

Professur für Grafische Datenverarbeitung
Fachbereich Informatik und Mathematik



Diplomarbeit

Konzeption und Implementierung
einer Kommunikation zwischen
Second Life und Web 2.0
Anwendungen

vorgelegt von
Krešimir Gronjak

Prüfer: Prof. Dr.-Ing. Detlef Krömker
Betreuerin: Sarah Voß

2. März 2009

Zusammenfassung

Im Rahmen dieser Arbeit entwickeln wir ein Konzept und einen Prototypen zur Kommunikation zwischen Second Life und Web 2.0-Anwendungen.

Die Motivation für diese Arbeit entstand durch die Veranstaltung „Einführung in das Projektmanagement“. In dieser wurden Meetings in Second Life abgehalten, wobei die Studierenden in der Übung Protokolle erstellt haben, um sie anschließend im Internet zu veröffentlichen. Die Protokollierung musste hierbei immer manuell durchgeführt werden und war dadurch fehleranfällig und nicht ausfallsicher. Ziel dieser Arbeit ist es, diese Lücke zu schließen und somit die Protokollierung automatisch und ausfallsicher zu machen.

Die Arbeit ist wie folgt aufgebaut: Im ersten Kapitel geben wir eine kurze Einführung in die Problemstellung. Im zweiten Kapitel behandeln wir die Grundlagen, die für das Verständnis der Arbeit notwendig sind. Wir beschreiben Second Life, die virtuelle Umgebung von Linden Lab. Außerdem stellen wir Blog und Wiki als Repräsentanten von Web 2.0 vor. Fälschlicherweise wird Second Life zu den Web 2.0-Anwendungen hinzugezählt. Deshalb zeigen wir eine klare Abgrenzung zwischen diesen Technologien auf.

In der Analyse betrachten wir Technologien, Möglichkeiten und Einschränkungen von Second Life. Wir untersuchen bestehende Verbindungen zwischen Second Life und Web 2.0. Desweiteren stellen wir Mashups vor, die eine Verbindung zwischen Second Life und Flickr, bzw. Delicious herstellen.

Dies führt uns direkt zum Konzept mit welchem wir eine Architektur beschreiben, die eine Verbindung zwischen Second Life und Web 2.0 darstellen soll. Wir unterteilen hierbei das Konstrukt in drei Teile: dem Objekt in Second Life, dem HTTP Supervisor und dem Wiki Bot.

Wir beweisen mit der Implementierung des Prototyps, dass die Kommunikation zwischen Second Life und Web 2.0 möglich ist. Als Repräsentant von Web 2.0 dient MediaWiki.

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Diplomarbeit ohne fremde Hilfe und nur unter Verwendung der zulässigen Mittel sowie der angegebenen Literatur angefertigt habe.

Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Frankfurt am Main, den 2. März 2009

Krešimir Gronjak

Inhaltsverzeichnis

Abbildungsverzeichnis	V
Tabellenverzeichnis	VII
Abkürzungsverzeichnis	VIII
1 Einleitung	1
1.1 Motivation und Aufgabenstellung	1
1.2 Zielsetzung	4
1.3 Aufbau der Arbeit	5
2 Grundlagen	6
2.1 Die Veranstaltung „Einführung in das IT-Projektmanagement“	6
2.2 Second Life	7
2.2.1 Entstehung	7
2.2.2 Avatar	10
2.2.3 Objekte	10
2.3 Second Life, Web 2.0 und MMORPGs	13
2.3.1 Gemeinsamkeiten von Web 2.0 und Second Life	14
2.3.2 Unterschiede zwischen Web 2.0 und Second Life	14
2.3.3 Die Gemeinsamkeiten von MMORPGs und Second Life	17
2.3.4 Die Unterschiede zwischen MMORPGs und Second Life	18
2.3.5 Second Life und MUVES	20
2.3.6 Zusammenfassung	21
2.4 Web 2.0	21
2.4.1 Entstehungsgeschichte	23
2.4.2 Blog	24
2.4.3 Blogs in der Bildung	25
2.5 Wiki	26
2.5.1 Entstehungsgeschichte	27

2.5.2	Lizenzen	28
2.5.3	Syntax	28
2.6	Wiki-Klone	29
3	Analyse	33
3.1	Second Life	34
3.1.1	Linden Scripting Language	34
3.1.2	Second Life und LMS	36
3.1.3	Kommunikation und Interaktion	37
3.2	Second Life-Konkurrenz	40
3.2.1	Active Worlds	40
3.2.2	Croquet	41
3.2.3	Nachteile von Croquet	43
3.2.4	Google Lively	44
3.3	Web 2.0	44
3.3.1	Architektur von Wiki-Systemen	45
3.3.2	Wiki-Bots	46
3.3.3	Links	48
3.4	Konkurrenzprodukte	49
3.4.1	Sloodle	49
3.4.1.1	Registrierung und Anmeldung	50
3.4.2	Virtual World Mashups	51
3.4.2.1	Delicious	52
3.4.2.2	Flickr	52
3.4.2.3	Mashups	53
3.5	Anforderungen	53
4	Konzept	56
4.1	Umsetzung der Anforderungen	57
4.1.1	Teil 1: Second Life	57
4.1.1.1	Protokollierung von Gesprächen	58
4.1.1.2	Verschicken von E-Mails	59
4.1.1.3	Anlegen eines Artikels in MediaWiki	60
4.1.1.4	Öffnen eines Browsers	62
4.1.1.5	Zugriff auf die Kalenderfunktion	62

4.1.1.6	Verschicken von Notecards	62
4.1.1.7	Kommunikation mit Second Life	63
4.1.1.8	Sensoren	65
4.1.2	Teil 2: HTTP Supervisor	65
4.1.3	Teil 3: MediaWiki	68
4.1.3.1	Automatischer Zugriff auf die MediaWiki-Inhalte	68
4.1.3.2	Wiki-Bots	73
4.1.3.3	Speicherung der Protokollierung	75
4.1.3.4	Anlegen eines Artikels in MediaWiki	75
4.1.3.5	Speicherung einer Notecard	76
4.1.3.6	E-Mail Versand und Archivierung	76
4.1.3.7	Kalenderfunktion	77
4.1.3.8	Formatierung in MediaWiki	78
4.1.3.8.1	Namensgebung	78
4.1.3.8.2	Formatierung einer Seite	78
4.2	Kommunikationsprotokolle	79
4.2.1	HTTP	79
4.2.2	XMLRPC	81
4.3	Client/Server Modell	81
4.3.1	Kommunikation im Client/Server Modell	82
4.4	Bedeutung für die Veranstaltung	85
5	Implementierung	87
5.1	Umsetzung	87
5.1.1	Die Funktionen	88
5.1.1.1	Second Life Implementierung	88
5.1.1.2	Die Server-Implementierung	91
5.1.2	Benutzerarten	91
5.1.3	Pseudo Zufallszahlen-Generator	91
5.1.4	Nachrichtenübermittlung	92
5.1.5	Wiki-Bot	93
5.1.6	Initialisierung	94
5.1.7	Aufbau der Verbindung und Anmeldung des Moderators .	95
5.1.8	Touch - Berühren des Objekts	96
5.1.9	Abmeldung	96

5.1.10	Protokollierung	97
5.1.11	Notecard verschicken	98
5.1.12	Wiki-Seite anlegen	99
5.1.13	E-Mail	100
5.1.14	Kalenderfunktion	102
5.2	Probleme und Lösungsansätze	104
6	Zusammenfassung und Ausblick	107
	Literaturverzeichnis	109

Abbildungsverzeichnis

2.1	Die Universität von Princeton in Second Life	10
2.2	Ein Avatar in Second Life	11
2.3	Das Tool zum Verändern der Kleidung und der äußeren Erscheinungsform des Avatars	12
2.4	Die Second Life Sandbox	12
2.5	Bauen in Second Life	13
2.6	Darstellung des Frankfurter Hauptbahnhofs in Second Life	19
2.7	Web 2.0	22
3.1	Erstellung eines Primitiv in Second Life	35
3.2	Eine typische Dialogbox in Second Life	38
3.3	Simulation von Second Life innerhalb von Croquet	43
4.1	Schematische Darstellung des Konzepts der Arbeit	58
4.2	Protokollierung von Gesprächen	59
4.3	Verschicken von E-Mails	60
4.4	Grafische Darstellung des Anlegens der Artikel	61
4.5	Grafische Darstellung des Verschickens einer Notecard	63
4.6	Speicherung in temporäre Dateien	66
4.7	Oberfläche zum Bearbeiten des Inhalts	68
4.8	Die Anmeldung in MediaWiki mit der anschließenden Antwort.	69
4.9	Die Abmeldung in MediaWiki mit anschließender Antwort.	70
4.10	Query zum Erfragen des <i>edit</i> -Tokens und des Zeitstempels.	71
4.11	Aktion zum Bearbeiten des Inhalts mit der anschließenden Antwort.	71
4.12	Speicherung der Protokollierung in MediaWiki	76
4.13	Anlegen eines neuen Artikels oder Anhängen des Inhalts	77
4.14	Ein Beispielcode für ein XMLRPC Aufruf	82
4.15	Client-Server Modell	83

4.16	Client-Server Modell mit mehreren Servern	83
4.17	Der Nachrichtenaustausch in einer Client-Server-Server Umgebung	84
5.1	Implementierung der Anforderungen	88
5.2	Implementierung aus der Sicht des Moderators.	89
5.3	Implementierung aus der Sicht des normalen Benutzers	92
5.4	Initialisierung des Programms und Anmeldung des Moderators . .	95
5.5	Auswahlfunktion aus der Sicht des Moderators	96
5.6	Die Protokollfunktion in Second Life	97
5.7	Die Protokollseite Testmeeting	98
5.8	Anlegen einer Wiki-Seite. Ansicht der Second Life-Funktion, ohne das Anhängen der Inhalte an eine bestehende Seite.	100
5.9	Die Schritte zur Erstellung einer E-Mail	101
5.10	Die Kalenderfunktion und die Schritte in Second Life	102

Tabellenverzeichnis

2.1	Die verschiedenen Blogarten.	24
4.1	Die Befehle in einem Datenpaket, das an den HTTP Supervisor geschickt wird.	64
4.2	Erforderliche Daten für die Kommunikation mit dem MediaWiki und dem Second Life Server	67
4.3	Parameter der Anmeldung	69
4.4	Fehlermeldungen bei der Anmeldung	70
4.5	Parameter zum Bearbeiten einer WikiSeite mit API.	72
4.6	Fehlermeldungen, die beim Bearbeiten der Seite auftreten können	73
4.7	Die Familien der Statuscodes in dem HTTP Anwendungsprotokoll	80
4.8	Meldungen, auf die der Prototyp reagieren muss	81
5.1	Verwendete Technologien in Second Life	87
5.2	Die acht Skripte der Second Life Implementierung.	89
5.3	Befehle in Second LIfе	93

Abkürzungsverzeichnis

- AJAX** Asynchronous JavaScript and XML
- API** Application Programming Interface
- AWEDU** ActiveWorlds Educational Universe
- CMS** Content Management System
- GPL** General Public License
- GUI** Graphical User Interface
- HTML** HyperText Markup Language
- HTTP** HyperText Transfer Protocol
- IIS** Internet Information Service
- LGPL** Lesser General Public License
- LMS** Learning Management System
- LSL** Linden Scripting Language
- MMORPG** Massively Multiplayer Online Role Playing Game
- MUVE** Multiuser Virtual Environment
- MUD** Multi-User Dungeons
- PHP** Hypertext Preprocessor (urspr.: Personal Home Page)
- RPC** Remote Procedure Call
- TCP** Transmission Control Protocol

IP Internet Protocol

URL Universal Ressource Locator

WWW World Wide Web

XML eXtensible Markup Language

XMLRPC eXtensible Markup Language Remote Procedure Call

Kapitel 1

Einleitung

1.1 Motivation und Aufgabenstellung

Die Anwendungsmöglichkeiten von Second Life ¹ sind vielfältig. Eine Möglichkeit ist, Konferenzen und Meetings im virtuellen Raum abzuhalten, wobei die Teilnehmer auf verschiedenen Teilen der Erdkugel sitzen.

Second Life ist weiterhin ein interessantes Gebiet für die Aus- und Weiterbildung. Mit Hilfe der virtuellen Welt ist es nun nicht mehr erforderlich, sich an einem Ort zu versammeln, um Informationen auszutauschen oder Vorträge zu halten. Second Life kann auch auf verschiedene Weisen eingesetzt werden, da es viele Möglichkeiten der Zusammenarbeit bei einem Treffen bietet. Im dreidimensionalen Raum können manche Beispiele so anschaulich verdeutlicht werden, wie es in einem normalen Hörsaal nie möglich sein wird.

Die virtuelle Umgebung gibt uns die Möglichkeit, 3D-Modelle im Klassenzimmer erscheinen zu lassen. Diese können von den Studierenden von allen Seiten betrachtet und wenn nötig verändert werden, um den Lehrstoff besser zu verstehen. Außerdem können die Studierenden verschiedene Aufgaben in Second Life auf diese Art und Weise gemeinsam bearbeiten.

Ein weiterer Vorteil ist, dass der Avatar, der eine Person im Second Life darstellt, auch eine gewisse Distanz und Anonymität bietet, die manchem Teilnehmer in Vorlesungen und Übungen fehlt. Dadurch können die Studierenden ihre Fragen unbefangen stellen, ohne sich dabei unsicher zu fühlen. Gerade diese Kommunikation oder Rückfragen fehlen in vielen Vorlesungen und Übungen, so dass es Dozenten nicht möglich ist, auf die Probleme der Studierenden einzugehen.

