sich die Kamerapositionen verändern und angleichen (s. 3.1.).

2.3.5. DisplayCell

2.3.5.1. Kurzbeschreibung

Das Modul *DisplayCell* ist Bestandteil der *CELLmicrocosmos DisplayCell*-Umgebung und bietet verschiedene Möglichkeiten, die Zelle und ihre Umgebung auf komfortable Weise zu manipulieren. Dabei lässt sich die Zelle auf zwei verschiedenen Granularitätsebenen explorieren: Die erste Ebene wird von den kompletten Zellkomponenten gebildet, welche auf der Grundlage von wissenschaftlichen Daten visualisiert wurden. Die zweite Ebene gibt den Aufbau einer Membranoberfläche exemplarisch wieder und nutzt hierzu *PDB*-Modelle. Der Benutzer dieses Modules kann die Farben und die Transparenz-Einstellungen ändern und unterschiedliche Informationen und Animationen dazu anzeigen lassen.

2.3.5.2. Voraussetzungen

Die folgenden Dateien müssen bereits im amira®-System installiert sein, um das Modul nutzen zu können:

- amira® 3.1 incl. amiraMol®, amiraDev®,
- die aktuelle Version der *CELLmicrocosmos DisplayCell*-Umgebung (z.B. „Cm_DisplayCell__3_0“) incl. einer hx-Datei und aller dazugehörigen *WRL*-, *PDB*-, *Amira-Surface* und *Amira-CameraPath*-Dateien, welche sich im zur hx-Datei gehörigen Verzeichnis befinden,
- das aktuelle TCL-Script (z.B. „DisplayCell__1_2.tcl“),
- das aktuelle sound-Verzeichnis, in dem sich die auditiven und textuellen Informationen für die Zellkomponenten befinden, incl. WAV- und INFO-Dateien,
- eine kompilierte Version von *DisplayCell* („MyDisplayCell.h“, „MyDisplayCell.cpp“, etc.), die sich im AmiraLocal-Verzeichnis befinden muss (zur Kompilierung wird eine Entwicklungsumgebung wie z.B. Microsoft® Visual C++® 6.0 genutzt),
- die korrekt Definition der Pfade in der C++- und der TCL-Datei. Dabei sollte es sich bereits um die endgültige Position der Dateien handeln, da im nächsten Schritt bereits amira® den ersten Dateipfad zugewiesen bekommt.

Nun muss in amira® der „Development Wizard“ aufgerufen werden. Hier wird das AmiraLocal-Verzeichnis festgelegt. Anschließend wird amira® neu gestartet und erneut der „Development Wizard“ aufgerufen. Diesmal wird „Create build system“ angewählt, „single package“ und „mypackage“ ausgewählt und mit „OK“ bestätigt.

Unter dem Arbeitsbereich im Visual C++® sind die Dateien „So.h“ und „So.cpp“ zu entfernen, da sie nicht benötigt werden und Fehler erzeugen.

Bevor das Projekt kompiliert werden kann, muss die Datei „winmm.lib“ hinzugefügt werden. Sie ist wichtig für das Abspielen von Sounds und wird i.d.R. bei der Standard-Installation vom Visual C++® mitinstalliert. Dazu wird unter dem Arbeitsbereich in Visual C++® auf den Namen „mypackage“ mit der rechten Maustaste geklickt und „Dateien dem Projekt hinzufügen ...“ ausgewählt. Die Datei befindet sich i.d.R. Im Verzeichnis „C:/Programme/Microsoft Visual Studio/VC98/Lib/Winmm.lib“.

Dabei ist darauf zu achten, dass die ausgegebene Datei „mypackage.dll“ unter folgenden Verzeichnis gespeichert wird: „../bin/arch-Win32-Optimize/“. Dazu müssen evtl. die Projekteinstellungen geändert werden oder unter „Erstellen/Aktive Konfiguration festlegen ...“ „mypackage – Win32 Release“ ausgewählt werden.

Nun muss das Projekt einmal komplett kompiliert werden über „Erstellen -> Alles neu erstellen“.

2. Interaktive 3D-ZellVisualisierung

Abschließend muss lediglich die aktuelle hx-Datei gestartet werden. Um die Grundfarbe zu ändern, kann unter amira® über „Edit-> Database-> Materials -> Interior“ der Eintrag geändert werden. Die Änderung wird allerdings erst nach einem „Update Display“ von *DisplayCell* übernommen.

2.3.5.4. Erzeugung

Um ein *DisplayCell*-Objekt zu erzeugen, muss zunächst ein beliebiges *Surface* unter amira® erstellt werden. So ist z.B. die Importierung eines *WRL*-Objekts möglich, welches dann mit einem *IvToSurface*-Objekt verbunden und um den Vorgang abzuschließen der „Dolt“-button ausgewählt wird. Anschließend muss das neu erzeugte *Surface*-Objekt ausgewählt und über das Menü „Local/DisplayCell“, das nach einem Rechts-Klick auf das Objekt erscheint, angewählt werden. Die „data“-Verbindung, die zwischen dem *Surface*- und dem *DisplayCell*-Objekt erstellt wird und normalerweise eine Beziehung zwischen zwei Objekten darstellt, hat keine Funktion.



Es ist möglich, mehr als ein *DisplayCell*-Objekt zu generieren. Dabei zeigt der orangefarbene Sichtbarkeits-Indikator das aktuell aktivierte *DisplayCell*-Objekt an. Um ein anderes, bereits erzeugtes *DisplayCell*-Objekt anzuzeigen, ohne die Einstellungen zu verändern, muss es nur angeklickt und in der *DisplayCell*-Oberfläche „Update View“ angewählt werden.

Anmerkung: Insbesondere, wenn mehr als ein *DisplayCell*-Objekt verwendet werden, ist es wichtig, dass alle Labels der Objekte mit „*DisplayCell*“ beginnen. Wenn also der Name durch das TCL-Kommando „setLabel“ in der *amira* console geändert werden muss, sollte das auf folgende Weise geschehen:

```
DisplayCell setLabel DisplayCellNewName or  
DisplayCell setLabel DisplayCell_NewName or  
DisplayCell setLabel DisplayCell-NewName .
```

2.3.5.5. Bedienung

2.3.5.5.1. Die Zelle

Globale Zell-Parameter

Die Schaltfläche der „Global Cell Parameters“ bietet direkten Zugriff auf das Erscheinungsbild der gesamten Zelle und ihrer einzelnen Komponenten:



„Update Display“ aktualisiert die Einstellungen aller Komponenten mit denen von *DisplayCell*. Diese Schaltfläche ist besonders nützlich, wenn mehrere *DisplayCell*-Objekte initialisiert sind. „Show All Comp.“ zeigt die gesamte Zelle und ihre Komponenten, während „Hide All Comp.“ sie entsprechend ausblendet.

Zell-Animationen



„Cell animations“ bietet Zugriff auf die Animationen, welche die gesamte Zelle behandeln, wobei

- „STOP“ alle Animationen stoppt (auch die aller Komponenten),
- „animationname“ startet die Animation mit dem ausgewählten Thema,
- „ALL“ spielt alle Animationen iterativ ab.

Die „info“-Checkbox reguliert die Sichtbarkeit des Textes, der die auditiven Informationen der aktuellen Animation, optional einige zusätzliche Kommentare und die Quellenangaben wiedergibt. Dabei handelt es sich um eine globale Einstellung, das bedeutet, es reguliert den Status des Textfeldes für alle Animationen.

Die „scale“-Checkbox de- oder aktiviert die nm-Skala.

Die „light“-Checkbox schaltet die Lichtquellen ein und aus.

2.3.5.5.2. Die Zellkomponenten

Farben, Sichtbarkeit und Intro



Die zwei Schalter können benutzt werden, um den „specular“ und den „emissive color“ der Komponente zu ändern.

Die „visible“-Checkbox wird benutzt, um eine Komponente ein- oder auszublenden.

Die „intro“-Checkbox hebt eine Komponente hervor und kommentiert sie.

Diese Einstellungsmöglichkeit für die Komponente „Cellmembrane“ gibt es unter dem entsprechenden Namen auch für die folgenden Zellkomponenten:

„Agranularer“, „Granularer“, „Mitochondrium“, „Golgi apparatus“, „Vesicle“, „Nucleus“, „Nucleolus“, „Cellenv“.

Transparenz



Der „Memtrans“-Slider kann benutzt werden, um die Transparenz-Einstellungen einer Komponente zu verändern. Bei 1 ist sie komplett durchsichtig und somit unsichtbar, bei 0 erscheint sie völlig massiv.

Die Einstellung „Memtrans“ für die Plasmamembran existiert ebenfalls unter den entsprechenden Namen für die übrigen Zellkomponenten:

„Aertrans“, „Gertrans“, „Mitotrans“, „Vesicletrans“, „Nucleustrans“, „Nucleolustrans“, „Cellenvtrans“.

Animationen



„Cellmembrane animations“ bietet Zugriff auf alle Animationen, die die Plasmamembran betreffen, wobei

- „STOP“ alle Animationen stoppt (auch die der gesamten Zelle),
- „animationname“ startet die Animation mit dem ausgewählten Thema,
- „ALL“ spielt alle Animationen iterativ ab.

Diese Einstellungsmöglichkeit existiert bisher nur für die Plasmamembran und die Zelle.

Abb. 2.17 - 2.21: Alle Abbildungen zeigen Screenshots des DisplayCell-Moduls, die direkt unter amira® erzeugt werden können (s. 2.3.5.7.)