¹vgl. Kapitel 2.2

Im Rahmen der Veranstaltung „Einführung in das IT-Projektmanagement“ wurde versucht, die Möglichkeiten, die Second Life bietet, aufzuzeigen und zu nutzen. Im Wintersemester 2007/08 wurde ein Teil der Übungen zu der Vorlesung im Second Life abgehalten. Dabei wurden die Studierenden in Gruppen eingeteilt, die alle eigene Aufgaben im Software-Entwicklungsprozess hatten. Innerhalb der Gruppen nahmen die Studierenden verschiedene Rollen an. Die Rollen halfen ihnen, besser zu verstehen, wie die Projektierung einer Anwendung funktioniert, und wo es zu Problemen kommen kann. Außerdem wurden die Gruppen ermutigt, miteinander zu kommunizieren, damit sie die Schnittstellen zwischen diesen Gruppen besser verstehen und optimal ausnutzen.

Nach jedem Meeting stellte der Moderator die Protokolle des Textchats ins Internet, damit sie von anderen Studierenden eingesehen werden konnten. [Bal98] Hier wurden fehlende Funktionalitäten von Second Life sichtbar. Bei einem Meeting ist es wichtig, die besprochenen Themen in schriftlicher Form festzuhalten. Obwohl es möglich ist, Informationen ins Second Life zu übertragen und diese dort auch anzuzeigen, sind die Möglichkeiten für den Informationsfluss aus der virtuellen Umgebung noch dürftig. Insbesondere gibt es keine Möglichkeit, die Gespräche zu protokollieren, obwohl diese Protokolle für eine spätere Nachbearbeitung der Meetings und für Studierende, die am Meeting nicht teilgenommen haben, wichtig wären.

Dabei sind die Möglichkeiten der Speicherung und Archivierung solcher Meetings außerhalb von Second Life vielfältig. In den Zeiten von Web 2.0² ist das Erstellen von Wikis³, Blogs⁴ und anderen Webinhalten auch für normale Internet-Nutzer einfach und intuitiv geworden. Dem Web 2.0 ist es zu verdanken, dass Blogs immer zahlreicher werden. Wikipedia, die freie Enzyklopädie [wik09a], ist auch ein mächtiges Nachschlagewerkzeug geworden, welches zur Speicherung von solchen Protokollen benutzt werden kann. Was fehlt, ist eine Verbindung zwischen diesen beiden Werkzeugen: Second Life und Web 2.0-Anwendungen. Die Vorteile solcher Verbindungen sind ersichtlich. Eine Vorlesung, die nur gehört wird, von der aber keine Aufzeichnungen existieren, gerät sehr schnell in Vergessenheit. Die Aufzeichnungen manuell zu erstellen, ist sehr umständlich und fehlerbehaftet. Es passiert nämlich sehr schnell, dass dieser letzte Schritt nach dem Zusammentref-

²vgl. Kapitel 2.4

³vgl. Kapitel 2.5

⁴<http://www.itwissen.info/definition/lexikon/weblog-Blog-Weblog.html>

fen vergessen wird. Die Daten, die für den Lernprozess dringend benötigt werden, sind dann nämlich unwiederbringlich verloren, und Studierenden fehlt eine wichtige Lernquelle.

Der zweite Punkt ist das Erstellen der Wikiseiten. Obwohl die Werkzeuge zur Erstellung dieser Seiten mittlerweile ziemlich intuitiv geworden sind, erfordert die manuelle Erstellung der Wikiseiten immer noch eine Nachbearbeitungszeit, die nicht zu vernachlässigen ist. Diese Zeit hat der Dozent nicht immer. Dadurch kann es passieren, dass das Material erst einige Tage später online ist. Mit Hilfe der Automatisierung ist es möglich, das Protokoll sofort nach dem Meeting ins Netz zu stellen. Die Nachbearbeitung beschränkt sich hierbei auf das Durchlesen des Protokolls und die Korrektur von Rechtschreibfehlern.

Wenn diese Automatisierung zuerst einmal für Wikis funktioniert, ist es relativ einfach, sie auch für andere Web 2.0-Anwendungen zu erweitern. Durch diese Erweiterung wird Second Life um ein Werkzeug bereichert, das bspw. die Meetings im virtuellen Konferenzraum einfacher, effizienter und produktiver gestaltet.

1.2 Zielsetzung

Das Ziel dieser Diplomarbeit ist, eine Verbindung zwischen Second Life und Web 2.0 zu schaffen. Die Verbindung soll hauptsächlich Informationen aus Second Life extrahieren. Ziel ist es, einen Prototypen zu schreiben, der es dem Benutzer ermöglicht, die Daten aus Second Life zu extrahieren. Dieser Prototyp wird danach das Weiterleiten der Rohdaten an den Wiki-Server steuern. Dabei soll es möglich sein zu konfigurieren, welches Dokument verschickt werden soll, an welchen Server diese Daten geschickt und wie sie am Server weiterverarbeitet werden sollen. Eine weitere Möglichkeit ist, die Termine auf einer Webseite zu veröffentlichen. Termine, die sich bereits auf der Seite befinden, werden nicht gelöscht, wenn neue Termine angelegt werden, sondern geordnet eingefügt.

Weiterhin soll auch die Möglichkeit gegeben sein, den Chat eines wichtigen Meetings aufzuzeichnen, um ihn dann später auf einer Webseite zur Verfügung zu stellen. Hierbei ist es wichtig zu sagen, dass Second Life in einer Programmiersprache implementiert ist, die in ihrem Umfang stark eingeschränkt ist. Die Einschränkung der Linden Scripting Language (LSL)⁵ im Auge behaltend, wird es möglich sein, die Daten durch ein geeignetes Kommunikationsprotokoll zu übertragen. Die Kommunikation wird in beide Richtungen erfolgen. Diese Arbeit konzentriert sich auf die Kommunikation von Second Life zu den Web 2.0-Anwendungen, aber der umgekehrte Weg wird auch nicht vernachlässigt. Hierbei wird das Augenmerk nicht auf die Visualisierung von Webseiten im Second Life gelegt, sondern auf die Sicherstellung der Kommunikation. Am Ende wird in dieser Richtung eine gute Ausgangsbasis für die Visualisierung vorgelegt werden.

Die Daten, die im ersten Bereich verschickt wurden, müssen auf dem Server gesammelt werden. Wenn die Kommunikation mit dem Second Life-Server abgeschlossen ist, wird ein entsprechendes Dokument aufgebaut und veröffentlicht. In dieser Diplomarbeit konzentrieren wir uns vor allem auf die automatische Erstellung der Wikiseiten.

⁵vgl. Kapitel 3.1.1

1.3 Aufbau der Arbeit

Die schriftliche Ausarbeitung dieser Arbeit ist in sechs Kapitel aufgeteilt. Das erste Kapitel, die Einleitung, macht den Leser mit dem Thema vertraut.

Mit Grundlagen, dem Kapitel 2 der Diplomarbeit, werden die einzelnen Anwendungen, wie Second Life und Wiki, erklärt. Außerdem diskutieren wir, ob Second Life ein Teil von Web 2.0 ist oder nicht. Anschließend werden das Wiki allgemein und die verschiedenen Wiki Klone vorgestellt.

Im dritten Kapitel, der Analyse⁶, wird die Ausgangssituation untersucht. Dabei wird besonderes Augenmerk auf die Konkurrenzprodukte⁷ zu Second Life und auf die Errungenschaften, was die Verbindungen zwischen Second Life und Web 2.0 anbelangt⁸, gelegt. Anschließend werden die Anforderungen an das zu entwickelnde System formuliert⁹. Im Kapitel Konzept¹⁰ werden die einzelnen Komponenten des Prototyps vorgestellt, die sich aus den Anforderungen ergeben. Danach werden die Kommunikationsprotokolle¹¹ erläutert, die in der Arbeit Einzug gefunden haben. Anschließend wird das Client/Server Modell¹² besprochen, um den Prototypen und seinen Aufbau besser zu verstehen. Am Ende des Kapitels wird noch einmal Bezug auf die Veranstaltung „Einführung in das IT-Projektmanagement“ genommen und die Auswirkungen dieser Arbeit darauf¹³.

Das Kapitel 5, die Implementierung, behandelt die Umsetzung des Konzepts. Dabei wird vor allem auf die verwendeten Technologien, die realisierten Funktionen und die Probleme eingegangen, die sich bei der Implementierung ergaben.

Das letzte Kapitel enthält die Zusammenfassung¹⁴ der Arbeit und schließt mit dem Ausblick auf weitere Entwicklungen ab. In diesem Kapitel werden auch Ideen und Lösungsansätze für aufbauende Arbeiten gegeben.

⁶ vgl. Kapitel 3

⁷ vgl. Kapitel 3.2

⁸ vgl. Kapitel 3.4

⁹ vgl. Kapitel 3.5

¹⁰ vgl. Kapitel 4

¹¹ vgl. Kapitel 4.2

¹² vgl. Kapitel 4.3

¹³ vgl. Kapitel 4.4

¹⁴ vgl. Kapitel 6

Kapitel 2

Grundlagen

In diesem Kapitel werden die Grundlagen behandelt, die für das Verständnis der Arbeit notwendig sind. Zunächst erfolgt im Abschnitt 2.1 eine kurze Erläuterung der Veranstaltung „Einführung in das IT-Projektmanagement“ von Herrn Vuong und Frau Voß, die zu der Idee dieser Diplomarbeit geführt hat. Anschließend wird im Abschnitt 2.2 Second Life beschrieben, in dem Teile der Arbeit ausgeführt werden. Als nächstes wird die Frage geklärt, ob Second Life als Teil von Web 2.0 angesehen werden kann oder ob es sich doch um ein eigenständiges Gebiet handelt ¹. Dabei wird auch kurz auf die heute sehr beliebten Massively Multiplayer Online Role Playing Game (MMORPG) eingegangen und die Unterscheidung zwischen ihnen und Second Life erläutert. Nachdem diese Frage geklärt wurde, wird auch beschrieben, was sich hinter dem Namen Web 2.0 verbirgt ². Die Wikis sind dabei ein wichtiger Unterpunkt, weil der Prototyp eine Verbindung zwischen Second Life und MediaWiki darstellen wird. Deswegen werden die Wiki-Systeme und eine Auswahl von Klonen im Abschnitt 2.5 vorgestellt.

2.1 Die Veranstaltung „Einführung in das IT-Projektmanagement“

In der Veranstaltung „Einführung in das IT-Projektmanagement“ werden den Studierenden die Phasen eines Projekts, die Managementaufgaben und die Werkzeuge vorgestellt. Die Werkzeuge sind unter anderem Anforderungsmanagement, Projektorganisation, Planung und Steuerung, Vorgehensmodelle für die Entwick-

¹vgl. Kapitel 2.3

²vgl. Kapitel 2.4

lung, Wasserfallmodell, Objektorientiertes Modell, Konfigurationsmanagement, Qualitätsmanagement, Gruppendynamik, Management der technischen Lösung, Risikomanagement, Qualitätsmanagement usw.

In der Übung sollen die Studierenden die verschiedenen Management-Methoden und Management-Werkzeuge einsetzen und beurteilen [gdv08]. Dabei werden die Studierenden in Gruppen eingeteilt, die verschiedene Aufgaben lösen. Die einzelnen Aufgaben sind miteinander verknüpft und erfordern eine Kommunikation zwischen den Gruppen. Deshalb treffen sich Repräsentanten der Gruppen im Second Life, um weiteres Vorgehen zu besprechen und die gelösten Aufgaben zu präsentieren. Einer der Studierenden wird angehalten, das Meeting zu halten. Da nur ein Teil der Studierenden beim Meeting anwesend ist, muss ein Protokoll festgehalten werden, damit die restlichen Studierenden die Ergebnisse des Meetings nachvollziehen können. Deswegen muss sich der Moderator um die Protokollierung des Meetings kümmern.

Das Meeting wurde protokolliert, indem der Textchat vom Moderator kopiert und ins Internet gestellt wurde. Dabei musste er die einzelnen Schritte manuell durchführen. Diese Vorgehensweise war unsicher und fehleranfällig. Falls der Moderator vergessen hat, die Protokollierung durchzuführen, waren die Daten unwiederbringlich verloren.

2.2 Second Life

Second Life ist eine Online-3D-Architektur, die es Benutzern ermöglicht, sich in einer virtuellen Welt zu bewegen und diese nach ihren Bedürfnissen zu gestalten. Die Benutzer können sich in Second Life bewegen, fliegen, sich mit anderen Benutzern unterhalten, Land kaufen, bauen oder miteinander Handel betreiben.³

2.2.1 Entstehung

Um auf die Geschichte von Second Life zu verstehen, muss auch die Geschichte der Idee von virtuellen Welten bekannt sein. In diesem Zusammenhang werden wir einen kurzen geschichtlichen Hintergrund geben.

³vgl. Was ist Second Life <http://de.secondlife.com/whatis>

Die virtuellen Welten aus den frühen Jahren des World Wide Web (WWW) beschränkten sich auf textbasierte Rollenspiele.[Alb07] Diese wurden auch Multi-User Dungeons (MUD) genannt. Die Rollenspiele hatten noch keine grafischen Oberflächen und beschränkten sich deswegen auf eine textuelle Beschreibung der Ereignisse und Räume. Die Benutzer lösten Aufgaben mit Hilfe von Textkommandos.

Die Literatur war zu diesem Zeitpunkt schon weiter. Dabei prägte William Gibson schon im Jahr 1984 in seinem Buch „Neuromancer“⁴ den Begriff *Cyberspace*. Dieser Begriff wurde oft als Synonym für das WWW angesehen. Unter anderem taucht in dem Roman auch eine *Matrix* auf, die wiederum eine virtuelle Realität des *Cyberspace* darstellt.