2. Interaktive 3D-ZellVisualisierung

2.3.5.6. Animationen und Texte

Bei der Verknüpfung der Texte mit den Animationen wurden zwei unterschiedliche Arten der Wiedergabe genutzt: Zum einen das „Intro“, welches die Zellkomponenten im einzelnen kurz vorstellt. Hierbei muss der Benutzer die „Intro“-Checkbox neben der entsprechenden *DisplayCell*-Komponente anwählen, zu der dann ein Informationstext wiedergegeben und das entsprechende drei-dimensionale Modell hervorgehoben wird. So lange die Checkbox aktiviert ist, kann der Benutzer durch die Zelle navigieren, um die Komponente im hervorgehobenen Zustand zu explorieren.

Zum anderen sind das die Animationen. Diese werden neben der entsprechenden Komponente im Animationsmenü angewählt und entweder in ihrer Gesamtheit iterativ oder einzeln bei Direktauswahl abgespielt (s. 2.3.5.5.2.).

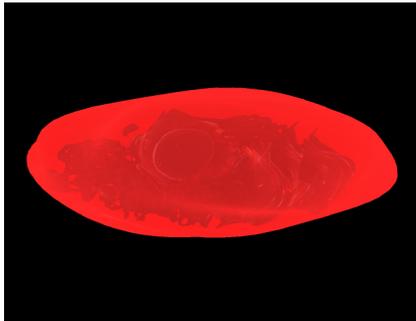


Abb. 2.22
Plasma-
membran

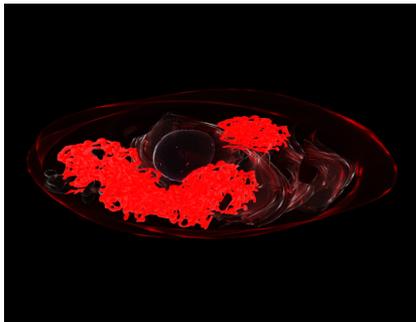


Abb. 2.23
Agranuläres
Endoplas-
matische
Reticulum

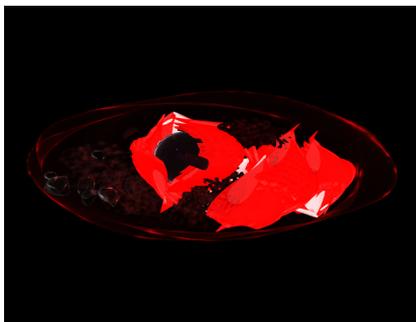


Abb. 2.24
Granuläres
Endoplas-
matisches
Reticulum

2.3.5.6.1. Intro: Zellkomponenten

Es folgen nun die Intro-Texte im einzelnen, die als Bestandteil der *DisplayCell*-Umgebung verwendet werden:

Plasmamembran

Die Plasmamembran ist die äußere Hülle, die das Cytoplasma einer lebenden Zelle umgibt (Alberts 2002, S. G:27; Nelson/Cox 2001, S. 1313). Sie besteht aus einer Lipiddoppelschicht, welche zusätzlich mit Proteinen versehen ist (Cooper 1997, S. 651).

Agranuläres Endoplasmatisches Reticulum

Beim Agranulären Endoplasmatischem Reticulum handelt es sich um ein System aus Doppelmembranen und sekretorischen Gängen (Nelson/Cox 2001, S. 1304). Hier findet der größte Teil der zellulären Fettsäuresynthese statt (Cooper 1997, S. 654). Die synthetisierten Stoffe werden zum Golgi-Apparat weitergeleitet (Nelson/Cox 2001, S. 36).

Granuläres Endoplasmatisches Reticulum

Beim Granulären Endoplasmatischem Reticulum handelt es sich um ein System aus Doppelmembranen und sekretorischen Gängen (Nelson/Cox 2001, S. 1304). Seine Oberfläche ist mit einer großen Anzahl von Ribosomen ausgestattet (Cooper 1997, S. 653). Hier wird Biosynthese von sekretorischen und membrangebundenen Proteinen betrieben (Alberts 2002, S. G:31). Die synthetisierten Stoffe werden zum Golgi-Apparat und zum Lumen weitergeleitet. Letzteres ist ein vom Cytosol abgetrennter Bereich, welcher u.a. Proteine enthält (Nelson/Cox 2001, S. 36).

2. Interaktive 3D-ZellVisualisierung

Mitochondrium

Mitochondrien sind die Energielieferanten der Zelle. Dabei handelt es sich um von einer Membran umgebene Organellen innerhalb der Cytoplasmas, welche Enzymsysteme enthalten. Letztere sind wichtig für den *Citratzyklus*, die Fettsäureoxidation, den Elektronentransport und die *Oxidative Phosphorylierung* (Nelson/Cox 2001, S. 1311).

Golgi-Apparat

Der Golgi-Apparat modifiziert und sortiert Proteine und Lipide, welche er vom Endoplasmatischem Reticulum erhalten hat (Alberts 2002, S. G:16). Die Endprodukte werden dann mittels eines Schlüssels innerhalb oder außerhalb der Zelle verschickt. Beim Golgi-Apparat unterscheidet man die cis-Seite, welche dem Zellkern und dem Endoplasmatischem Reticulum zugewandt ist, und die trans-Seite, welche in Richtung des Cytoplasmas weist (Nelson/Cox 2001, S. 36).

Vesikel

Zu den Vesikeln zählen u.a. die Lysosomen und die Peroxisomen:

Die Lysosomen befinden sich im Cytoplasma und sind membranumhüllte Organellen, die zahlreiche hydrolytische Enzyme enthalten. Diese wiederum dienen als Abbau- und Recyclingzentren für nicht benötigte Bestandteile (Nelson/Cox 2001, S. 1310), welche z.B. der Golgi-Apparat hervorgebracht hat (Nelson/Cox 2001, S. 36).

Das ebenfalls von einer Membran umgebene Peroxisom liegt im Cytoplasma und enthält Peroxid bildende und zerstörende Enzyme (Nelson/Cox 2001, S. 1313). Somit finden auch hier oxidative Reaktionen statt (Cooper 1997, S. 651).

Auch das Endoplasmatische Reticulum und der Golgi-Apparat können den Vesikeln zugeordnet werden (Lodish et al. 1996, S. 1392).

Nucleus

Der Zellkern bzw. der Nucleus enthält die in Chromosomen organisierte DNA und ist von einer Doppelmembran umgeben (Alberts 2002, S. G:25; Nelson/Cox 2001, S. 1312). Diese steht in enger Verbindung mit dem Endoplasmatischen Reticulum und ist mit Membranöffnungen, den sog. Kernporen versehen, durch welche Proteine und Enzyme in das Innere des Zellkerns gelangen können (Nelson/Cox 2001, S. 37).

Abb. 2.25
Granuläres
Endoplas-
matisches
Reticulum

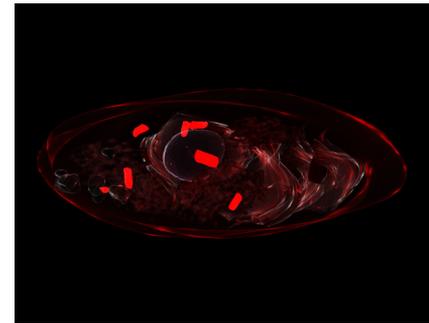


Abb. 2.26
Golgi-
Apparat

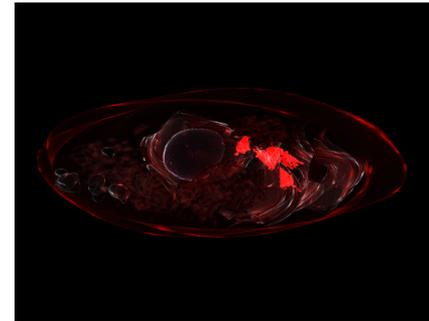


Abb. 2.27
Vesikel

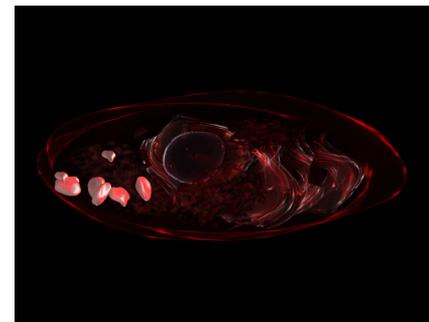
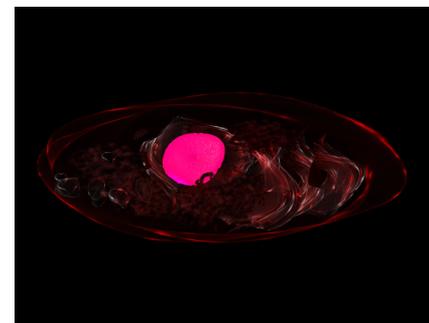


Abb. 2.28
Nucleus



2. Interaktive 3D-ZellVisualisierung

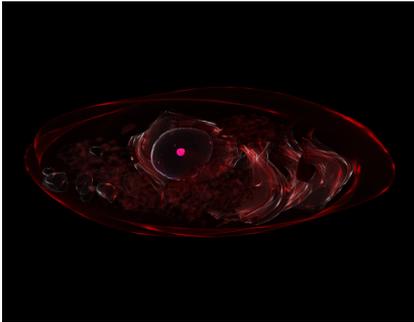


Abb. 2.29
Nucleolus



Abb. 2.30
Zellumgebung

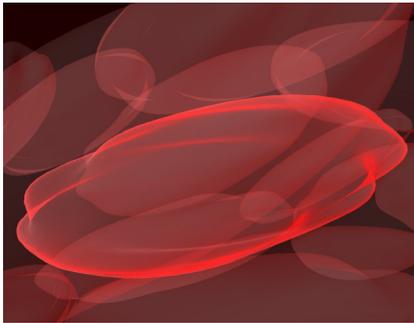
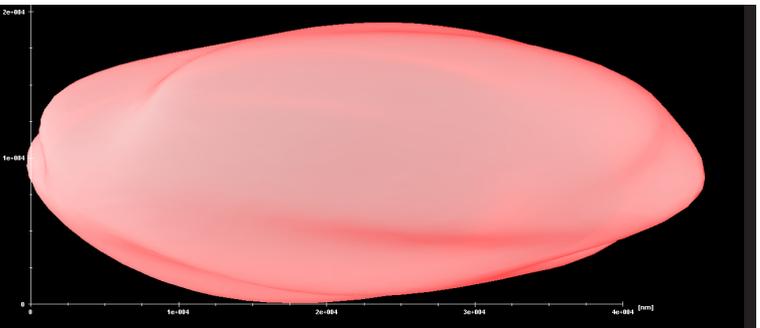


Abb. 2.31
Zellanimation:
1. Hepatozyten

Abb. 2.32
Zellanimation:
2. Form des
Hepato-
zyten



Nucleolus

Beim Nucleolus handelt es sich um einen sehr dunklen, membranlosen Bereich innerhalb des Nucleus (Lodish et al. 1996, S. 1388), welcher an der Synthese und Transkription der ribosomalen RNA und der Bildung der Ribosomen beteiligt ist (Cooper 1997, S. 650). Dabei gehen die Ribosomenuntereinheiten aus einer Kombination der eben genannten rRNA und ebensolcher Proteinen hervor (Lodish et al. 1996, S. 1388).

Zellumgebung

Jeder Hepatozyt ist durch den sog. Junctional Complex fest mit seinen umliegenden Nachbar-Zellen verbunden. Zusammen machen sie etwa 60% aller Zellen in der Leber aus und aufgrund ihrer Größe 80% des Gesamtvolumens. Die Zellen bilden ein weit verzweigtes, perforiertes, miteinander verbundenes Netzwerk, in dessen Zwischenräumen sich die Sinusendothelzellen befinden. Diese gehören nicht dem Junctional Complex an und machen zusammen etwa 40% der Zellen-Masse aus; allerdings nur 10% des Gesamtvolumens. Der Rest des Lebervolumens wird vom extrazellulären Raum ausgefüllt (Berry, Edwards und Barritt, S. 2f).

2.3.5.6.2. Animation: Hepatozyt

Um den Hepatozyten in seiner Gesamtheit vorzustellen, wurden folgende zwei kurze Animationen, welche über das „Cell animation“-Menü anwählbar sind, integriert.

1. Die Leberzelle besteht zum größten Teil aus Hepatozyten: Diese repräsentieren ca. 60% aller Leberzellen und machen 80% des Gesamtvolumens aus (Berry et al. 1991, S. 2f).
2. Die Form der Hepatozyten ist nahezu kubisch und weist eine Kantenlänge von 15.000 nm auf (Lodish et al. 1996).

2. Interaktive 3D-ZellVisualisierung

2.3.5.6.3. Animation: Plasmamembran

Die nun folgenden Texte sind die Resultate aus der wissenschaftlichen Recherche zur Thematik des Aufbaues der menschlichen Hepatozytenmembran (s. 2.2.1.1.).

Ziel war es, einen prägnanten thematischen Überblick zu bieten, der aber nicht das Thema erschöpfend behandeln sollte (s. 2.2.1.2.).

1. Die Plasmamembran des Hepatozyten besteht, wie alle anderen Membranen ebenfalls, aus der ihr formgebenden Doppellipidschicht (Lodish et al. 1996, S. 621).
2. Die Lipide unterteilen sich in drei Gruppen: Die der Phospholipide, der Glycolipide und der Cholesterole (Cooper 1997, S. 468f).
3. Die erste Lipid-Gruppe sind die Phospholipide, welche grundsätzlich amphiphil sind. Das heißt, sie haben eine hydrophobe Seite, die der Membranmitte zugewandt ist und deren Wasserstoffketten der Fettsäurereste zusammen eine ca. 3 nm dicke Schicht bilden. Die hydrophile Gegenseite ist mit einer polaren Kopfgruppe versehen, die zum einen der extrazellulären Matrix und zum anderen dem intrazellulären Zytoplasma zugewandt ist. Nominal erkennt man die dieser Gruppe zugehörigen Lipide i.d.R. an dem Zusatz Phosphatidyl. Sie weisen alle eine ähnliche Molekülgestalt und meist eine neutrale Ladung auf (Lodish et al. 1996, S. 622ff).
4. Die Phospholipide der Hepatozyten gliedern sich ungefähr folgendermaßen auf: 24% der Gesamtanzahl entfallen auf die Phosphatidylcholine und 19% auf die Sphingomyeline; beide befinden sich eher in der äußeren Membran. In der inneren Membran befinden sich eher die Phosphatidylethanolamine, welche 7% der Lipide ausmachen und die mit 4% zu veranschlagenden Phosphatidylserine. Letztere sind zusätzlich die einzigen aus dieser Aufzählung, die eine negative Ladung aufweisen. Die übrigen 22% werden von anderen, selteneren Phospholipiden besetzt (Lodish et al. 1996, S. 624; Alberts 2002, S. 589).

Abb. 2.33
Zellmembran-
animation:
1. Doppellipid-
schicht



Abb. 2.34
Zellmembran-
animation:
2. Lipidgruppen

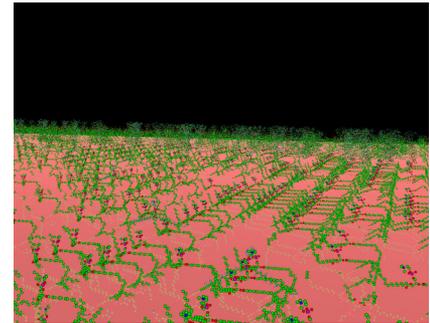


Abb. 2.35
Zellmembran-
animation:
3. Phospholipide

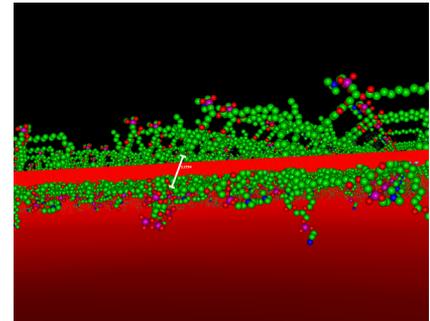
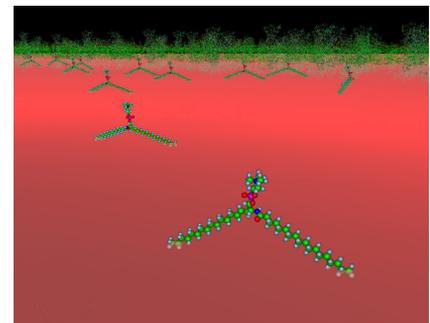


Abb. 2.36
Zellmembran-
animation:
4. Phosphatidyle



2. Interaktive 3D-ZellVisualisierung

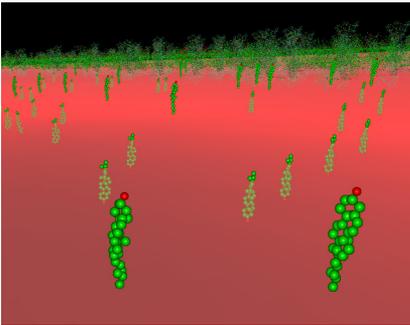


Abb. 2.37
Zellmembran-
animation
5. Cholesterole

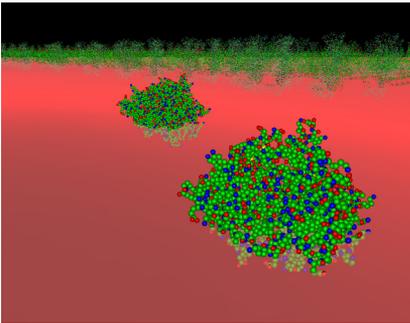


Abb. 2.38
Zellmembran-
animation
6. Glycolipide

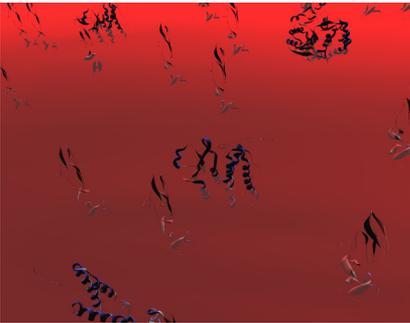


Abb. 2.39
Zellmembran-
animation
7. Membran-
proteine

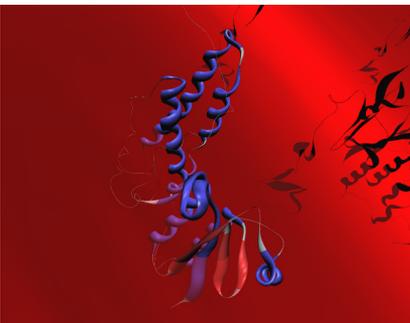


Abb. 2.40
Zellmembran-
animation
8. Ionenkanäle

5. Die zweite Lipid-Gruppe wird von den Glycolipiden gebildet, welche 7% der Membran besetzen und jeweils einen Kohlenhydratrest aufweisen (Alberts 2002, S. 589; Berman et al. 2000; Nelson/Cox 2001, S. 1306). Dieser ist besonders wichtig für die Interaktion der Zelle mit ihrer Umgebung und für die Zellstabilität (Alberts 2002, S. 592).