Neal Stephenson beschrieb 1992 in seinem Buch „Snow Crash“⁵ eine virtuelle Welt im Internet. Daraufhin wurde auf der Basis des „Metaverse“, wie Stephenson seine virtuelle Welt nannte, das Active Worlds vorgestellt. Active Worlds wurde kein großer Erfolg, und die dahinter stehenden Unternehmen kämpften ums Überleben. Die Gründe dafür sind zum einen die fehlende Kundschaft und zum anderen die fehlende Technologie. Das Internet war zu dem Zeitpunkt noch zweidimensional und schnelle Anbindungen waren nicht weit verbreitet. Active Worlds war seiner Zeit voraus.

2003 stellte Linden Lab⁶ die virtuelle Welt Second Life vor. Der Zeitpunkt für Second Life war gut gewählt, weil die Breitbandverbindungen schon in vielen Haushalten Einzug gefunden hatten, und die Computertechnologie durch die Spieleindustrie so weit ausgereift war, dass sie die virtuelle Welt unterstützen konnte.

Second Life fand große Unterstützung bei Menschen und Unternehmen. Die Privatpersonen waren von den Möglichkeiten, die Second Life bot, begeistert. Die Unternehmen sahen im „zweiten Leben“ eine gute Verdienstmöglichkeit. Sofort entstanden Communities, und nach anfänglichen Schwierigkeiten schien Second Life das World Wide Web revolutionieren zu können. Der Hype zu Second Life hat sich heute schon abgeschwächt. Die meisten Parzellen haben keine Besucher. Die Unternehmen merkten schnell, dass die erhoffte Kundschaft meistens ausblieb⁷.

⁴vgl. <http://www.williamgibsonbooks.com/books/neuromancer.asp>

⁵vgl. <http://www.hinternet.de/buch/s/stephenson2.php>

⁶vgl. <http://lindenlab.com/>

⁷vgl. http://www.hr-online.de/website/rubriken/nachrichten/indexhessen34938.jsp?rubrik=36082&key=standard_document_35247176

In der neueren Zeit entdecken viele Ausbildungsstätten die Vorteile von Second Life. Mittlerweile haben verschiedene Universitäten eigene Inhalte in Second Life gestellt. Second Life ist zu einer festen Größe geworden. Es wird das zweidimensionale Internet zwar nicht ersetzen, weil das World Wide Web durch Suchmaschinen und Web 2.0-Anwendungen einige Vorteile bietet, aber es wird zu einer willkommenen Ergänzung. Zur Zeit hat Second Life mehr als 11 Millionen registrierte Nutzer.[sl101]



Abbildung 2.1: Die Universität von Princeton in Second Life

2.2.2 Avatar

Ein Avatar ist eine digitale Figur in Second Life, mit der sich die Benutzer identifizieren und in Second Life auftreten. Das Aussehen des Avatars kann beliebig verändert werden. Dadurch soll der Benutzer die Möglichkeit erhalten, seine Individualität auszudrücken. Die Benutzer wählen das Geschlecht des Avatars frei aus.

Die Avatare haben bei der Anmeldung menschliches Aussehen. Diese können andere Texturen bekommen und somit ihr Erscheinungsbild ändern. Die Möglichkeiten dafür sind im Second Life gegeben.

Die Avatare müssen nicht unbedingt menschlich aussehen. Es gibt auch spezielle Oberflächen, durch die diese in Fabelwesen oder andere nicht menschlich aussehende Geschöpfe umgestaltet werden können.

2.2.3 Objekte

Durch die integrierten Tools für die Inhaltserstellung wird es dem Benutzer ermöglicht, eigene Objekte in Echtzeit zu erstellen. Die Benutzer, die durch Avatare



Abbildung 2.2: Ein Avatar in Second Life

dargestellt werden, können bei der Erstellung zusammenarbeiten und sich somit die Arbeit teilen. Bei der Erstellung der neuen Objekte sind diese nur durch den zur Verfügung stehenden Platz beschränkt. Normalerweise ist es notwendig Land zu kaufen, um darauf bauen zu können. Viele Benutzer scheuen aber den finanziellen Aufwand, der damit einhergeht oder wollen in Second Life nur „schnuppern“, um sich einen Überblick zu verschaffen. Für diese Benutzer wurden sogenannte Sandboxes geschaffen.

Eine Sandbox ist eine Landfläche im virtuellen Raum von Second Life, auf der jeder nach seinen Wünschen beliebige Objekte bauen kann. Diese Landfläche hat die Eigenschaft, dass sie nach zwei Stunden alle Objekte löscht, die sich darin befinden. Dadurch bildet sie ein perfektes Terrain zum Ausprobieren der Werkzeuge für die Objekterstellung. Dabei können die Bewohner diese Objekte auch verkaufen und damit die lokale Währung, die Lindendollar erwerben. Die Lindendollar können wieder in reales Geld umgetauscht werden, so dass die Bewohner dadurch ein eigenes Unternehmen im Second Life aufbauen können.

KAPITEL 2 GRUNDLAGEN



Abbildung 2.3: Das Tool zum Verändern der Kleidung und der äußeren Erscheinungsform des Avatars

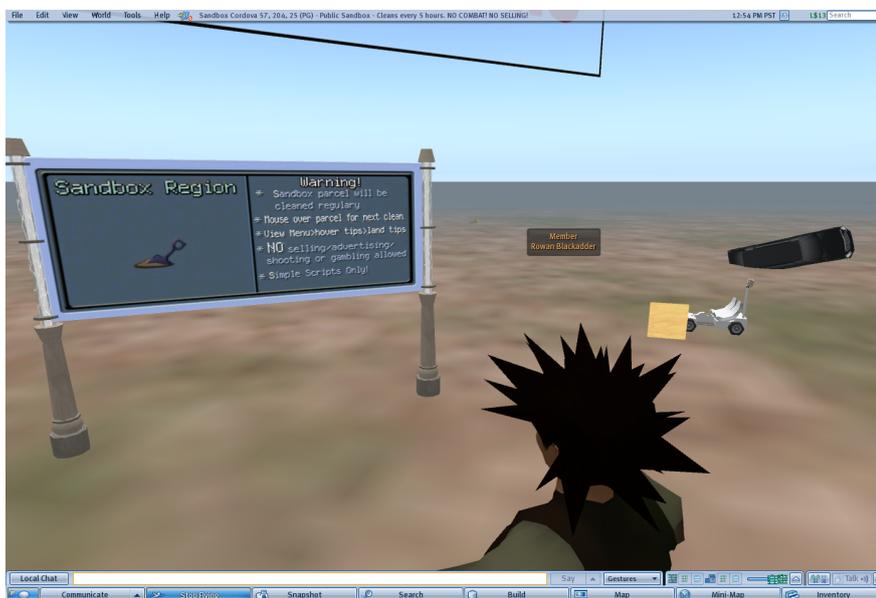


Abbildung 2.4: Die Second Life Sandbox



Abbildung 2.5: Bauen in Second Life

2.3 Second Life, Web 2.0 und MMORPGs

Die Frage ob Second Life ein Teil von Web 2.0 ist, wird von uns als eine grundlegende Frage angesehen. Zuerst werden die Gemeinsamkeiten und die Unterschiede zwischen Second Life⁸ und Web 2.0⁹ betrachtet. Hierbei soll deutlich werden, dass sich diese Technologien in einigen signifikanten Punkten unterscheiden, weshalb Second Life vorerst nicht als eine Web 2.0 Anwendung angesehen wird.

Danach werden die Gemeinsamkeiten und Unterschiede zwischen den MMORPGs

⁸vgl. Kapitel 2.2

⁹vgl. Kapitel 2.4

und Second Life aufgeführt. Hier versuchen wir, die Ähnlichkeiten zwischen diesen Anwendungen aufzuzeigen. Gleichzeitig wird hier auch eine klare Trennung eingeführt, damit am Ende sichtbar wird, dass es sich bei Second Life um eine eigenständige Anwendung handelt.

Abschließend versuchen wir, Second Life eindeutig einer Gruppe zuzuordnen.

2.3.1 Gemeinsamkeiten von Web 2.0 und Second Life

Zwischen Web 2.0-Anwendungen und Second Life existieren einige Gemeinsamkeiten. Die erste Gemeinsamkeit ist der Zugang über ein Netzwerk. Alle Web 2.0-Anwendungen und Second Life sind über das Internet zugänglich. Das Internet ist fast überall auf der Welt verfügbar. Die Ausnahme bilden schwer zugängliche und dünn besiedelte Orte, in denen keine Telefonverbindung existiert.¹⁰ Trotz dieser Einschränkung ist die Anzahl der Menschen mit einem Breitband-Zugang sehr groß. Mittlerweile haben 75 Prozent aller Haushalte in Deutschland einen Breitbandzugang zum Internet¹¹. Deswegen setzen Web 2.0-Anwendungen und Second Life auf das WWW.

Weiterhin werden sowohl die Inhalte im Web 2.0 sowie in Second Life von den Benutzern erstellt. Die Nutzer können selbst entscheiden wie der Inhalt aussieht. Diese Inhalte werden im Internet veröffentlicht und können von jeder interessierten Person eingesehen werden.

Second Life kann auch Inhalte von Web 2.0-Anwendungen in die virtuelle Welt einbinden. Beispiele dafür sind Flickr¹² oder Sloodle¹³. Hierbei wird auch deutlich, dass Second Life durch Web 2.0-Anwendungen erweitert wird. Obwohl die Gemeinsamkeiten zu dem Schluss führen könnten, dass es sich bei Second Life und Web 2.0 um eine ähnliche Technologie handelt, gibt es auch einige Unterschiede, die beachtet werden müssen.

2.3.2 Unterschiede zwischen Web 2.0 und Second Life

Zuerst werden wir die Eigenschaften aufzeigen, die in Second Life vorkommen aber nicht in Web 2.0-Anwendungen vertreten sind und umgekehrt. Danach wird

¹⁰vgl. http://www.telekom-presse.at/channel_internet/news_28808.html

¹¹vgl. <http://www.laptopkarten.de/breitband-internet/index.html>

¹²vgl. Kapitel 3.4.2.2

¹³vgl. Kapitel 3.4.1

das Gleiche mit Web 2.0-Anwendungen gemacht.

Second Life ist eine Anwendung, die eine dreidimensionale virtuelle Welt erzeugt. Der Benutzer bewegt sich als Avatar durch diese Welt, um die vorgestellten Inhalte zu sehen. Für die Übermittlung der Informationen, die ein Rechner braucht, um diese virtuelle Welt darzustellen, wird eine Breitband-Verbindung¹⁴ benötigt.¹⁵ Üblicherweise sind Web 2.0-Anwendungen in dieser Hinsicht genügsamer. Wenn einzelne Einschränkungen wie lange Ladezeiten außer acht gelassen werden, können alle Web 2.0-Anwendungen auch über eine Verbindung mit einem 56k-Modem¹⁶ oder eine ISDN-Verbindung¹⁷ angesprochen werden.

Auch wenn wir berücksichtigen, dass z.B. ein YouTube¹⁸ Video bei einer langsamen Verbindung lange braucht, um geladen zu werden, können wir es doch auf dem Computer ansehen. Hingegen benötigt Second Life viele Information in Echtzeit¹⁹. Deswegen kann Second Life ohne eine DSL²⁰- oder Kabelverbindung²¹ nicht gestartet werden.

Second Life basiert zudem auf der Skriptsprache LSL [Mel08]. Es ist nicht möglich, Inhalte durch eine andere Sprache zu erstellen. Die Einschränkungen in dieser Sprache sind absichtlich eingeführt worden, um die Sicherheit von Second Life zu erhöhen. Auf die Einschränkungen von LSL gehen wir im Kapitel 3.1.1 genauer ein. Dadurch ist das Erstellen eigener Funktionen in Second Life umständlich oder manchmal gar nicht möglich. Manche Features, wie z.B. das Exportieren von Bildern oder Videos aus Second Life, sind gar nicht möglich, weil LSL diese Funktionen unterbindet. Die Inhalte der Web 2.0-Anwendungen können von verschiedenen Programmen erstellt werden. Diese Programme müssen nicht zwangsläufig Web 2.0-Anwendungen sein. Als Beispiel können in diesem Fall die Web 2.0-Anwendungen YouTube und Flickr²² dienen. Die Videos können von einer Kamera oder einem ähnlichen Gerät aufgenommen, auf dem eigenen Rechner bearbeitet und auf YouTube zur Verfügung gestellt werden. Die Bilder bei Flickr,

¹⁴vgl. <http://www.zukunft-breitband.de/BBA/Navigation/hintergrund.html>

¹⁵vgl. <http://secondlife.com/support/sysreqs.php>

¹⁶vgl. <http://www.zeix.ch/de/lexikon/56k-modem/index.html>

¹⁷vgl. <http://www.itwissen.info/definition/lexikon/ISDN-Verbindung-ISDN-connection-type.html>

¹⁸vgl. <http://www.youtube.com>

¹⁹vgl. <http://www.itwissen.info/definition/lexikon/Echtzeit-RT-realtime.html>

²⁰vgl. Definition von DSL

²¹vgl. Link zur Kabelverbindung

²²vgl. Kapitel 3.4.2.2

können auch zuerst unabhängig von Flickr bearbeitet und dann ebenfalls hochgeladen werden. Second Life bietet keine andere Möglichkeit, die Objekte oder Skripte zu bearbeiten, weil der Compiler nur innerhalb der virtuellen Welt auf dem Second Life Server eingesetzt wird²³. Damit unterliegen Web 2.0-Programme im Gegensatz zu Second Life keiner Einschränkung.