Die Komponente ist extrahiert aus 1KNN.pdb (Titel: Hydrolyase: streptomyces lividans xylan binding domain CBM13 in complex with lactose; streptomyces lividans) (Berman et al. 2000).

6. Die dritte Lipid-Gruppe ist das zu den Sterinen zählende Cholesterol, welches 17% der Lipide ausmacht (Alberts 2002, S. 589; Kleywegt/Jones 1998). Es wird teilweise durch die LDL-Rezeptoren in der Zellmembran als Fettsäuren für die Phospholipidproduktion eingelagert (Cooper 1997, S. 309f). Es steigert die Durchlässigkeit der Membran, tendiert allerdings auch dazu, die Fluidität der Membran einzuschränken (Alberts 2002, S. 588).

Die Komponente ist extrahiert aus 1N83.pdb (Titel: Lipid binding protein: crystal structure of the complex between the orphan nuclear 2 hormone receptor ROR(Alpha)-LBD and cholesterol; homo sapiens) (Berman et al. 2000).

7. Zusätzlich enthält die Membran sehr viele unterschiedliche Proteine, deren Eigenschaften einen bedeutenden Teil der Charakteristik einer Zelle ausmachen (Lodish et al. 1996, S. 621). Das gewichtsbezogene Verhältnis von Lipiden zu Proteinen ist i.d.R. ausgewogen (Cooper 1997, S. 469).

8. Zu den zwischen den Phospholipiden verankerten Proteinen zählen die Ionenkanäle, welche hier ca. 10% der Gesamtmembranoberfläche ausmachen. Bei dem hier gezeigten Protein handelt es sich um einen interzellulären Chloridionenkanal (Li/Weinman 2002; Berman et al. 2000).

Bei dem Modell handelt es sich um 2AHE.pdb (Titel: Metal transport: crystal structure of a soluble form of CLIC4. Inter-cellular chloride ion channel; homo sapiens) (Berman et al. 2000).

2. Interaktive 3D-ZellVisualisierung

9. Die LDL-Rezeptoren machen hier 14% der Membranoberfläche aus. Dabei handelt es sich um Low Density Lipoproteinrezeptoren (Berman et al. 2000), welche zu den Glycoproteinen zählen und Cholesterin ins Zellinnere leiten: Dort wird es von den Hepatozyten für die Biosynthese der Gallensäure verwendet (Lodish et al. 1996, S. 753).

Bei dem Modell handelt es sich um 1HZ8.pdb (Titel: Lipid binding protein: solution structure and backbone dynamics of a concatemer of EGF-homology modules of the human low density lipoprotein receptor; homo sapiens) (Berman et al. 2000).

10. Transferrin-Rezeptoren, hier ca. 6% der Membranoberfläche ausmachend, versorgen Zellen mit Transferrin, ein Glycoprotein, welches allen Zellen im Gewebe aus der Leber Eisen zur Verfügung stellt (Lodish et al. 1996, S. 757f).

Bei dem Modell handelt es sich um 1CX8.pdb (Titel: Metal transport: crystal structure of the ectodomain of human transferrin receptor; homo sapiens) (Berman et al. 2000).

11. Bei der Membran handelt es sich nicht um eine fixierte Oberfläche, sondern um eine kontinuierlich sich verändernde, temperaturabhängige fluide Ansammlung von Stoffen. So können sich die Glyco- und Phospholipide um ihre Längsachse drehen und sich innerhalb einer Membranschicht bewegen. Ein Lipidmolekül kann seine Position 107 mal pro Sekunde mit seinem Nachbarlipid tauschen und in manchen Membranen sogar durch bestimmte Transmembranproteine die Seite wechseln (Lodish et al. 1996, S. 625).

Abb. 2.41
Zellmembran-
animation:
9. LDL-
Rezeptoren



Abb. 2.42
Zellmembran-
animation:
10. Transferrin-
Rezeptoren

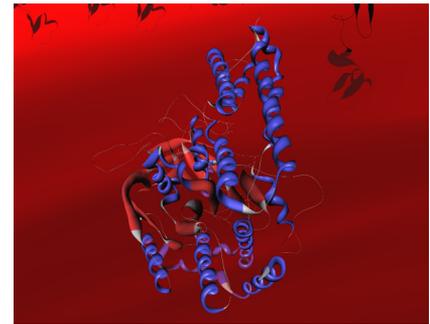
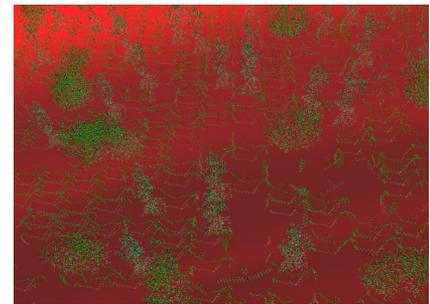


Abb. 2.43
Zellmembran-
animation:
11. Lipid-
bewegung



2.3.5.7. Dokumentierung unter amira®

Teile dieser Dokumentation (s. 2.3.5.5.1., 2.3.5.5.2.) sind bereits in amira® integriert. Hierzu ermöglicht amira® es, zunächst ein LaTeX-kompatibles Grundgerüst einer Dokumentation direkt aus dem *DisplayCell*-Modul zu exportieren. Dabei werden die Grundstruktur des Moduls berücksichtigt und automatisch sämtliche Screenshots der unterschiedlichen Schalter erstellt. Diese können dann mit einem simplen Texteditor überarbeitet, als HTML exportiert und abschließend im Dokumentations-Verzeichnis von amira® abgelegt werden. Zu diesem Zweck müssen die HTML- und PNG-Dateien, welche sich unter „\$AmiraLocalDirectory/src/mypackage/doc/mypackage“ befinden, in das amira-Verzeichnis, i.d.R. „C:/Programme/Amira-3.1/share/doc/usersguide“, kopiert werden.

Durch eine Suche nach *DisplayCell* im User's Guide von amira® wird die auf diese Weise integrierte Seite automatisch angezeigt. Ebenso können für die Suche Begriffe aus der Dokumentation verwendet werden, z.B. „Global Cell Parameters“. Alternativ ist es auch möglich, direkt den Hilfe-Button „?“ von *DisplayCell* anzuwählen.

2. Interaktive 3D-ZellVisualisierung

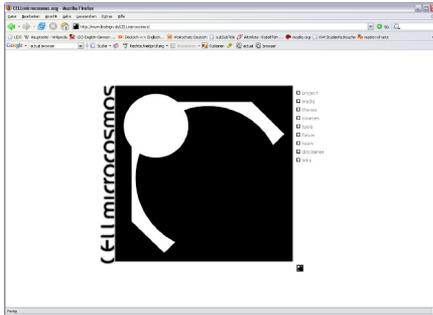


Abb. 2.44

Die
CELL-
microcosmos-
Website im
Mozilla
Firefox™
Browser

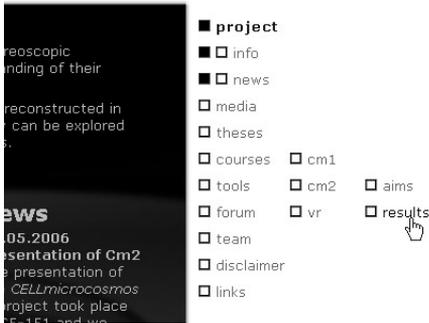


Abb. 2.45

Die Navigation
erweitert sich,
wenn der
Mauszeiger
darüber
bewegt
wird

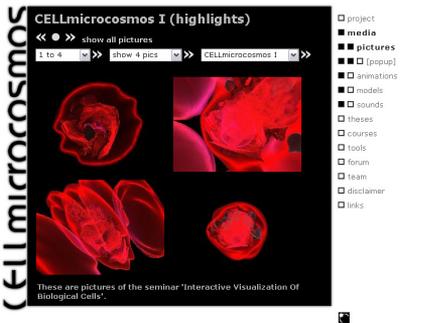


Abb. 2.46

Die Pictures-
Seite in die Web-
site integriert

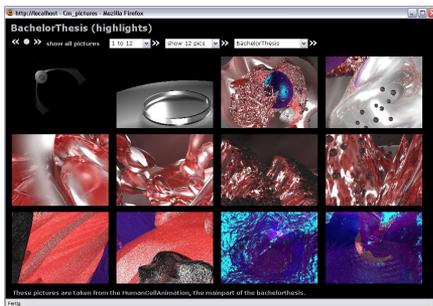


Abb. 2.47

Die Pictures-
Seite in einem
Popup-Fenster

2.4. Projekt-Dokumentation und Internetpräsenz

2.4.1. Inhalte

Ein Bestandteil dieser Masterarbeit war die Dokumentierung der Ergebnisse des CELLmicrocosmos-Projektes. Ziel hierbei war es zum einen, Interessenten einen prägnanten Überblick über die bisherigen Ergebnisse zu bieten und zum anderen, eine Plattform in Form eines Forums zu schaffen, welche für zukünftige Projekte genutzt werden kann. Da die Internetpräsenz auf eine internationale Zielgruppe abzielt, wurde sie sogleich komplett in englischer Sprache verfasst, wobei einige Inhalte wie Präsentationen nach wie vor in Deutsch vorzufinden sind.