Die Einschränkungen, auf die sich der Nutzer in Second Life einstellen muss, basieren nicht nur auf Sicherheitsaspekten. Manche Funktionen werden erst freigeschaltet, wenn der Benutzer einen Premium-Account²⁴ beantragt. Dieser Account erlaubt es dem Nutzer, Land zu kaufen und erweiterten Support zu benutzen. Der Premium-Kunde muss aber eine monatliche Gebühr entrichten.²⁵ eingesehen werden. Der Erwerb des Landes ist mit weiteren monatlichen Zahlungen verbunden.²⁶ Dieses Verhalten ist bei keinem der Web 2.0-Programme verzeichnet, weil es diese Unterscheidung und diese Preispolitik im Web 2.0 nicht gibt. Im Web 2.0 sind die Accounts zur Zeit nicht mit einer Gebühr verbunden. Diese Aussage könnte sich in Zukunft ändern, falls einige Web 2.0-Dienstleister ihre Dienste kostenpflichtig machen. Zur Zeit sind uns keine solchen Entscheidungen oder Pläne, die in diese Richtung führen könnten, bekannt.

Die Softwareanforderungen sind ein zusätzlicher Punkt, der eine wichtige Unterscheidung darstellt. Damit Second Life vom Nutzer betreten werden kann, wird eine spezielle Software benötigt. Diese Software kann nicht auf jedem Rechner installiert werden, da die Hardwareanforderungen hoch sind.²⁷ Web 2.0-Software arbeitet ausschließlich über den Browser oder einen Browser-Simulator.²⁸ Mittlerweile wird sogar ein Browser im Internet angeboten, der auf Web 2.0 Angebote ausgerichtet ist²⁹. Ein Browser ist in der heutigen Zeit in jedem Betriebssystem vorhanden. Unter Microsoft Windows³⁰ handelt es sich meistens um den Inter-

²³vgl. Kapitel 2.2

²⁴vgl. <http://secondlife.com/whatis/pricing.php>

²⁵http://static-secondlife-com.s3.amazonaws.com/corporate/Second_Life_Pricing_List_20081208.pdf

²⁶vgl. <http://secondlife.com/land/pricing.php>

²⁷vgl. <http://secondlife.com/support/sysreqs.php>

²⁸vgl. Snoopy - der Browser Simulator <http://sourceforge.net/projects/snoopy/>

²⁹vgl. Flock Browser <http://flock.com/>

³⁰vgl. Microsoft <http://www.microsoft.com/>

net Explorer³¹. GNU Linux³² wird mit dem Mozilla Firefox³³ ausgerüstet. Bei Mac OS Rechnern wird Safari³⁴ eingesetzt. Die Betriebssysteme können auch mit alternativen Browsern ausgestattet werden. Hier wurden nur die gängigen Konfigurationen genannt.

Die gängigen Browser bieten die Möglichkeit der Darstellung der Web 2.0-Inhalte. Manche Web 2.0-Anwendungen brauchen noch zusätzliche Erweiterungen, sog. Plug-ins wie Flash³⁵. Diese Plug-ins werden für alle gängigen Browser zur Verfügung gestellt. Jeder Rechner mit aktuellem Betriebssystem wird deswegen mit Web 2.0-Anwendungen funktionieren, was für Second Life nicht zutrifft.

Die Unterschiede zwischen Web 2.0 und Second Life sind nicht trivial und sollten nicht ignoriert werden. Im nächsten Unterkapitel werden die Gemeinsamkeiten zwischen den MMORPGs, die nachfolgend abgekürzt Online-Rollenspiele genannt werden, und Second Life erörtert. Als Referenz-Rollenspiel wird hierbei das zur Zeit bekannteste Rollenspiel „World of Warcraft“³⁶ von *Blizzard*³⁷ herangezogen.

2.3.3 Die Gemeinsamkeiten von MMORPGs und Second Life

Die Gemeinsamkeiten zwischen Online-Rollenspielen und Second Life werden sofort sichtbar, wenn die Programme gestartet werden. Beide beinhalten eine dreidimensionale virtuelle Welt, die auf einer Grafikengine³⁸ basiert.

Die Nutzer haben eine Spielfigur, die sie in der virtuellen Welt repräsentiert. Diese wird in Second Life Avatar³⁹ genannt; in den Online-Rollenspielen heißen sie meist Charaktere⁴⁰ genannt. Mit Hilfe von diesen Spielfiguren können sich die Nutzer in der Welt frei bewegen und sich mit anderen Nutzern austauschen. Die

³¹Internet Explorer <http://www.microsoft.com/windows/products/winfamily/ie/default.msp>

³²vgl. The Linux Homepage <http://www.linux.org/>

³³vgl. Mozilla <http://www.mozilla.com/>

³⁴vgl. Safari <http://www.apple.com/de/safari/>

³⁵Adobe Flash Player <http://www.adobe.com/de/products/flashplayer/>

³⁶World of Warcraft <http://www.worldofwarcraft.com/>

³⁷vgl. <http://www.blizzard.com/>

³⁸vgl. <http://www.dimension-seven.de/Grafik-Engine.php>

³⁹vgl. Kapitel 2.2.2

⁴⁰Charakter in „World of Warcraft“ <http://www.worldofwarcraft.com/info/basics/gettingstarted.html>

Unterhaltung wird meistens über den Textchat⁴¹ geführt, aber mittlerweile wird vermehrt auch der Voice-Chat⁴² benutzt.

Weiterhin basieren beide Anwendungen auf einem Client/Server Modell⁴³.

Die Software- und Hardwareanforderungen sind ähnlich. Beide Anwendungen benötigen Platz auf der lokalen Festplatte und ein aktuelles System, damit sie flüssig laufen können. Die Anwendung wird lokal gestartet und der Nutzer muss sich über diese Anwendung auf dem Server anmelden, um die Dienste nutzen zu können.

2.3.4 Die Unterschiede zwischen MMORPGs und Second Life

Die erste Unterscheidung, die dem Nutzer sofort auffällt, ist die fehlende Handlung in Second Life. Second Life besitzt keine Hintergrundgeschichte, die der Nutzer durchlaufen muss, um im Programm voranzukommen. Dadurch erhält der Avatar schon am Anfang alle Möglichkeiten, die sich ihm in Online-Rollenspielen erst langsam erschließen würden. Der Avatar muss keine Erfahrungspunkte⁴⁴ sammeln oder Missionen, auch Quests⁴⁵ genannt, bestehen, um auf spezielle Fähigkeiten zuzugreifen.⁴⁶

Die Inhalte, die Second Life bietet, sind von anderen Nutzern oder von Second Life-Administratoren erstellt worden⁴⁷. Die Inhalte in Online-Rollenspielen sind von Programmierern der Anwendung vorgegeben und können nicht erweitert werden. Eine Spielfigur in einem Rollenspiel kann keine Objekte selbst erstellen. Das Erstellen beschränkt sich meist auf das Brauen von Tränken, die im Programm benutzt werden.⁴⁸ Im Gegensatz dazu kann der Nutzer von Second Life komplexe

⁴¹vgl. Kapitel 3.1.3

⁴²vgl. http://www.mylanguageexchange.com/VoiceChat_gmm.asp

⁴³vgl. Kapitel 4.3

⁴⁴vgl. Erfahrungspunkte in World of Warcraft <http://wowsource.4players.de/erfahrungspunkte.php>

⁴⁵vgl. WoW Quests <http://quests.extreme-gaming.de/spiel,1072,world-of-warcraft-quests.html>

⁴⁶vgl. die Hintergrundgeschichte von World of Warcraft <http://wowsource.4players.de/geschichte.php>

⁴⁷vgl. Was ist Second Life? <http://de.secondlife.com/whatis>

⁴⁸vgl. World of Warcraft Alchimie http://wowsource.4players.de/tradeskills_alchemie.php

Objekte erstellen und diese einsetzen.⁴⁹ Abbilder von realen Gebäuden aus der realen Welt sind in Second Life oft anzutreffen. Als Beispiel sei der Hauptbahnhof von Frankfurt in Second Life genannt, wie in Bild 2.6 zu sehen ist.



Abbildung 2.6: Darstellung des Frankfurter Hauptbahnhofs in Second Life

Neben der Größe der Objekte können diese auch mit verschiedenen Funktionen versehen werden. Die Objekte können frei stehen oder an den Avatar angehängt werden. Dadurch kann der Avatar zusätzliche Freiheiten, wie z.B. das Fliegen in große Höhen, erlangen.

Eine weitere Möglichkeit, die in Online-Rollenspielen schon in den Allgemeinen Geschäftsbedingungen verboten wird, ist echtes Geld zu verdienen.⁵⁰ Der Nutzer kann für Objekte, Skripte und verschiedene Dienstleistungen, die er in Second Life anbietet, Geld verlangen.⁵¹ Dieses Feature ist auch in anderen Online-Rollenspielen gegeben, aber das virtuelle Geld in echte Währung umzuwandeln nicht. *Blizzard*⁵² droht sogar mit der Sperrung des Accounts, wenn ein Benutzer seinen Account für solche Tätigkeiten ausnutzt. Die Online Rollenspiele sind zum Spielen und als Freizeitspaß zu verstehen. Second Life dagegen versucht ein eigenes Wirtschaftssystem im Spiel aufzubauen, um eigenes virtuelles und echtes Geld zu vermehren.

⁴⁹Was ist Second Life? <http://de.secondlife.com/whatis>

⁵⁰vgl. <http://www.wow-europe.com/de/info/faq/antigoldselling.html>

⁵¹vgl. Was ist Second Life? <http://de.secondlife.com/whatis>

⁵²vgl. <http://www.blizzard.com/>

Second Life ist also keine Web 2.0 Anwendung und auch kein MMORPG. Zu beiden Gruppen gibt es zwar Berührungspunkte, aber die Unterschiede zwischen Second Life und diesen Gruppen überwiegen. Welche Gruppe beinhaltet dann Second Life? Im Folgenden wird diese Frage geklärt.

2.3.5 Second Life und MUVes

Obwohl es bisher so aussah, als ob Second Life eine Online-Anwendung außerhalb jeglicher Kategorie ist, wird in Kapitel 3.2 das Gegenteil bewiesen. Als Second Life ins Netz gestellt wurde, gab es nur wenige vergleichbare Programme. Mittlerweile gibt es einige Online-Anwendungen, die eine Ähnlichkeit mit Second Life aufweisen. In manchen Fällen übertreffen die Eigenschaften von diesen Anwendungen die von Second Life.

Als Beispiele für solche Umgebungen können Active Worlds⁵³, Croquet⁵⁴ und die mittlerweile wieder abgeschaltete Google Lively⁵⁵ genannt werden. Diese Umgebungen bieten nicht die gleichen Möglichkeiten zur Interaktion mit der dreidimensionalen Umgebung wie Second Life. Lively war z.B. eine virtuelle Umgebung, die in virtuelle Räume eingeteilt war. Diese Räume konnten beliebig gestaltet werden. Bei Croquet können zusätzlich zu den Objekten, virtuelle Welten gebaut werden. Croquet geht aber anders mit Web 2.0-Anwendungen um. Statt diese über Skripte aufzurufen, wird ein interner Browser in der virtuellen Umgebung gestartet.

Obwohl es einige Unterschiede zu Second Life gibt, ist das Prinzip in allen Umgebungen gleich. In keiner der Anwendungen werden dem Benutzer Vorgaben gemacht, wie er/sie sich zu verhalten hat. Eine Hintergrundgeschichte existiert in diesen Anwendungen nicht. Der Benutzer kann sich alles ansehen und eigene Inhalte in Form von Objekten, Skripten und Räumen erstellen. Alle Umgebungen haben eine eingebaute Chat-Funktion, die entweder nur als Text-Chat oder als Text- und Voicechat implementiert ist. Croquet und Lively sind Open-Source⁵⁶ Lösungen. Es existieren aber auch kommerzielle Lösungen, deren Code nicht freigegeben ist, die aber trotzdem mit Second Life vergleichbar sind. Eine dieser

⁵³vgl. Kapitel 3.2.1

⁵⁴vgl. Kapitel 3.2.2

⁵⁵vgl. Kapitel 3.2.4

⁵⁶vgl. <http://www.opensource.org/>

Lösungen ist Active Worlds⁵⁷. ActiveWorlds ist eine virtuelle Welt. Wie in Second Life auch, beinhaltet sie kostenlose und kostenpflichtige Accounts. Weiteres darüber kann im Kapitel 3.2.1 nachgelesen werden.

Wegen der Fülle der Übereinstimmungen bilden diese Anwendungen eine eigene Gruppe. Diese Gruppe ist unter dem Namen Multiuser Virtual Environments (Multiuser Virtual Environment (MUVE) ⁵⁸) bekannt.

2.3.6 Zusammenfassung

Die Unterschiede zwischen Web 2.0, MMORPGs und MUVES sind deutlich. Deswegen konnten wir belegen, dass sich ein MUVE von einem MMORPG unterscheidet. Die Frage, ob Second Life und Web 2.0 eine Einheit bilden, kann trotzdem nicht ohne weiteres beantwortet werden. Der Grund hierfür ist die Definition von Web 2.0. Web 2.0 beinhaltet die neueren Methoden wie soziale Netzwerke⁵⁹ und Wikis. Es gibt keine klare Definition, was ein Programm beinhalten muss, um als ein Web 2.0 Programm anerkannt zu werden. Im Sinne der Definition der sozialen Netzwerke ist auch Second Life ein soziales Netzwerk, weil es Menschen mit gleichen Interessen verbindet. Andererseits könnte diese Definition auch für die Spieler von Online-Rollenspielen angewandt werden.

Wir sehen aus diesem Grund Second Life als eine Weiterführung oder als eine mögliche Zukunft von Web 2.0 und vom WWW insgesamt und nicht als einen weiteren Teil von Web 2.0.

2.4 Web 2.0

“I define Web 2.0 as the design of systems that harness network effects to get better the more people use them, or more colloquially, as “harnessing collective intelligence.” This includes explicit network-enabled collaboration, to be sure, but

⁵⁷vgl. Kapitel 3.2.1

⁵⁸siehe auch: <http://www.cs.princeton.edu/~funk/ring.html>

⁵⁹Unter Web 2.0 hat sich ein Community-Netzwerk gebildet, das als soziales Netzwerk bezeichnet wird. Ein solches Netzwerk ist ein Beziehungsgeflecht aus Social-Network-Services, die von Teilnehmern mit gleichen Interessen genutzt werden und über die diese persönliche Daten austauschen und Beziehungen zueinander herstellen und vertiefen. <http://www.itwissen.info/definition/lexikon/Soziales-Netzwerk-social-network.html>

*it should encompass every way that people connected to a network create synergistic effects.*⁶⁰

Web 2.0 More Than Explicit Socialness

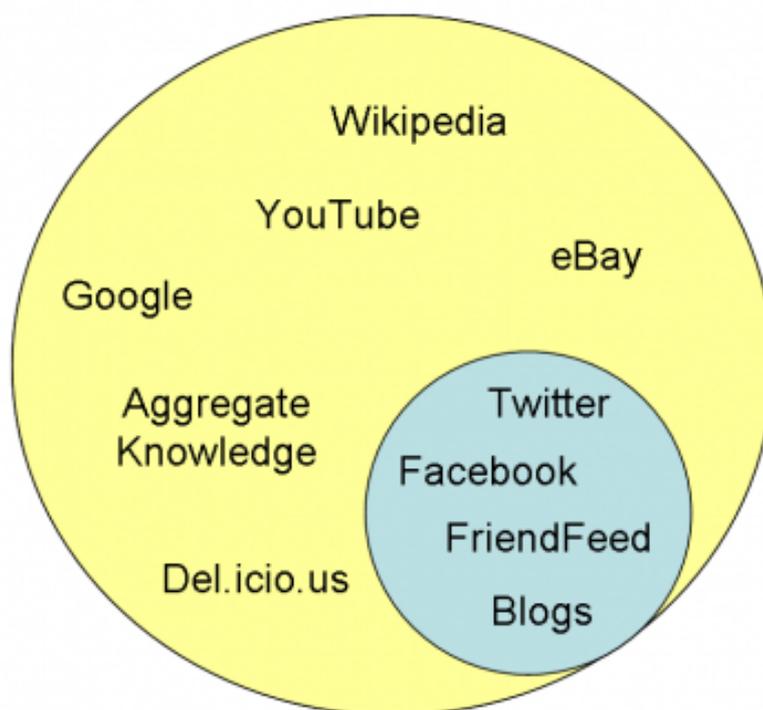


Abbildung 2.7: Web 2.0

Web 2.0 ist eine neue Technologie, die von Tim O'Reilly 2005 in einem Artikel ihren Namen bekam.⁶¹ Die Web 2.0-Programme sind keine monolithischen Programme, sondern eher eine Ansammlung von vielen relativ kleinen Anwendungen, die nach den Wünschen der Benutzer erschaffen und weiterentwickelt werden.⁶² Die Benutzer sind im Fall von Web 2.0 nicht Unternehmen, sondern eher private

⁶⁰Zitat: <http://radar.oreilly.com/2008/09/why-dell-dot-com-is-more-enterprise.html>

⁶¹vgl. <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>

⁶²vgl. http://koeln-bonn.business-on.de/die-sieben-paradigmen-des-web-2-0_id10673.html

Nutzer, die eine Webseite errichten und damit eigene Werke allen zur Verfügung stellen.

Diese Anwendungen sind einer ständigen Änderung unterzogen. Die Nutzer entscheiden bei einer Änderung, ob diese dauerhaft im Programm bleiben soll. Flickr⁶³ ist eine Web 2.0-Anwendung. Die neuen Features werden von den Benutzern bewertet. Im Folge dessen finden zwei von drei Features dauerhaften Einzug in die Anwendung. Weitere bekannte Anwendungen im Web 2.0 sind z.B. Delicious⁶⁴, YouTube⁶⁵ oder Wikipedia⁶⁶.

2.4.1 Entstehungsgeschichte

Um die Geschichte des Web 2.0 zu verstehen, muss man zuerst die Entstehung des WWW genau kennen. Bis 1996 bestand das Internet im Wesentlichen aus E-Mails und Datenkommunikation.⁶⁷ Nur versierte Benutzer konnten sich im Internet bewegen. Für alle anderen war die Bedienung zu komplex und nicht benutzerfreundlich.

Im Jahr 1996 entstand das Web 1.0. Die Erstellung eigener Webseiten wurde vereinfacht, und das Internet wurde für Unternehmen als Werbeplattform interessant. In dieser Zeit entstanden viele Webseiten, und das Internet wurde für die breite Masse zugänglich und interessant.

Bis 2001 entwickelte sich das WWW rasant. Im Jahr 2001 platzte die Internetblase. [Alb07] Vielen Benutzern wurde klar, dass das Internet überschätzt wurde. Die Kapitalgeber zogen daraufhin ihr Geld zurück.

Das World Wide Web änderte sich langsam. Diese Änderungen wurden dieses Mal durch die Benutzer initiiert, die weiterhin eigene Plattformen im WWW haben wollten. Dabei entwickelte sich eine neue Technologie, die 2004 zum ersten Mal von Tim O'Reilly als „Web 2.0“ bezeichnet wurde.⁶⁸

Als einer der ersten Teile der neuen Technologie entwickelte sich das Blog, das

⁶³vgl. Kapitel 3.4.2.2

⁶⁴vgl. Kapitel 3.4.2.1

⁶⁵vgl. <http://de.youtube.com/>

⁶⁶<http://wikipedia.org/>

⁶⁷http://koeln-bonn.business-on.de/die-geschichte-des-web-2-0_id10672.html

⁶⁸vgl. <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>

im nächsten Abschnitt behandelt wird.

2.4.2 Blog

Bei dem Begriff Blog handelt es sich um eine Abkürzung für den Begriff Weblog, der wiederum eine Zusammenführung der Begriffe „Web“ und „Log“ ist.[Alb07] Ein Blog ähnelt einem Tagebuch oder einem Journal. Es wird im World Wide Web veröffentlicht. Vor der Geburt des Begriffs „Blog“ führten zwar Autoren Journale im Internet, aber erst mit der Verfügbarkeit der Blogging-Software wurde es jedermann ermöglicht, ein solches Blog zu führen.

Eine interessante Definition des Blogs stammt von Robert Basic und wurde in seinem Blog veröffentlicht: *„Blogs sind keine Gemeinschaftswohnungen. Es gibt einen Hausbewohner, den Gäste je nach Bedarf besuchen, weil er möglicherweise leckeren Kaffee und Kuchen anzubieten hat. Auf Foren und Newsgroups machen die Hausbewohner gemeinsam die Musik, jeder kann dazu seinen Kuchen mitbringen. Party! Auf einem Blog macht nur einer die Musik, er bestimmt die Lautstärke, er backt den Kuchen.“*⁶⁹

Die Blogs beschäftigen sich mit verschiedenen Themen wie z.B.:

Watchblogs	Blogs, die Medien und Firmen kritisch beobachten.
Litblogs	Blogs, die sich mit der Literatur beschäftigen.
Corporate Blogs	Blogs von Firmen.
Blawgs	Blogs mit juristischen Themen.
Fotoblogs	Blogs, in denen vor allem Fotos veröffentlicht werden.

Tabelle 2.1: Die verschiedenen Blogarten.

Aber ein Blog hat mehr Funktionen zu bieten als ein einfaches Tagebuch. Eine zusätzliche Funktion ist, dass die Leser die Beiträge kommentieren können. Dadurch können sie den Autor auf mögliche Schwachstellen in seinen Ausführungen aufmerksam machen oder weitere Aspekte des Themas aufgreifen.

⁶⁹Ausschnitt aus dem Blog: Unterschied Forum, Chat,Newsgroup, Blog <http://www.basicthinking.de/blog/2005/12/21/unterschied-forum-chatnewsgroup-blog/>

Blogs sind keine neue Form von Newsgroups, weil es große Unterschiede zwischen Blogs und Newsgroups gibt. Einer der Unterschiede ist, dass nicht jeder beliebige Besucher eine neue Diskussion starten kann. Der Originalbeitrag ist immer der Ausgangspunkt jeder Diskussion, wobei hier nicht ausgeschlossen wird, dass die Diskussion so weit ausufern kann, dass sie keinen Bezug mehr zum eigentlichen Beitrag hat. Zudem bestimmt die Persönlichkeit des Bloggers den Stil des Blogs. Der Blogger ist die Person, die Beiträge in das Blog schreibt. Der Leser kann diese Person entweder anziehend oder abstoßend finden, oder der Blogger kann ihm gleichgültig lassen. Auf jeden Fall ist der Erfolg des Blogs von diesen Punkten abhängig.

2.4.3 Blogs in der Bildung

Blogs, die im Bildungsbereich eingesetzt werden, sind in den USA ziemlich weit verbreitet. In Deutschland haben sie sich bis heute noch nicht durchgesetzt. Sie bieten eine Menge Möglichkeiten für den Einsatz im Unterricht. Zum einen kann ein Blog als Kursportal dienen. Dann bietet das Blog den Kursteilnehmern die Möglichkeit, jederzeit auf die Kursmaterialien und Aufgaben zuzugreifen. Die Schüler oder Studierenden, die eine Unterrichtseinheit verpasst haben, müssen dadurch nicht auf die nächste Einheit warten, um auf den neuesten Stand zu kommen. Sie können sich sofort im Internet über den verpassten Stoff informieren.

Außerdem können Blogs für die gemeinsame Arbeit verwendet werden, wie z.B. zum Lösen von Gruppenaufgaben. Bei dem Gemeinschaftsprojekt können dadurch auch Experten von außerhalb mit Studierenden zusammenarbeiten.

Die dritte Möglichkeit, die sich bei der Benutzung der Blogs eröffnet, ist das Lesen von Experten-Blogs. Die Studierenden werden dadurch immer auf dem Laufenden gehalten und haben Zugang zu neuesten Informationen. Außerdem können sie lernen, unwichtige von wichtigen Informationen zu trennen und zu verarbeiten.

Mittlerweile ist auch eine Gemeinde entstanden, die mit Hilfe der Web 2.0-Werkzeuge eine Enzyklopädie im Internet zu Verfügung stellt. Diese Enzyklopädie

heißt Wikipedia⁷⁰. Sie basiert auf einer Web 2.0-Anwendung, die MediaWiki⁷¹ genannt wird. MediaWiki gehört zu der Gruppe der Wikis. Im nächsten Abschnitt gehen wir genauer auf diese Technologie ein.

2.5 Wiki

Ein Wiki ist eine Software und eine Sammlung von Webseiten, die von den Benutzern gelesen und auch verändert werden können. Dadurch entstehen Werke, die gemeinschaftlich von verschiedenen Autoren verfasst werden. Ziel eines Wikis ist es im Allgemeinen, die Erfahrung und den Wissensschatz der Autoren kollaborativ in Texten auszudrücken. Deswegen werden Wikis im Internet oder auch auf privaten Rechnern eingesetzt. Das Wiki übernimmt dabei die Rolle eines Content Management Systems⁷².

Ein Content Management System ist ein Anwendungsprogramm, das eine gemeinschaftliche Erstellung von Dokumenten ermöglicht und organisiert. Dabei ist es unerheblich, ob es sich bei den Programmen um Text- oder Multimedia-Dokumente handelt. Ein Autor kann ein solches System auch ohne Programmierkenntnisse bedienen. Das ist gleichzeitig auch die Stärke von Wikis, weil die Erstellung eines Dokuments im Wiki üblicherweise intuitiv über den Browser geschieht. Ein Autor kann in kürzester Zeit erhebliche Fortschritte im Umgang mit dem Wiki erzielen, und zwar ohne über HyperText Markup Language (HTML)-, Hypertext Preprocessor (urspr.: Personal Home Page) (PHP)- oder ähnliche Programmierkenntnisse zu verfügen, die bei einem Webauftritt notwendig sind.

Ein Unterschied zum Content Management System (CMS) ist der offene Zugriff. Während im CMS alle Arbeitsabläufe genau reguliert werden, wird das Wiki durch die Benutzer, die auch die Informationen bereitstellen, selbst kontrolliert. Das Wiki hat mehrere Klone⁷³. Diese sind meistens freie Software, d.h. unter anderem liegt ihr Quellcode jedermann offen. Er kann verändert und erweitert werden. Die Software ist häufig modular aufgebaut. Die Modularisierung erlaubt es, über die Schnittstellen das Programm zu erweitern, ohne den Inhalt der zu erweiternden Software zu kennen.

⁷⁰vgl. <http://wikipedia.org/>

⁷¹2.6

⁷²<http://de.wikipedia.org/wiki/Content-Management-System>

⁷³Erläuterungen im Kapitel 2.6

Der Name Wiki, der im hawaiianischen „schnell“ bedeutet, will in diesem Zusammenhang auf das schnelle und unkomplizierte Hinzufügen von Inhalten zu einem Thema hinweisen. Die Idee, eine Seite direkt im Browser zu verändern, ist nicht neu. Diese Idee hatte schon Tim Berners-Lee, der Erfinder des WWW. Berners-Lee war schon zu Beginn der Ansicht, dass ein Web-Browser eine Kombination aus Viewer und Editor sein sollte. Leider wurde sie im WWW bis zum Erscheinen der Wikis nicht zufriedenstellend realisiert.⁷⁴

2.5.1 Entstehungsgeschichte

Das erste Wiki wurde vom amerikanischen Softwareentwickler Ward Cunningham 1994 entwickelt und 1995 über das Internet zur Verfügung gestellt. Das Ur-Wiki „Portland Pattern Repository“ wurde in Perl geschrieben.⁷⁵ Es wurde ursprünglich als Wissenssammlung und Forum über Entwurfsmuster ⁷⁶ genutzt. Inzwischen sind noch weitere Wiki Engines entwickelt worden, die von Entwicklern für die eigenen Bedürfnisse angepasst wurden. Diese Wikis sind nicht mehr auf Perl⁷⁷ beschränkt, sondern es existieren Implementierungen in verschiedenen Sprachen, wie z.B. JavaScript⁷⁸, Java⁷⁹ oder PHP. Die bekanntesten Wikis, die eine große Verbreitung gefunden haben, sind MediaWiki ⁸⁰ und MoinMoinWiki⁸¹. Zu der Verbreitung der Wikis hat am meisten Wikipedia, die freie Online-Enzyklopädie, beigetragen. Zur Zeit registriert allein die deutsche Hauptseite von Wikipedia.org mehr als eine halbe Million Besucher pro Tag⁸².