Dabei sollte die Präsentation visuell anspruchsvoll aber nicht zu überfrachtet sein, weshalb auf aufwendige, z.B. mit Macromedia® Flash™ realisierbare Effekte, verzichtet wurde.

Als URL wurde ebenfalls eine international ausgerichtete Adresse gewählt, die bewusst nicht ausschließlich kommerziell („.com“) oder auf die Lehre („.edu“) beschränkt sein sollte: „http://www.CELLmicrocosmos.org“.

2.4.2. PHP-Programmierung

Eine zentrale PHP-Datei, die „index.php“, reguliert das Erscheinungsbild der Website. Sie enthält einen IFrame (einen internen Frame), in dem die Inhalte dynamisch aktualisiert werden können. Diese werden aus externen PHP-Dateien bezogen.

Als programmiertechnische Besonderheit sind hierbei das Menü und das „Pictures“-Script hervorzuheben. Bei dem Menü handelt es sich um ein Popup-Menü (s. Abb. 2.45), bei dem die unterschiedlichen Level der Navigation durch einen MouseOver-Effekt eingblendet werden können. So sind alle Teilbereiche der Webpage von jeder Seite aus zu erreichen. Realisiert wurde dies durch eine Kombination von PHP, JavaScript und StyleSheets (CSS).

Aus der gleichen Kombination ging das „Pictures“-Script hervor (s. Abb. 2.46), das es auf relativ einfache Art und Weise ermöglicht, mehrere Unterordner für Bilder anzulegen, über eine Konfigurationsdatei zu integrieren, Beschreibungen zu dem jeweiligen Ordner, die Lage der zu verwendenden CSS-Datei und die Bilder-Darstellungs-Parameter festzulegen.

2. Interaktive 3D-ZellVisualisierung

So ist es auch möglich, diese Seite als Popup-Fenster mit einem deutlich größeren Anzeigebereich zu öffnen (s. Abb. 2.47). Statt insgesamt vier Bilder darzustellen, werden hier zwölf gleichzeitig dargestellt. Bei der Variante mit vier Bildern wird bei einem Klick auf ein Bild zunächst eine frame-angepasste Variante des Bildes angezeigt. Wird jetzt erneut darauf geklickt, wird es in Originalgröße in einem Popup geöffnet. Bei der Variante mit zwölf Bildern wird das Bild bereits beim ersten Klick in Originalgröße angezeigt.

Um die VRML-Modelle aus der Bachelorarbeit (Cm) anzeigen zu können, wird der Cortona®-Viewer empfohlen, der sich in das Layout der Webpage hervorragend einfügt.

Als Minimal-Auflösung wird von der Webpage 1024X768 Pixel vorausgesetzt. Zur optimalen Nutzung ist allerdings eine Auflösung von 1280X960 Pixel (s. Abb. 2.44) anzuraten.

Abb. 2.48
Die VRML97-
Version der Cm-
Gesamtzelle im
Cortona®-Viewer

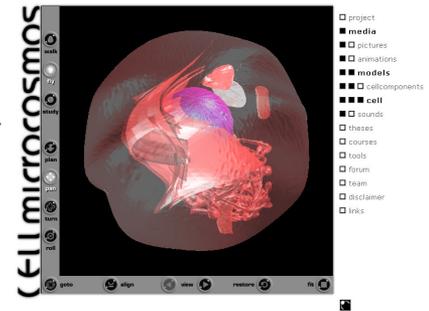
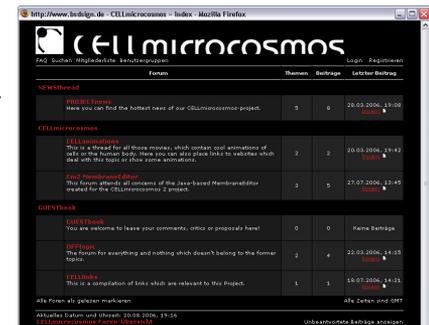


Abb. 2.49
Das Forum in
einem Popup-
Fenster



2.4.3. Validierung

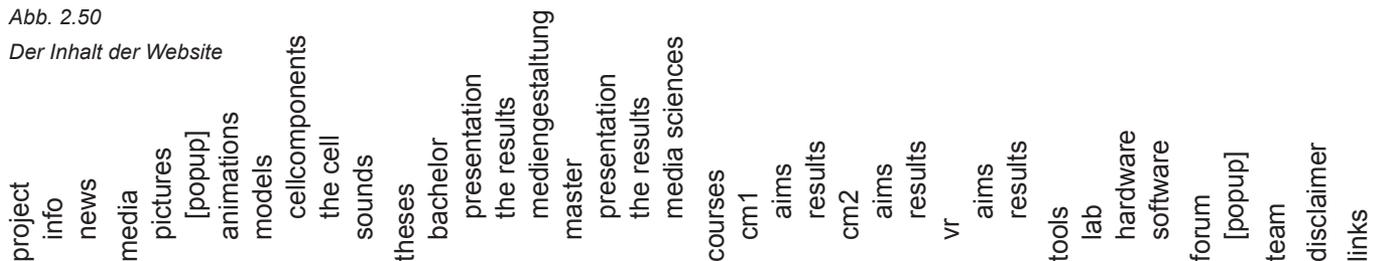
Um maximale Kompatibilität zu allen aktuellen Browsern bieten zu können, wurde die Webpage darüberhinaus W3C-validiert. Das heißt, es wurde eine Überprüfung der Style-Sheets daraufhin durchgeführt, ob sie die Vorgaben des W3C für das CSS-Format 1.0 einhalten und die Struktur der aus dem PHP-Code erzeugten HTML-Dateien „XHTML 1.0 Transitional“ konform sind. Dabei musste „Transitional“ gewählt werden, da die Inhalte in einem IFrame dargestellt werden, der nicht kompatibel zum „XHTML 1.0 Strict Format“ ist. Besonders für die Einbettung des Cortona® Viewer und die des Forums ist die IFrame-Lösung aber besonders praktikabel (W3C 2006).

2.4.4. Forum

Der interaktive Teil der Seite wird durch das „phpBB“-Forum gebildet, das zum einen aus einem Mitgliederbereich besteht, der nur für Projektteilnehmer zugänglich ist, und zum anderen aus einem öffentlichen Bereich, in dem Meinungen und Kritik auch von externen Interessierten geäußert werden können. Bei „phpBB“ handelt es sich um ein PHP-basiertes OpenSource Bulletin Board Package, welches im Rahmen dieser Arbeit an eine MySQL-Datenbank angebunden und dann in die CELLmicrocosmos-Website integriert wurde (phpBB Group 2006). Wie die Pictures-Seite ist das Forum im IFrame direkt als auch als Popup aufrufbar (s. Abb. 2.49).

Abb. 2.50

Der Inhalt der Website



3. Resultate

3.1. Neue Problemfelder und Lösungsansätze

Bei dem Einsatz der für diese Masterarbeit relevanten Software haben sich einige Schwächen gezeigt, die im folgenden kurz angesprochen und erste Lösungsansätze erörtert werden sollen.

3.1.1. Der Membraneditor

Die derzeitige Version des Membraneditors weist noch viele Weiterentwicklungsmöglichkeiten auf. So basiert er bisher nur auf einem einzigen Lipidtyp und die Proteinverteilung erfolgt noch völlig willkürlich nach einem Zufallsalgorithmus. Die Darstellung ist somit noch sehr weit von der realen Beschaffenheit einer Membran entfernt, bildet aber eine hervorragend Grundlage für eine Weiterentwicklung.

Zur Zeit benutzt die erste Version des Membraneditors ein einziges Lipid als Grundlage für die Platzierung der Proteine. Darum war die relative Integration weiterer Lipide problematisch. Die zukünftige Version wird nicht mehr auf einem einzigen Protein basieren, sondern es ermöglichen, beliebig viele unterschiedliche Lipide einzufügen.

Darüberhinaus war die Anzahl der einfügbaren *PDB*-Dateien in die Membran beschränkt, weswegen es nicht möglich war, eine ausgeglichene Zusammensetzung der Membran zu realisieren: 50% Lipide würden bei einer Hepatozytenmembran 50% Proteinen gegenüber stehen. Die der Membran zugrundeliegende Lipidanzahl ist auf ein Minimum von ca. 30% der Gesamtmembran beschränkt, da die Verteilungsberechnung für die Lipide z.Zt. auf einem Zufallsalgorithmus basiert:

Dieser arbeitet alle in der Membran enthaltenen *PDB*-Dateien iterativ ab. D.h., er beginnt mit dem Protein, welches sich an oberster Stelle des Baumes befindet, berechnet die Anzahl der auf der Membran zu verteilenden Proteine, und verteilt sie dann auf der Membranoberfläche. Danach wird das nächste Protein gewählt und der gleiche Vorgang wird mit diesem ausgeführt. Folglich erfolgt die Berechnung der Positionen eines Proteintyps völlig unabhängig von den der anderen. Das hat zur Folge, dass sehr viel freier Platz verschwendet wird und letztendlich die Verteilungsberechnung abbricht, sobald der Zufallsalgorithmus keinen freien Platz mehr finden kann.

Folglich sind von dem am häufigsten vertretenen Phosphatidylcholin, welches für diese Membran als Grundlipid gewählt wurde, nicht 12% der Membran besetzt, sondern 32%. Die zukünftige Version des Membraneditors wird einen Algorithmus bieten, welcher die Platzierung der Komponenten adäquat berechnet.