Die Seiten eines Wikis können von jedem Nutzer, der einen Browser besitzt, angelegt und bearbeitet werden. Die Software läuft vollständig auf dem Server, wodurch keine Installation einer Zusatzsoftware notwendig ist. Das Editieren von

⁷⁴vgl. Tim Berners-Lee and HTML <http://inventors.about.com/od/computersoftware/a/html.htm>

⁷⁵<http://c2.com/>

⁷⁶„Ein Entwurfsmuster beschreibt ein spezielles Entwurfsproblem, das in bestimmten Entwurfskontexten immer wieder auftaucht, und liefert ein bewährtes generisches Schema für dessen Lösung.“[Dro05]

⁷⁷<http://www.perl.org/about.html>

⁷⁸<http://de.selfhtml.org/intro/technologien/javascript.htm>

⁷⁹<http://java.com/de/about/>

⁸⁰vgl. Kapitel 2.6

⁸¹vgl. Kapitel 2.6

⁸²<http://stats.grok.se/de/200901/Hauptseite>

Wikis ist sehr einfach gehalten, da es sich auf das Schreiben des Artikels und das Hochladen von Bildern beschränkt. Es sind keine HTML-Kenntnisse erforderlich. Die Seiten werden nicht nur von einem Benutzer editiert, sondern mehrere Nutzer können den Artikel bearbeiten und gefundene Fehler verbessern.

2.5.2 Lizenzen

In einem Wiki kann jeder Nutzer Inhalte hinzufügen. Es ist wichtig, darauf zu achten, dass keine Urheberrechte verletzt werden. Materialien dürfen nicht ohne das Wissen oder die Angabe des Autors übernommen werden. Viele Wiki-Klone verwenden die „GNU General Public License (GPL)“⁸³. Die GNU GPL beinhaltet, dass jeder das Programm für jeden Zweck nutzen kann. Es wird aber ausdrücklich darauf hingewiesen, dass es sich um freie Software handelt, deren Quellcode offengelegt werden muss. Jeder Nutzer darf das Programm an seine individuellen Bedürfnisse anpassen und weiter verbreiten. Diese Programme müssen jedoch auch wieder unter die GNU GPL gestellt werden. Für diese Programme gilt die gleiche Lizenz wie für das Ursprungsprogramm. Dies ist das sogenannte Copyleft Prinzip⁸⁴.

2.5.3 Syntax

Die Syntax in Wikis ist je nach verwendeter Software unterschiedlich. Bis zum heutigen Zeitpunkt wurde kein Standard erstellt, der die Migration von einem Wiki-System zum nächsten vereinfachen würde. Alle Dialekte haben aber die Eigenschaft, dass sie einfacher aufgebaut sind als HTML, die meistgenutzte Sprache des WWW. Diese Beschränkung der Wiki-Syntax auf das Wesentliche hat es ihr ermöglicht, von vielen Menschen verstanden und benutzt zu werden.

Die Wikis sind aber auch miteinander verwandt. Es gibt einige Charakteristika, die in allen Wikis vorkommen. Dies sind der Bearbeitungsmodus, die interne Verlinkung, die Speicherung der Versionen und die Suchfunktion.

Im Bearbeitungsmodus kann ein Autor einen Artikel neu anlegen oder verändern. Der eingegebene Text ist in den meisten Fällen kein HTML-Code, sondern eine Wiki-Syntax⁸⁵. Diese Vereinfachung wurde eingeführt, weil die Autoren sich nicht

⁸³<http://www.gnu.org/licenses/gpl.html>

⁸⁴<http://www.gnu.org/copyleft/>

⁸⁵<http://de.wikipedia.org/wiki/Hilfe:Textgestaltung>

auf die Syntax des HTML-Quellcodes, sondern vor allem auf den Inhalt konzentrieren sollen. Es ist wichtig anzumerken, dass immer der Autor als der Verfasser der Änderungen mitprotokolliert wird. Dies soll Vandalen abschrecken, also verhindern, dass Nutzer Informationen absichtlich verfälschen. Die Speicherung der Versionen dokumentiert alle vorangegangenen Versionen einer Seite. Dadurch ist gewährleistet, dass eine gezielte Schädigung des Inhalts rückgängig gemacht werden kann.

Vandalismus ist eine gezielte Zerstörung von Seiten. Darunter fallen die Löschung des Inhalts einer Seite, das Löschen von Textabschnitten und das Verfälschen von Informationen.

Viele Wikis bieten eine Differenzfunktion⁸⁶, die es dem Autor erlaubt, die Änderungen, die zwischen den Versionen erfolgt sind, zu überblicken. Durch die optischen Hervorhebungen sieht der Benutzer, welche Teile des Inhalts korrigiert, gelöscht oder ergänzt wurden[Rah07]. Zudem sollte jeder Autor, der einen Artikel korrigiert oder ergänzt, einen kurzen Kommentar abgeben, weshalb er diese Ergänzungen oder Änderungen vorgenommen hat.

Die Suchfunktion ist eine Funktion, die mittlerweile sehr wichtig geworden ist. Durch die große Anzahl der Seiten wäre eine Suche nach einem Artikel mühsam, wenn nicht sogar unmöglich. Über die Suchfunktion lässt sich ein Beitrag direkt über den Titel oder mittels der Volltextsuche finden. Neben der integrierten Suchfunktion kann ein Beitrag auch über Suchmaschinen im Internet gefunden werden.[wik09a]

2.6 Wiki-Klone

Es gibt mehr als 100 verschiedene Wiki-Klone⁸⁷. Die Unterschiede zwischen diesen sind erheblich, weil sie von Entwicklern implementiert wurden, die sie dem Anwendungsgebiet angepasst haben. Dadurch unterscheiden sie sich in der Funktionalität, dem Schwierigkeitsgrad der Installation und sogar der Syntax teilweise beträchtlich voneinander. Obwohl sich die Wiki-Systeme stark voneinander unterscheiden, gibt es Funktionen, die bei jedem Wiki vorhanden sein müssen. Diese Funktionen wurden im vorherigen Kapitel ausführlich beschrieben. Die bekann-

⁸⁶<http://www.wikiservice.at/gruender/wiki.cgi?DifferenzFunktion>

⁸⁷<http://www.wikimatrix.org/>

testen Wiki-Klone sind MediaWiki⁸⁸ und MoinMoin⁸⁹, die in diesem Kapitel vorgestellt werden.

Die einzelnen Wikis werden im Internet in einer Vergleichstabelle aufgelistet⁹⁰. In der Auflistung erhält man Informationen über die Lizenz, die verwendete Programmiersprache, die Art der Datenspeicherung, die Sicherheitsvorkehrungen, die Anti-Spam Mechanismen, die Syntax und die verschiedenen Ausgabeformate.

MediaWiki

MediaWiki ist eine Software zum Betrieb eines Wikis, die ursprünglich für die freie Enzyklopädie Wikipedia entwickelt wurde. Mittlerweile wird sie auch für verschiedene andere Projekte der gemeinnützigen Wikimedia-Stiftung⁹¹, wie Wiktionary⁹², Wikibooks⁹³ und Wikinews⁹⁴, benutzt. Sie ist unter der GNU GPL lizenziert und in der Skriptsprache PHP geschrieben. Zum Speichern der Inhalte nutzt MediaWiki die relationale Datenbank MySQL⁹⁵. Alternativ kann ab Version 1.5 auch PostgreSQL⁹⁶ als Datenbank-Backend verwendet werden, jedoch ist die Unterstützung noch experimentell.[Rah07]

Entstehungsgeschichte

MediaWiki entstand aus einer Wiki-Engine, die der deutsche Biologe Magnus Manske für die Online-Enzyklopädie Wikipedia entwickelte, nachdem sich gezeigt hat, dass die zuvor eingesetzte UseModWiki-Engine den Anforderungen nicht gewachsen war. [Rah07] Am 25. Januar 2002 wurde die erste Version, damals Phase II genannt, erstmals eingesetzt. Nach einer hauptsächlich durch Lee Daniel Crocker geschriebenen Neufassung wurde im Juni 2002 eine verbesserte Version der offiziell immer noch namenlosen Software auf dem Wikipedia-Server installiert. In den Folgejahren entwickelte sich MediaWiki zu einem erfolgreichen Open-Source-Projekt, an dem im Jahr 2005 über 60 Programmierer und Helfer beteiligt waren.

⁸⁸vgl. Kapitel 2.6

⁸⁹vgl. Kapitel 2.6

⁹⁰<http://moinmoin.wikiwikiweb.de/WikiEngineComparison>

⁹¹<http://www.wikimedia.org/>

⁹²<http://de.wiktionary.org/wiki/Wiktionary:Hauptseite>

⁹³<http://de.wikibooks.org/wiki/Hauptseite>

⁹⁴<http://de.wikinews.org/wiki/Hauptseite>

⁹⁵<http://www.mysql.de/>

⁹⁶<http://www.postgresql.org/>

Neben Wikipedia und ihren Wikimedia-Schwesterprojekten setzen heute zahlreiche Organisationen, Firmen und Institutionen MediaWiki ein. Seit der Version 1.7 wird PHP 4 nicht mehr unterstützt, es ist ein PHP 5-Interpreter⁹⁷ notwendig.

MoinMoin

MoinMoin [moi09] ist ein in der Skriptsprache Python⁹⁸ geschriebener Wiki-Klon. MoinMoin wird unter der GNU GPL-Lizenz verbreitet. Die erste Version erschien 2000. Der Name „MoinMoin“ ist eine Anspielung auf den norddeutschen Gruß und auf die Doppelung der „CamelCase“⁹⁹-Schreibweise von „WikiWiki“, einem Synonym von Wiki. MoinMoin benutzt keine Datenbank, wie MediaWiki, sondern schreibt die Artikel in Dateien. MoinMoin verwendet den Python-Interpreter, der auf einigen Webservern nicht installiert ist. Diese Tatsache bremst die Verbreitung dieser ansonsten sehr einfachen Wiki-Implementierung.

JSPWiki

JSPWiki ist eine in der Programmiersprache Java geschriebene Wiki-Software, die auf Java Server Pages basiert.¹⁰⁰ Sie wurde 2001 von dem Finnen Janne Jalkanen entwickelt und steht auch unter der GNU GPL-Lizenz. Es wird vor allem von Firmen im Intranet eingesetzt und besitzt eine aktive Entwicklergemeinde.

DokuWiki

DokuWiki ist eine einfache Wiki-Software, die in PHP implementiert wurde. Mit DokuWiki kann der Benutzer jede Art von Dokument erstellen. DokuWiki benutzt keine Datenbank, sondern die Daten werden in eine Datei geschrieben. Die Syntax der Software ist einfach gehalten, ist aber trotzdem noch mächtig genug, um beliebige Dokumente zu erstellen.¹⁰¹

⁹⁷<http://www.e-teaching.org/glossar/interpreter>

⁹⁸<http://python.org/>

⁹⁹<http://dotnet.mvps.org/dotnet/articles/camelcase/>

¹⁰⁰<http://www.jspwiki.org/>

¹⁰¹<http://www.dokuwiki.org/dokuwiki>

PHPWiki

PHPWiki ist ein Klon von WikiWikiWeb, der ersten Wiki-Implementierung. Es ist die erste Wiki-Software, die in PHP geschrieben wurde. PHPWiki wurde 1999 veröffentlicht und unterstützt die Mehrzahl der Datenbanken. Die Installation von PHPWiki ist einfach und ist schon nach der Installation, ohne weitere Konfiguration, bereit für die Benutzung. Beim PHPWiki können auch Plug-ins eingesetzt werden, die die Funktionalität von PHPWiki erweitern.¹⁰²

PmWiki

PmWiki gehört zu den Wiki-Systemen, die keine Datenbank benötigen. Das Wiki ist sehr einfach zu installieren. Es unterstützt ein Template-System¹⁰³, mit dem es möglich ist, das Aussehen einer Seite sowie ihre Funktionalität zu verändern. Das System hat auch einige Sicherheitsfeatures, wie z. B. die Rechtevergabe für einzelne Seiten. Wie bei MediaWiki können bei PmWiki die Rechte individuell für die Seiten angepasst werden, damit nicht unkontrolliert wichtige Seiten verändert werden. Wie PHPWiki hat auch dieses Wiki die Möglichkeit, seine Funktionalität durch Plug-ins zu erweitern.¹⁰⁴

¹⁰²<http://phpwiki.sourceforge.net/>

¹⁰³<http://www.pmwiki.org/wiki/Cookbook/Templates>

¹⁰⁴<http://www.pmwiki.org/>

Kapitel 3

Analyse

Um die Anforderungen an das zu entwickelnde System definieren zu können, findet zunächst eine ausführliche Analyse der Ausgangssituation statt. Das Ziel ist es, die Konkurrenzprodukte kurz vorzustellen, die es zur Zeit gibt. Second Life wurde schon in Kapitel 2 vorgestellt. Deswegen wird hier darauf verzichtet. Vielmehr konzentriert sich den Abschnitt 3.1 auf die Sprache in Second Life ¹ und auf die verschiedenen Kommunikationsmöglichkeiten, die Second Life für ein Skript bietet ². Außerdem werden die Möglichkeiten der Web 2.0-Anwendung MediaWiki genauer erläutert. Es wird genau untersucht, wie MediaWiki Seiten aufbaut und welche Formatierung benötigt wird.