3.1.2. amira®

Aufgrund des modularen Aufbaues in Bezug auf die Text- und Sounddateien (s. 2.3.3.2.3.) ist es relativ problemlos möglich, im Nachhinein eine andere Sprachversion für die Texte hinzuzufügen. Durch eine weitere, noch *DisplayCell* hinzuzufügende Auswahlmöglichkeit wird es dann möglich sein, die Sprache der Texte während des Programmablaufs zu ändern, z.B. von Deutsch auf Englisch.

Der DemoMaker unter amira® weist einen relativ unpraktische Fehlerquelle auf, welcher in Zukunft programmiertechnisch zu beheben ist, sofern dass bei der neuen Version von amira®, 4.0, nicht schon geschehen ist: Das Zurücksetzen eines abgelaufenen Demos führt dazu, dass einige TCL-kompatible Kommandos, welche sich zwischen dem Anfang und dem Ende des Demos befinden, ein weiteres Mal kurz ausgeführt werden. Dieser Fehler tritt besonders markant in den Vordergrund, wenn mehrere Demos hintereinander abgespielt werden sollen.

Ebenso ist amira® leider nicht in der Lage, beim CameraPath-Objekt eine gleich bleibende Position einer Kamera zu interpolieren. Das heißt, erstellt man zwei Keyframes, zwischen denen die Kamera theoretisch in ihrer Position

bleiben müsste, berechnet amira® eine längere oder sogar mehrere kurze Kamerabewegungen hinzu, welche zu einem unruhigen Bild führen. Um diesen Fehler zu umgehen, kann der Benutzer entweder versuchen, diesen Nachteil durch möglichst viele Keyframes auszugleichen, wozu der Einsatz eines Texteditors nötig ist, da der Zeitverlust ansonsten unter amira® noch größer wäre. Eine andere Möglichkeit ist, ab einem bestimmten Zeitpunkt das DemoMaker von dem CameraPath-Objekt per TCL-Kommando zu lösen. Das hat zur Folge, dass die Kamera an der letzten Position verweilt und erst ab dem gewünschten Punkt wieder auf den CameraPath gesetzt wird. Hierzu müssen die beiden Objekte wieder durch den DemoMaker verbunden werden.

Auch die Ausrichtung der Kamera des CameraPaths ist relativ ungenau zu regulieren, da die Größe des Kameraobjektes nicht dem Modell entsprechend skaliert wird.

3.2. Fazit

Im Rahmen meiner Masterarbeit wurden nun die Vorarbeiten aus den Vorgänger-Projekten zusammengefasst und überarbeitet. Zum einen erfolgte die Dokumentation und die Erstellung einer Forum-Plattform im Internet in Form einer Internetpräsenz. Zum anderen wurde der Membraneditor leicht modifiziert, die resultierende Membran in die amira®-Umgebung integriert und das Visualisierungsmodul überarbeitet und um einige grundlegende Funktionen erweitert. Eine dieser Funktionen ist die Darstellung von audio-visuellen Animationen, welche einen grundlegenden Einblick in die Funktionen einer Hepatozyten-Membran bieten und beliebig erweiterbar sind.

Somit wurde eine Infrastruktur geschaffen, auf welche Projektseminare, Bachelor- und Masterarbeiten aufbauen können. Es gilt nun, die Voraussetzungen optimal zu nutzen.

3.3. Ausblick

Die langfristige Vision des *CELLmicrocosmos*-Projektes ist die Entwicklung einer interaktiven 3D-Umgebung, welche strukturell und visuell eine Zelle simuliert.

Es soll ein Zellexplorer erschaffen werden, welcher die verschiedenen Granularitätsebenen in sich vereint:

Auf der obersten Ebene stehen die Zellmodelle, welche entweder mit dem Modellierungstool 3ds max® erstellt werden oder aber durch Zellmikroskopie importiert werden können. Hierfür wird eine intensivierete Auswertung mikroskopische Aufnahmen vonnöten sein. Dabei wird ein wissenschaftlich vertretbarer Kompromiß zwischen der technisch möglichen Echtzeit-Visualisierung und der qualitativ und quantitativ molekularbiologisch realistischen Darstellungsweise erarbeitet werden müssen.

Die unterste Granularitätsebene bilden die PDB-Membranen. Diese sollen durch den weiterzuentwickelnden Zellmembraneditor erstellt und automatisch integriert werden können. Hierzu wird eine intensive Recherche zur Zusammensetzung der unterschiedlichen Membranen der Zellkomponenten nötig sein.

Da die Ergebnisse z.B. zur Lage der einzelnen Lipide zueinander nicht in Rahmen dieses Projektes zu errechnen sind, muss es das Ziel sein, externe Forschungsergebnisse mit dem Editor zu korrelieren. So könnten z.B. die Ergebnisse der Fakultät für Physik, welche sich mit der Berechnung von vereinfachten Membranmodellen nach der „Condensed Matter Theory“ beschäftigt (Lenz/Schmid 2005), als Grundlage für die Positionierung der PDB-Modelle genutzt werden. Zu diesem Zweck könnten z.B. die Richtungsvektoren und die Abstände der Lipide aus dem physikalischen Modell übertragen werden.

Die PDB-Membranen sollen durch einen dynamisch konfigurierbaren Zelleditor erzeugt werden. Während die erste Version des Membraneditors aus *Cm2* bereits bei im Rahmen dieser Arbeit Verwendung gefunden hat, wird in dem für das Wintersemester 2006/2007 geplanten Projekt „CELLmicrocosmos 2.1“ eine neue Version erarbeitet werden.

3. Resultate

Das mittelfristige Ziel ist es, den Editor in amira® zu integrieren. Das heißt, das bereits in Amira integrierte Modul *DisplayCell* wird dahingehend erweitert werden, einen direkten Zugriff auf den Editor zu ermöglichen. So soll z.B. ein Button namens „Membran“ hinzugefügt werden, der den Membraneditor öffnet, in welchem man die Membran der aktuellen Komponente manipulieren, abspeichern und gleich in amira® wieder integrieren kann.

Aber nicht nur die Membran soll per Editor variierbar sein, auch die Zelle an sich muss vom Benutzer definiert werden können. Folglich muss es eine Editiermöglichkeit geben, um die Lage der Komponenten zueinander bestimmen zu können. Dabei muss immer die oberste Priorität die Visualisierbarkeit sein: Ein perfekt an die Realität angepasstes Model ist nicht mehr sinnvoll, wenn es hardwaretechnisch nicht explorierbar ist.

Die Zusammensetzung der Membran und der gesamten Zelle sollen dabei in einem XML-Format gespeichert werden und so für andere Zwecke exportierbar sein. Dabei bietet sich besonders das SBML-Format (System Biology Markup Language) an (Hucka, 2003).

Um die Performance des Visualisierungs-Systems unter amira® wahren zu können, wird eine Strategie erarbeitet werden müssen, um Überblendungen zwischen den unterschiedlichen Granularitätsebenen zu ermöglichen. Während die oberste Ebene permanent sichtbar sein kann, dürfen die unteren Ebenen erst ab einem zu definierenden Abstand des Viewers zu den Objekten eingeblendet werden.

Die Positionierung der Zellkomponenten kann darüberhinaus mit der von *Metabolischen* Netzwerken korreliert werden. Dieser Vorgang würde es ermöglichen, drei-dimensionale *metabolische* Pfade in Relation zur Position der Zellkomponenten explorieren zu können. So sollen solche Pfade des Mitochondriums, z.B. den Prozess der Oxidativen Phosphorylierung, oder den der Erzeugung von Ribosomenuntereinheiten im Nucleus und ihren Weg ins Zytoplasma, untersucht werden können. Das Resultat wird folglich teilweise eine Übertragung der zwei-dimensionalen *metabolischen* Pfade der Biochemical Pathways (Michal 1992) in den drei-dimensionalen Raum sein. Auf diese Weise soll die Analyse der Stoffwechselfade erleichtert und ein besseres Verständnis für die zellinternen Zusammenhänge ermöglicht werden. Hierzu könnte auf die Ergebnisse zweier Diplomarbeiten aus der AG Bioinformatik zurückgegriffen werden, welche von Claas Faber und Ralf Brand unter einer diesem Projekt ähnlichen Umgebung programmiert worden sind (Faber 2005; Brand 2005).

Wie bei der Darstellung der Granularitätsebenen, deren Abstraktionsniveau antiproportional zur Nähe des Betrachters zur Zellkomponente ist, so soll auch bei der Entwicklung die Top-Down-Methode angewandt werden. Folglich soll die Basis des Programmes dynamisch erweiterbar sein und Methoden geschaffen werden, mit deren Hilfe eine realistische Modell- und Membranverteilung berechnet werden kann.

In Bezug auf die Animationstechniken wäre eine Exportierfunktion sinnvoll. D.h., es sollte auf einfache Weise möglich sein, Animationen erstellen zu können, indem eine Liveaufzeichnung des Explorierungsvorganges vorgenommen wird. Auf diese Weise könnte die Lage der einzelnen Komponenten zueinander oder z.B. der Weg, den ein Lipid vom Endoplasmatischen Reticulum in den Golgiapparat und aus der Zelle hinaus zurücklegt, nachvollzogen werden.

Die Zielgruppe wären somit in erster Linie Biologen, Bioinformatiker oder auch Medizinstudenten, welche auf diese Weise ein besseres Verständnis für die räumlichen Zusammenhänge der Zelle, ihrer Komponenten und ihrer Reaktionswege erlangen könnten.

Somit soll ein visuell wie wissenschaftlich anspruchsvolles Frontend geschaffen werden, welches die Ergebnisse aus unterschiedlichen Bereichen der Zellforschung in sich vereint, diese in Echtzeit darstellen kann und so die Vision einer zellulären Mikrokosmos konsequent weiterverfolgt.

4. Anhang

4.1. Technik

4.1.1. Hardware

1 X 3 GHz X 2 Tower 2GB

2 X 3 GHz Tower 2GB

Sony Vaio PCG-GRX616MP

4.1.2. Software

Adobe® Photoshop® 7

ParallelGraphics Cortona® VRML-Client 4.2

discreet® 3ds max® 6 (heute Autodesk)

Eclipse 3.1.1

Sun Microsystems™ Java™ 2 Platform Standard Edition Runtime Environment Version 5.0

Macromedia® Dreamweaver® MX

Macromedia® Flash™ MX

Macromedia® Freehand® 10

Microsoft® Visual C++® 6.0

Microsoft® Windows® XP

TGS amira® 3.1, Mol-, Dev-Package (heute Mercury)

4.1.3. Technologien

Programmierung: C++, Java™

Scripting: TCL/TK

Web-Programmierung: PHP 4.4.1, CSS1.0, XHTML 1.0

4.2. Fachbegriffe

amira console

Die a. ist eine Eingabeaufforderungskonzole unter amira®, in welcher direkt auf Modelle, Bedienelemente und andere Objekte zugegriffen werden kann. I.d.R. wird dazu die Scriptsprache TCL, die Tool Command Language (s. 2.3.2.1.), genutzt.

CELLmicrocosmos

CELL: Die Hervorhebung dieses Begriffes soll den direkten Bezug zu dem zentralen Objekt dieses Projektes, der biologischen Zelle, herstellen.

microcosmos: Wird die Zielsetzung dieses Projektes auf einer anderen Begrifflichkeitsebene betrachtet, so gilt es, eine Welt zu schaffen, einen Miniatur-Kosmos, der den Nutzer visuell anspricht und gleichzeitig die informative Tiefe einer umfangreichen molekular-biologischen Datenbank bietet. Besonders die dreidimensionale, auf dem Stereoverfahren basierende Darstellung soll dem Nutzer das Gefühl vermitteln, in diesem virtuellen Kosmos integriert zu sein.

Während er nun auf der einen Seite temporär auf virtuelle Weise Bestandteil der Zelle wird, ist diese auf der anderen Seite realer Bestandteil von ihm selbst. Dieses bilaterale, wechselseitige Verhältnis soll durch den Begriff des „microcosmos“ hervorgehoben werden (Sommer 2004, S. 8).

Citratzyklus

Bei dem C. handelt es sich um eine Abfolge von Reaktionen, in deren Verlauf Acetyl-CoA zu CO₂ oxidiert wird (Cooper 1997, S. 645).

Cm

Die *CELLmicrocosmos*-Bachelorarbeit aus dem Jahr 2004, erstellt von Björn Sommer und betreut durch Dr. Dieter Lorenz (s. 2.1.1.).

Cm1

Das erste *CELLmicrocosmos*-Projektseminar mit drei Teilnehmern (Matthias Bartneck, Sebastian Janowski, Björn Sommer), welches unter der Leitung von Dr. Dieter Lorenz im Sommersemester 2005 stattgefunden hat (s. 2.1.2.).

Cm2

Das zweite *CELLmicrocosmos*-Projektseminar mit sechs Teilnehmern (Sebastian Janowski, Dominik Mertens, Jan Pieniak, Robert Schmieder, Björn Sommer, Martha Zakrzewski), welches unter der Leitung von Dr. Dieter Lorenz im Wintersemester 2005/2006 und Sommersemester 2006 stattgefunden hat (s. 2.1.3.).

Cortona® Viewer

Der C. ist ein VRML-Viewer, welcher als Plug-In für die gängigsten Browser heruntergeladen werden kann: „<http://www.parallelgraphics.com/products/cortona/>“.

metabolisch, Metabolismus

Der M. ist die lateinische Bezeichnung für Stoffwechselfvorgänge, zu denen z.B. die Photosynthese gehört. Sie dient also der Speicherung oder Weiterverarbeitung aufgenommener wachstumsfördernder Stoffe.

Display Cell

Bei D. handelt es sich um das Modul für die interaktive Zellvisualisierung unter amira®. Es basiert auf einer Kombination aus einem TCL-Script und C++-Programmen (s. 2.3.5.).

IvToSurface

s. *Surface*

Oxidative Phosphorylierung

Bei der O.P. (bzw. Atmungskette) wird Wasserstoff zu Wasser oxidiert. Bei dieser Reaktion wird ATP (adenosintriphosphat) erzeugt. Dieses wird durch eine Reihe von in der Mitochondrienmembran stattfindenden Reaktionen von Wasserstoffionen und Elektronen aus ADP (adenosindiphosphat) gebildet (Cooper 1997, S. 651).

PDB

Protein Data Base: Im Internet frei verfügbare 3D-Modelle von Protein- oder Nukleinsäure-Molekülen. Sie enthalten sämtliche Daten eines bestimmten Moleküls, welches über komplexe Suchfunktionen unter „www.rcbs.org“ gefunden werden kann. Zu den Daten zählen neben drei-dimensionalen Positionierungsangaben auch solche, mit deren Hilfe z.B. das elektrostatische Potenzial oder die Wasserstoffbrücken berechnet werden können.

Rendern, rendern

Das R. (von engl. to render) ist der Verarbeitungsvorgang von Rohdaten, aus denen durch unterschiedliche Verfahren neue Daten erzeugt werden.

4. Anhang

In diesem Fall handelt es sich um die Verarbeitung der drei-dimensionalen Modelle und ihrer Umgebung, die in *3ds max*® erzeugt wurden, zu zwei-dimensionalen Abbildungen. So wurden z.B. im Falle dieser Arbeit aus einer Datei, die drei-dimensionale Informationen enthält und ca. 60MB groß ist, Bilddaten von 5 GB erzeugt, welches Filmmaterial von 1.30 Minute lieferte.

Surface

Unter *amira*® ist ein S. ein Objekt, welches aus einer drei-dimensionalen Oberfläche besteht. Diese wiederum wird durch eine Vielzahl von drei- oder viereckigen Flächen gebildet, die auch Polygone genannt werden. Erzeugt werden kann ein S. z.B. über ein *IvToSurface*-Objekt, welches die interne, auf dem Open Inventor-Format basierende Datenstruktur in eine drei-dimensionale Fläche umwandelt. Dabei wurde das Open Inventor-Format vorher durch den Import einer externen Datei erzeugt, welche z.B. im *WRL*-Format importiert wurde.

Viewer

Der V. ist das Fenster unter *amira*®, in welchem auf drei-dimensionale Weise Objekte exploriert werden können. Dabei können verschiedene Instanzen des V.s geöffnet werden, um Modelle von unterschiedlichen Perspektiven gleichzeitig zu betrachten. Diese können wiederum alle individuell modifiziert werden.

VRML

V. (Virtual Reality Modeling Language) wird genutzt, um 3D-Modelle im Internet darstellen zu können. Dieses Dateiformat mit der Endung „.wrl“ (für world) wurde von Silicon Graphics® initiiert und wurde seitdem von unterschiedlichen Institutionen und Firmen weiterentwickelt. Es liegt in zwei Versionen vor: 1.0 und 2.0 bzw. 97. In naher Zukunft wird V. wohl durch das auf XML basierende X3D-Format (eXtensible 3D) ersetzt werden.

WRL

s. *VRML*

4.3. Danksagungen

Vielen Dank an alle, die mich bei meiner Arbeit unterstützt haben!

Besonderer Dank gilt:

Dr. Dieter Lorenz für die lange, sehr gute Zusammenarbeit, die Betreuung und interessante Aufgabenstellung in Verbindung mit dem Wunsch, auch weiterhin mit ihm an diesem Projekt arbeiten zu dürfen;

Prof. Ralf Hofestädt für die Bereitschaft, die Arbeit als Zweitgutachter zu betreuen und die Möglichkeit, dieses Projekt auch zukünftig weiterverfolgen zu dürfen;

allen Teilnehmern der *Cm*-Seminare für die gute Zusammenarbeit;

Matthias Bartneck für die guten Ratschläge zu den Texten;

Roswitha Hornschu für das geduldige Korrekturlesen;

Stefan Fuller für die guten Ratschläge in Bezug auf die Lernumgebung;

Michael Hertrich, Dirk Raufer, Marc Stache und Sebastian Janowski für die Tips zur Website;

und meiner Familie für die allzeit gute Unterstützung während meines Studiums.

Björn Sommer, Bielefeld den 28.08.2006

4.4. DisplayCell-Übersicht

Dieser Überblick zeigt die Objekte, welche unter amira® Verwendung finden und über das DisplayCell-Modul zentral reguliert werden können.