Die Konkurrenz muss hierbei in zwei Bereiche unterteilt werden. Zum einen sind hier die Konkurrenten von Second Life zu betrachten ³. Dabei werden die drei virtuellen Umgebungen vorgestellt, die zur Zeit der Erstellung dieser Arbeit entstanden sind ⁴ und die beiden ältesten virtuellen Umgebungen, die eine große Gemeinde haben.

Den zweiten Bereich stellen die Anwendungen dar, die eine Verbindung von Second Life zu externen Web 2.0-Anwendungen aufbauen. In Abschnitt 3.4 wird eine klare Abgrenzung zwischen der prototypischen Umsetzung und diesen Anwendungen aufgezeigt.

Am Ende des Kapitels werden die Anforderungen an den Prototyp spezifiziert ⁵.

¹vgl. Abschnitt 3.1.1

²vgl. Abschnitt 3.1.3

³vgl. Abschnitt 3.2

⁴vgl. Abschnitt 3.2.4

⁵vgl. Kapitel 3.5

3.1 Second Life

3.1.1 Linden Scripting Language

Linden Scripting Language (LSL) ist eine Programmiersprache, die von Second Life-Bewohnern benutzt wird. LSL kann das Verhalten von Objekten steuern.

Die Syntax von LSL ähnelt der von der Programmiersprache C⁶. LSL erlaubt die Interaktion mit der Second Life-Welt und dem Internet⁷. Ein LSL-Skript besteht aus Variablen, Funktionen, Definitionen und einem oder mehreren benannten Zuständen oder States. Jedes State definiert die Reaktion des Programms, wenn sich das Programm in dem State befindet. Das State, welches am Anfang definiert ist, ist das State Entry. Das State Entry wird immer dann ausgeführt, wenn das Skript zum ersten Mal gestartet wird.[Mel08] Das System sendet Ereignisse auch bezeichnet als Events, wie Timer, Bewegung, Chat, E-Mail und Kollisionen im virtuellen Raum an das Skript, die das Skript anschließend auswertet.

Skripte können das Objekt verändern und mit anderen Objekten und Bewohnern kommunizieren. Das Skript selbst ist gebunden an ein Objekt oder Primitiv. Ein Objekt kann ein Stuhl, Tisch oder ein anderer Gegenstand in der virtuellen Welt sein. Ein Primitiv ist die kleinste Einheit, wie z.B. eine Kugel oder ein Würfel, aus der sich ein Objekt zusammensetzen kann. Besteht ein Objekt nur aus einem Primitiv, dann wird das Primitiv automatisch zum Objekt.

Mittlerweile gibt es mehr als 300 Funktionen in LSL, auf die ein Benutzer zugreifen kann. Die Benutzer können aber auch eigene Funktionen definieren.

Die Skriptsprache LSL ist stark typisiert⁸. Der Quellcode wird zuerst in einen Bytecode kompiliert und danach auf einem der Linden Lab-Server ausgeführt.

Die zulässigen Datenstrukturen in LSL sind Integer, Float, String, (dreidimensionale)Vektoren und Rotationen. Außerdem werden heterogene Listen [GF05] erlaubt. Es existiert jedoch keine Möglichkeit, die Daten dauerhaft zu speichern.

⁶vgl. <http://home.fhtw-berlin.de/~junghans/cref/SYNTAX/syntax.html>

⁷siehe Kapitel 3.1.3

⁸Bei einer Typisierung bleibt eine einmal durchgeführte Bindung zwischen der Variable und dem Datentyp bestehen. Der entscheidende Vorteil der starken Typisierung ist, dass schon beim Übersetzen des Quellcodes eine Überprüfung der Zulässigkeit von Operationen durchgeführt wird. Nach dem erfolgreichen Kompilieren ist bereits gewährleistet, dass Fehler, die auf der unzulässigen Typkonvertierung beruhen, ausgeschlossen sind vgl. [http://msdn.microsoft.com/en-us/library/aa378693\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa378693(VS.85).aspx)

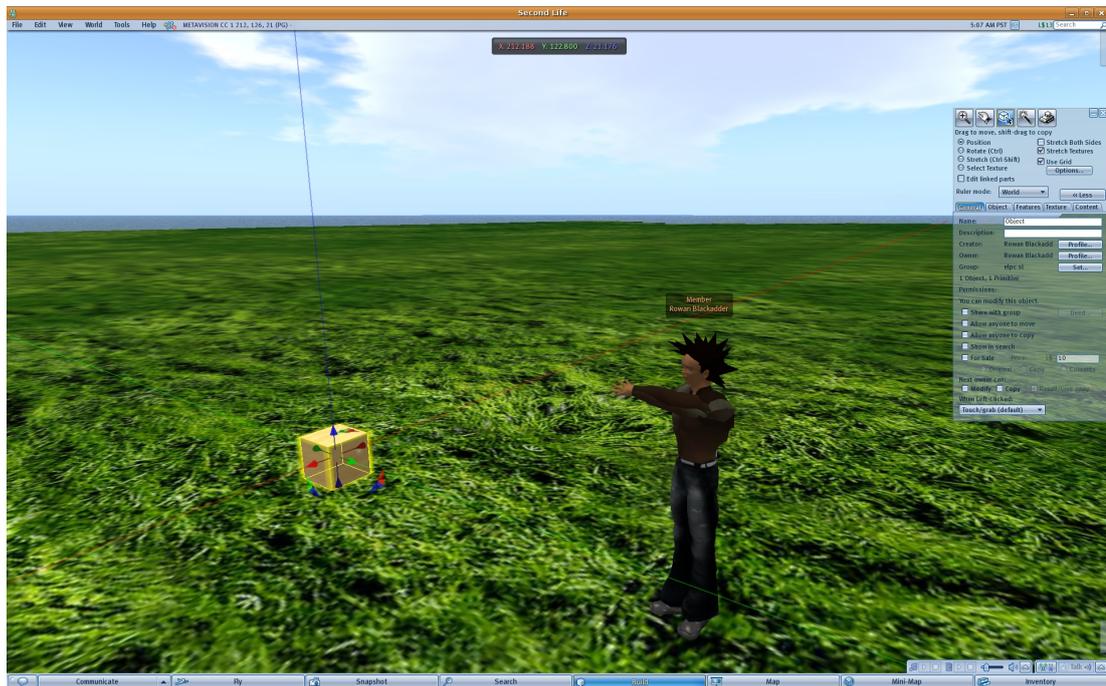


Abbildung 3.1: Erstellung eines Primitiv in Second Life

Speicherstrukturen wie Dateien oder Datenbanken sind in LSL nicht vorgesehen. Die LSL-Skripte arbeiten auch dann weiter, wenn sich der Benutzer abgemeldet hat, da diese Skripte an ein Objekt und nicht an den Benutzer gebunden sind. Die Objekte existieren auch dann, wenn der Benutzer nicht mehr anwesend ist. Eine Ausnahme bilden die Skripte in den Objekten, die an den Avatar angehängt sind. Diese Objekte werden nach dem Abmelden wieder im Inventar abgelegt. Einige LSL-Funktionen haben zusätzlich eingebaute Verzögerungen⁹. Diese Verzögerungen können von 0,2 bis zu 20 Sekunden dauern. Die Verzögerungen garantieren, dass das Skript das System nicht überlastet. Neben der Verzögerung haben die Skripte eine Einschränkung bezüglich der Größe. Die Größe des Skripts kann höchstens 16 kB betragen. Durch die Dimension wird auch die Funktion des Skripts eingeschränkt, weil der Benutzer keine komplexen Programme in einem Skript unterbringen kann. Als nächstes wird die Verbindung von Second Life und einem Learning Management System (LMS) betrachtet. Dabei werden die Vorteile dieser Fusion deutlich.

⁹siehe Abschnitt 3.1.3

3.1.2 Second Life und LMS

Second Life hat, im Gegensatz zu anderen Massive Multiplayer Online Role Playing Games (kurz MMORPG), keine Hintergrundgeschichte, an der sich der Benutzer orientieren muss. Dozenten können frei ihre Vorträge halten, ohne sich an einen Handlungsverlauf halten zu müssen. Außerdem können sie ihre eigenen Klassenräume erstellen, die alles beinhalten, was sie für ihre Lehrtätigkeit benötigen. Dadurch sind mittlerweile verschiedene Inseln entstanden, die sich speziell nach Bedürfnissen der Lehrtätigkeit richten. Als Beispiel wird hier „Democracy Island“ von der New York Law School genannt. [kem08]

Zudem sind die Werkzeuge zur Erstellung der Räume und verschiedener Geräte sehr einfach gehalten. Dadurch ergibt sich eine Fülle von Möglichkeiten der Darstellung der Lehrtätigkeit. Die Klassenräume können speziell an die Bedürfnisse der Dozenten und Lehrer angepasst werden; die Schüler und Studierenden erhalten einen praktischen Einblick in den Lernstoff. Dadurch wird die ansonsten trockene Theorie interessanter und anschaulicher.

Ein großes Problem in Second Life ist die Speicherung der Daten. Die Daten können nur in Form von Notecards gespeichert werden¹⁰. Diese Notecards werden in einer Standardschrift gehalten, die nicht verändert werden kann. Dadurch ist es auch nicht möglich, komplexe Daten, wie z.B. Dokumente mit Bildern, in Notecards zu speichern. Diese Daten können zwar zentral in einem Primitiv¹¹ gespeichert werden, von wo sie jedermann zugänglich sind, aber diese Möglichkeit wäre umständlich und bei vielen Daten auch unübersichtlich.

LMS-Systeme[kem08] haben mit der Speicherung der Daten keine Probleme. Zur Zeit werden sie sogar hauptsächlich als sog. Repositories¹² benutzt.

Die Nachteile der LMS liegen in der Darstellung der Daten. Die Möglichkeiten der Darstellung im zweidimensionalen Raum sind stark eingeschränkt. Diese Möglichkeiten beschränken sich auf die reine Theorie, die von Studierenden schnell wieder vergessen wird.

Hingegen können die Daten in Second Life auf vielfältige Art und Weise dargestellt werden. Zudem ist es möglich, diese Daten zu visualisieren und auch komplexe Materie, wie Beweise, können anschaulich dargestellt werden. Eine sol-

¹⁰siehe auch Kapitel 3.1.3

¹¹siehe auch Kapitel 2.2.3

¹²vgl. <http://www.e-teaching.org/glossar/repository>

che Vorgehensweise trägt maßgeblich zum besseren Verständnis bei [kem08].

Durch diese unterschiedlichen Stärken bietet es sich an, die beiden Plattformen zu vereinen. Mit der Vereinigung von LMS und Second Life wird eine virtuelle Oberfläche geschaffen, die viele Benutzer aufnehmen kann und zugleich einen Repository hat, in dem alle Informationen übersichtlich und strukturiert gesammelt werden können. Die Studierenden haben mehr Spaß an der Materie, weil sie in der virtuellen Umgebung gesehen und ausprobiert werden kann. Außerdem können sie jederzeit aus Second Life auf die verschiedenen Materialien zugreifen und nicht Verstandenes nachlesen.

3.1.3 Kommunikation und Interaktion

Die Kommunikation ist bei einem Server, der sich mit seinen Benutzern nur über das Internet austauschen kann, sehr wichtig. Es existieren verschiedene Möglichkeiten wie das LSL Skript mit der virtuellen Welt und mit der Netzumgebung kommunizieren kann. Hier werden einige Funktionen genannt, die auch später in das LSL Skript dieser Arbeit einfließen werden. Einige Funktionen haben eingebaute Beschränkungen. Die Beschränkungen und ihre Gründe werden bei jedem Skript kurz erläutert.

Textchat Mit Hilfe des Textchats kann man alle umstehenden Avatare im Second Life erreichen. Der Textchat ist in mehrere Kanäle unterteilt. Der Hauptchat, der von allen Avataren in der Umgebung gesehen werden kann, wird auf Kanal 0 gesendet. Die Nachrichten, die auf Kanal 0 gesendet werden, erscheinen kurz in der unteren linken Seite und verschwinden nach wenigen Sekunden. Der ganze Chat kann aber im Chatfenster eingesehen werden.

Die anderen Kanäle können von den Avataren genutzt werden, wenn die geschriebene Mitteilung nicht von umstehenden Avataren gesehen werden soll. Diese Mitteilung kann nur an ein Skript gesendet werden, weil die Avatare keine Möglichkeit haben, die anderen Kanäle zu überwachen.

Instant Message Instant Messages sind eine interessante Möglichkeit, private Unterhaltungen zu führen.[Mel08] Eine Instant Message kann sowohl von einem Benutzer, ala auch und von einem Objekt verschickt werden. Der Unterschied da-

bei ist, dass der Benutzer eine Mitteilung auch empfangen kann. Ein Objekt kann das nicht. Die maximale Länge einer Nachricht ist auf 1023 Zeichen beschränkt. Wenn zu viele Instant Messages auf einmal verschickt werden, kann das zu einem Problem führen. Dieses Problem nennt sich Instant-Messaging-Spam¹³. Um diesem Engpass entgegenzuwirken, wird das Skript um zwei Sekunden verzögert.

Dialogbox Dialogboxen leiten den Benutzer durch verschiedene Entscheidungsmöglichkeiten. Sie bieten verschiedene Buttons an, über die der Benutzer seine Entscheidungen dem System übergeben kann.