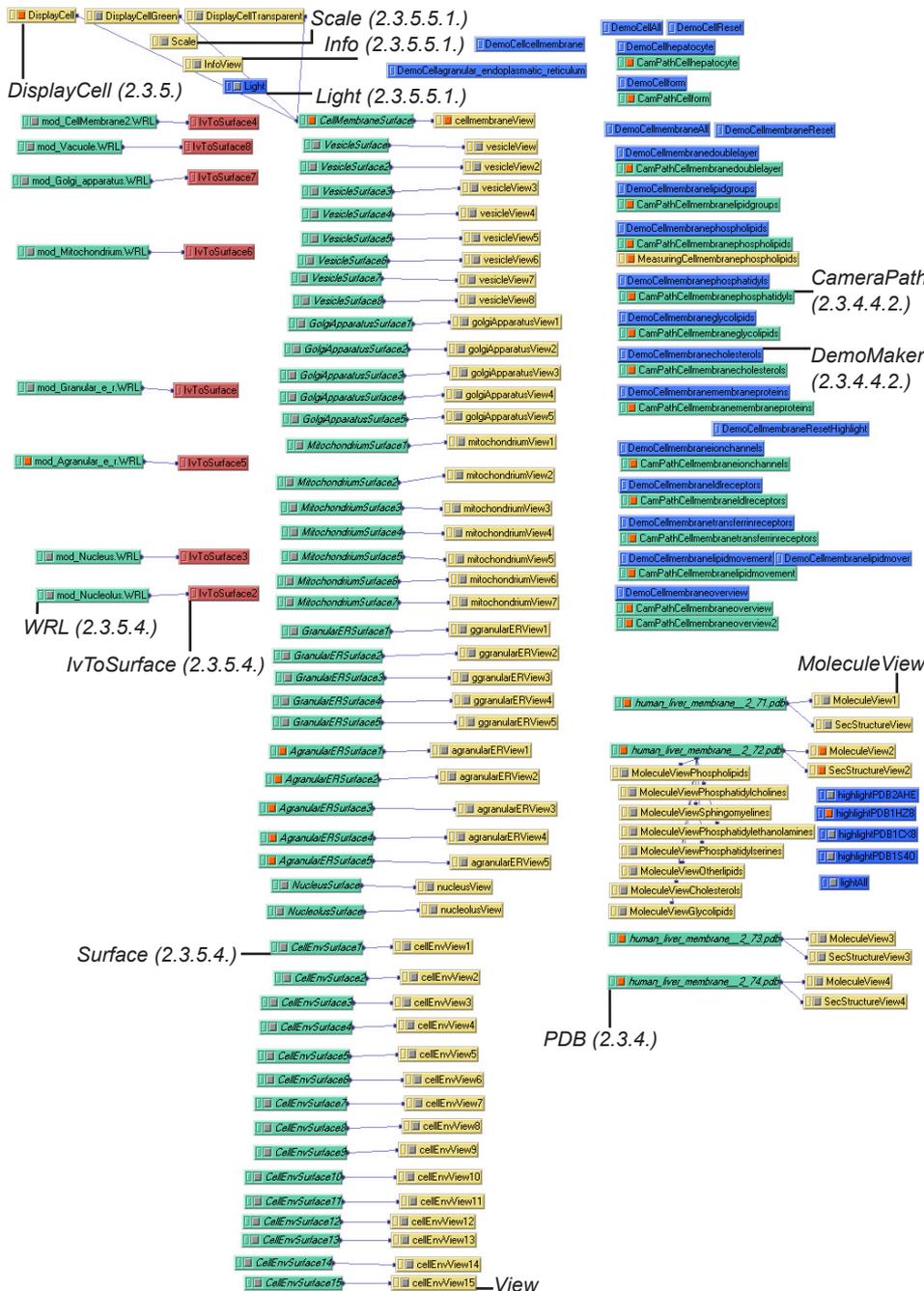


Abb. 2.51
Die Übersicht über alle amira®-Objekte mit Kapitelhinweisen

5. Quellen

5.1. Literatur

- Alberts, Bruce et al.: Molecular Biology of the Cell. Fourth Edition. New York: Garland Science, 2002.
- Alberts, Bruce et al.: Lehrbuch der Molekularen Zellbiologie, 3. Auflage. Weinheim: Wiley-VCH Verlag GmbH & Co. KgaA, 2005.
- amira extensions: amiraMol 3.0, amiraDeconv 3.0. Berlin: Konrad-Zuse-Zentrum für Informationstechnik, 2002.
- amira user's guide: amira 3.0. Berlin: Konrad-Zuse-Zentrum für Informationstechnik, 2002.
- Berman, H. M. et al.: The Protein Data Bank. In: Nucleic Acids Research. Volume 28. Oxford: Oxford University Press, 2000, S. 235–242.
- Berry, Michael N.; Edwards, Anthony M.; Barritt, Gregory J.: Isolated Hepatocytes. Preparation, Properties and Applications. In: Burdon, R. H.; Knippenberg, P. H.: Laboratory Techniques in Biochemistry and Molecular Biology. Volume 21. Amsterdam: Elsevier, 1991.
- Brand, Ralf: 3D-Visualisierung metabolischer Pfade mittels eines Kraftbasierten Layoutverfahrens. Diplomarbeit, Universität Bielefeld, Technische Fakultät, Arbeitsgruppe Bioinformatik, 05/2005.
- Chen, Ming; Hofestädt, Ralf: PathAligner. Metabolic Pathway Retrieval and Alignment. Applied Bioinformatics, 2004, 3(4): 241-252.
- Cooper, Geoffrey M.: The Cell. A molecular approach. Washington: ASM Press, 1997.
- Faber, Claas: Datenexploration metabolischer Pfade in 3D. Diplomarbeit, Universität Bielefeld, Technische Fakultät, Arbeitsgruppe Bioinformatik, 12/2005.
- Kahrmann, Klaus-Ove; Reise, Niels: Unmögliche Geschichten. Berlin: amBEATion / RANDLAGE, 1999.
- Kleywegt, G.J.; Jones, T.A.: Databases in protein crystallography. In: Baker, E. N.; Dauter, Z.: Acta Crystallographica Section D: Biological Crystallography. D54. Oxford: Blackwell Publications, 1998, S. 1119-1131.
- Lenz, Olaf; Schmid, Friederike: A Simple Computer Model for Liquid Lipid Bilayers. In: Journal of Molecular Lipids, Volume 117. Bethesda: American Society for Biochemistry and Molecular Biology (ASBMB), 2005, S. 147.
- Li, Xinhua; Weinman, Stefan A.: Chlorid Channels and Hepatocellular Function: Prospects for Molecular Identification. Galveston: Annual Reviews, 2002.
- Lodish, Harvey et al.: Molekulare Zellbiologie. 2. Auflage. Berlin: Walter de Gruyter, 1996.
- Michal, Gerhard: Biochemical Pathways Poster. Third Edition. Mannheim: Roche Diagnostics GmbH, 1993.
- Nelson, David; Cox, Michael: Lehninger Biochemie. Berlin: Springer Verlag, 2001.
- Sommer, Björn: CELLmicrocosmos. Zellvisualisierung. Bachelorarbeit, Universität Bielefeld, Technische Fakultät, Arbeitsgruppe Bioinformatik, 03/2004.
- Toner, Peter G.; Carr, Katharine E.: Cell Structure. London: Churchill Livingstone Edinburgh & London, 1971.
- Ude, Joachim; Koch, Michael: Die Zelle. Atlas der Ultrastruktur. Stuttgart, Gustav Fischer Verlag Stuttgart, 1982.

5.2. Internet

Cannon, Bryan et al. (2006): Klotho. Biochemical Compounds Declarative Database. URL: <http://www.biocheminfo.org/klotho/> [Stand 10.08.2006].

phpBB Group: phpBB. Creating Communities. URL: <http://www.phpbb.com> [Stand 19.08.2006].

W3C (2006): HTML-Validator. URL: <http://validator.w3.org> [Stand 20.08.2006].

School of Nursing (2004): The Hong Kong Polytechnic University. URL: <http://nhs.polyu.edu.hk/nhs/> [Stand 05.03.2004]

5.3. Abbildungen und Zusatzinformationen

Abb. 2.1: Zellen der erythropoetischen Reihe aus dem blutbildenden Gewebe der Maus. GefrierbruchTechnik. 10.000 X. Aufnahmen von Dr. H. Meyer, Jena. (Ude/Koch 1982, S. 27).

Abb. 2.2: Plasmacell. Tissue Human small intestine. Osmium fixation, lead staining. 32.000 X (Peter/Carr 1971, S. 133).

Abb. 2.3: : Leber-Zelle einer Maus gefärbt mit Haematoxylin und Eosin: Nuclei: blau/violett; Zellkernumgebung: rot/rosa (School of Nursing 2004)

5.4. Typogramm

Arial Regular 10/12;

Arial Bold 10/12;

Bauhaus Regular 18 pt;

links- / rechtsbündiger Flattersatz

Satzbreite 184,6 / 182,1 mm

5.5. Warenzeichen

3D Studio MAX, 3ds max, discreet sind entweder eingetragene Marken oder Marken der Autodesk, Inc. in den USA und/oder anderen Ländern.

Adobe und Photoshop sind entweder Marken oder eingetragene Marken der Adobe Systems Incorporated in den USA und/oder anderen Ländern.

amira ist eine eingetragene Marke des Konrad-Zuse-Zentrum für Informationstechnik Berlin.

backbuner ist entweder eine Marke der Autodesk Canada Inc. in den USA, Kanada und/oder anderen Ländern.

BSD ist eine eingetragene Marke von Björn Sommer.

Cortona ist eine eingetragene Marke von ParallelGraphics.

Firefox ist eine eingetragene Marke der Mozilla Foundation.

Google ist eine eingetragene Marke von Google, Inc..

Macromedia, Dreamweaver, Flash und FreeHand sind Marken oder eingetragene Marken von Macromedia, Inc. in den USA und/oder anderen Ländern.

Microsoft, Visual C++, Windows, Windows 2000, Windows XP sind entweder eingetragene Marken oder Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

Silicon Graphics ist eine eingetragene Marke von Silicon Graphics, Inc. in den USA.

Sun Microsystems und Java sind Marken oder eingetragene Marken von Sun Microsystems, Inc in den USA und anderen Ländern.