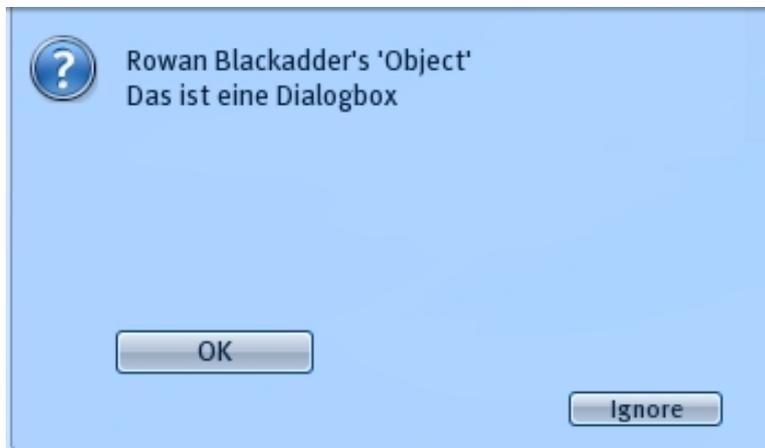


Abbildung 3.2: Eine typische Dialogbox in Second Life

E-Mails E-Mails waren am Anfang der einzige Weg der Kommunikation zwischen Second Life und dem Internet. Ein Objekt kann hier, im Gegensatz zu Instant Messages, sowohl senden, als auch empfangen. Die Nachrichten sind zur Zeit auf 4096 Zeichen beschränkt. Zu diesen 4096 Zeichen gehören aber auch die Informationen über den Absender der E-Mail. Das sind u.a. zählen hier: der Name des Objekts, der Name der Region, in der sich das Objekt befindet, die lokalen Koordinaten und eine Leerzeile.[Mel08] Es ist nicht möglich, eine E-Mail an mehrere Empfänger gleichzeitig zu verschicken, sondern jeder Empfänger muss gesondert angeschrieben werden. Wie in Second Life üblich, wird auch hier darauf

¹³Instant-Messaging-Spam oder Spim ist eine unerwünschte E-Mail die in Form einer Instant Message auf dem Bildschirm erscheint http://encarta.msn.com/dictionary_701710031/spim.html

geachtet, dass es keine Missbrauchsmöglichkeit gibt. Deswegen gibt es nach dem Versenden der E-Mails eine Verzögerung von 20 Sekunden. Erst danach können E-Mails wieder versendet werden. Ein Versenden von Massenmails wird durch diese Verzögerung unmöglich gemacht. Dadurch wollen die Administratoren von Second Life vermeiden, dass die Second Life-Server auf einer Blacklist¹⁴ landen.

HTTP-Requests HTTP-Requests wurden mit der Version 1.10 im Mai 2006 freigegeben.[Mel08] Davor war es nur möglich, Informationen effizient über E-Mails zu versenden. Die Kommunikation über XMLRPC gab es zwar noch, aber sie wurde wegen der Einschränkungen und der unvollständigen Implementierung nicht bevorzugt. XMLRPC wird im nächsten Paragraphen genauer erklärt.

Die HTTP-Funktion in Second Life funktioniert nur in eine Richtung, d.h. ein Nutzer kann keine Informationen von außen in Second Life über einen HTTP-Request verschicken, wenn die Kommunikation nicht aus Second Life gestartet wurde. Zu jedem HTTP-Request gehört ein HTTP-Response. Über diesen HTTP-Response können Informationen ins Second Life gelangen.

XMLRPC eXtensible Markup Language Remote Procedure Call (XMLRPC) ist eine weitere Methode, Informationen aus Second Life zu übermitteln.[Mel08] Es war vor den HTTP-Requests die einzige Methode, eine Kommunikation zwischen einem Objekt in Second Life und dem Internet zu ermöglichen. Der Zweck von XMLRPC ist es, Daten zwischen verschiedenen Systemen und verschiedenen Programmiersprachen auszutauschen. Linden Scripting Language erlaubt diese Art von Interaktion nicht. Deswegen ist es bei dieser Implementierung nur möglich, von außen auf ein Objekt in Second Life zuzugreifen. Es ist aber nicht möglich, einen Aufruf von Second Life aus, über XMLRPC, zu starten. Bisher kann man nur einen XMLRPC-Kanal öffnen und dann über einen HTTP-Request oder per E-Mail der betreffenden Applikation diesen Kanal mitteilen.

Notecards Notecards sind kurze Notizen, die in Second Life erstellt werden können. [Mel08] Diese Notizen werden oft als Werbegeschenke oder Anleitungen für die Benutzung bestimmter Räume und Objekte im Second Life gebraucht.

¹⁴<http://www.email-marketing-reports.com/lists/>

Buchstabengröße und Buchstabenart sind vorgegeben und können nicht verändert werden.

Leider kann der Inhalt dieser Notizen nicht durch ein Skript verändert werden.¹⁵. Dies führt in vielen Foren zu teilweise heftigen Diskussionen über den Sinn dieser Einschränkung. Bisher ist keine Einigung zwischen den Bewohnern und Linden Labs in Sicht.

3.2 Second Life-Konkurrenz

3.2.1 Active Worlds

Active Worlds ist eine virtuelle Umgebung, die zu MUVE gehört.¹⁶ Die Umgebung ist in einzelne Welten aufgeteilt, die von Benutzern erstellt werden. Die Gemeinde von Active Worlds zählt mittlerweile mehrere Tausend Mitglieder, die aktiv an der Umgebung mitarbeiten.

Avatare Das Besondere an Active Worlds sind zwei verschiedene Avatartypen, die der Nutzer verwenden kann.¹⁷ Zum einen gibt es den Touristen. Der Tourist ist ein Avatar, der für eine Erkundung benutzt wird. Für diesen Avatar ist keine Anmeldung notwendig. Der Tourist hat deswegen auch zahlreiche Einschränkungen. Der Nutzer ist nicht registriert, d. h. er kann nicht sicher sein, dass sein Name jederzeit in Active Worlds verfügbar sein wird. Weiterhin kann der Tourist wegen der fehlenden Registrierung auch kein Land erwerben oder irgendwelche Geschäfte in der virtuellen Welt tätigen. Wegen des fehlenden Namens kann der Benutzer auch kein Mitglied der Gemeinde werden. Die Auswahl der Avatare für den Touristen beschränkt sich auf die Auswahl des Geschlechts. Dabei kann der Benutzer jederzeit das Geschlecht des Touristen ändern. Äußere Merkmale, wie die Kleidung, können hier nicht verändert werden.

Der Tourist eignet sich für die Erforschung der virtuellen Welten in Active Worlds. Der Nutzer hat die Möglichkeit, die virtuellen Welten zu testen, ohne sofort dafür

¹⁵Es handelt sich um eine weitere Einschränkung von Seiten des Betreibers von Second Life, wie hier nachzulesen ist: <https://jira.secondlife.com/browse/SVC-402>

¹⁶vgl. <http://www.ActiveWorlds.com/>

¹⁷<http://www.ActiveWorlds.com/products/citizenships.asp>

Geld auszugeben. Wenn der Nutzer weitere Features verwenden möchte, muss er den vollen Account beantragen. Dieser Account ist der Citizen-Account.

Mit dem Citizen-Account kann der Benutzer seinen Avatar individuell anpassen. Außerdem kann der Nutzer Gegenstände kaufen und verkaufen.

Bei Active Worlds ist es ebenfalls möglich, eigene Welten zu erstellen, die unabhängig von der Hauptwelt sind.¹⁸ Diese Welten sind für Geschäftskunden interessant, die zwar eine virtuelle Welt haben wollen, aber gleichzeitig diese Welt vor ungebeten Gästen abschirmen möchten. Die Kunden können in dieser Welt eine Präsentation oder Werbung für eigene Produkte erstellen. Um eine eigene Welt ohne Anbindung an die Hauptwelt zu erstellen, muss der Nutzer Lizenzgebühren zahlen.

Active Worlds und das Bildungswesen Bildungsinstitute haben mittlerweile die 3D-Welten für sich entdeckt. Deswegen hat Active Worlds Inc., das Unternehmen hinter Active Worlds, das ActiveWorlds Educational Universe (AWEDU)¹⁹ ins Leben gerufen.

AWEDU stellt die Active Worlds-Technologie Ausbildungsinstituten, Dozenten, Schülern und Studierenden zur Verfügung. Die Ausbilder können dadurch neue Konzepte ausprobieren, die auf der 3D-Technologie von Active Worlds beruhen.

3.2.2 Croquet

Croquet ist eine weitere virtuelle Umgebung, die zur Zeit genutzt wird. Sie wurde als eine dreidimensionale Schnittstelle erstellt, die es den Benutzern ermöglicht, mit verteilten Systemen in einem Peer-to-Peer-Netzwerk [Dro05] zusammen zu arbeiten. Der Quellcode von Croquet wurde, wie bei Second Life auch, unter die GPL gestellt. Dadurch ist es jedem interessierten Programmierer erlaubt, den Quellcode einzusehen und zu verändern.²⁰

Squeak Squeak ist ein Dialekt von Smalltalk²¹. Es stellt die einzige Möglichkeit dar, in Croquet die Welten und Objekte zu erstellen und zu verändern.

¹⁸<http://www.ActiveWorlds.com/products/worlds.asp>

¹⁹<http://www.ActiveWorlds.com/edu/awedu.asp>

²⁰http://cscw.in.tu-clausthal.de/pdf/sl_croq.pdf

²¹<http://www.heeg.de/smalltalk/smalltalk.html>

Croquet-Programme Es gibt fünf Programme, die schon fest in Croquet integriert sind. Dies sind Croquet Chat, Tea Painter, Screenshot, Croquet-Views und VNC. Diese werden nun näher erläutert.

Croquet Chat Croquet Chat ist die einzige Möglichkeit der verbalen Kommunikation zwischen den Avataren. Es gibt drei Modi der Kommunikation. Dies sind Flüstern, Sprechen und Schreien. Der Unterschied zwischen diesen Kommunikationsarten ist die Reichweite, über die die anderen Avatare den sprechenden Avatar hören können. Dadurch kann der Kreis der Avatare, die einem zuhören, bestimmt werden.

Tea Painter Mit dem Tea Painter können eigene Polygone erstellt und in die virtuelle Welt eingesetzt werden. Der Tea Painter arbeitet ähnlich wie das Programm Paint unter Windows. Es ist also möglich, direkt mit der Maus in der virtuellen Welt zu malen und diese Bilder als Polygone in der Welt darzustellen.

Screenshot Screenshot ist eine Möglichkeit, ein 2D-Abbild der aktuellen Welt zu erstellen. Diese Funktionalität ist auch in Second Life gegeben.

Croquet Views In Croquet gibt es die Möglichkeit, externe Programme in einem Fenster auszuführen und mit ihnen zu arbeiten. Diese Programme müssen aber, im Gegensatz zum VNC, auf welchen wir später zu sprechen kommen, in Croquet implementiert sein. Dieser Ansatz erlaubt es, Textdateien und Bilder in Croquet anzuzeigen.

VNC Der VNC-Client ist ein weiteres Programm, das in Croquet implementiert ist. Es handelt sich um eine Erweiterung des Croquet Views. Diese Erweiterung ermöglicht dem Benutzer, jedes Programm in Croquet als Croquet View darzustellen. Eine interessante Variante ist das Ausführen von Second Life als View in Croquet. Leider gibt es keine Möglichkeit des Datenaustauschs zwischen Croquet und Second Life, da ein solches Feature in Croquet nicht implementiert ist.

Es ist sogar möglich, Videos in VNC darzustellen. Leider ist die Performance des Viewers sehr eingeschränkt. Die Geschwindigkeit beträgt lediglich ein paar Bilder pro Sekunde. Deswegen wird das Abspielen von Videos in der Praxis nicht eingesetzt. Die weitaus interessantere Möglichkeit, die sich hier bietet, ist das



Abbildung 3.3: Simulation von Second Life innerhalb von Croquet

Benutzen des Inworld Browsers. Ein Inworld Browser ist ein Browser, der in der virtuellen Umgebung gestartet wird. Die Webseiten brauchen keine schnelle Aktualisierung, und können deshalb in Croquet bedenkenlos eingesetzt werden.

3.2.3 Nachteile von Croquet

Obwohl Croquet seit 2001 entwickelt wird, ist das Programm immer noch nicht stabil. Auf manchen Systemen ist es anfällig für Abstürze. Außerdem erfordert es immer die aktuellsten Treiber. In der Open Source-Gemeinde werden diese Fehler normalerweise schnell behoben. Im Fall von Croquet, das auch Open Source ist, ist die Entwicklergemeinschaft klein, da Croquet nicht weit verbreitet ist. Deswegen gibt es sehr wenige Tutorials. Die Tutorials, die existieren, sind meist veraltet. Ein weiterer Nachteil von Croquet ist die Entwicklungssprache Squeak. Die Sprache ist im Gegensatz zu LSL schwierig zu erlernen.

3.2.4 Google Lively

Lively war ein Konkurrenzprodukt zu Second Life. Es bestand aus einer 3D-Welt, die wie bei Second Life mit Hilfe des Avatars begangen werden konnte. Anders als Second Life brauchte Lively zum Programm noch einen Browser. Der Vorteil von Lively war, dass es auch in Webseiten integriert werden konnte. Als Lively startete, wurden nur Mozilla Firefox und Internet Explorer unter Microsoft Windows unterstützt. Das Programm war Open Source, wie Second Life auch, d. h. der Quellcode konnte vom Benutzer eingesehen und verändert werden. Bei Lively handelte es sich noch um Beta-Software, die erst im Juli 2008 für die Öffentlichkeit freigegeben wurde. [liv08b]

Anders als bei Second Life wurde hier schon an die Einbindung von Web 2.0-Programmen gedacht, und diese wurde bereits teilweise implementiert. So war es möglich, YouTube-Videos, Blogs oder andere Webseiten einzubinden. Da die Software noch nicht vollständig implementiert und ausgetestet wurde, waren die Betreiber auf die Mitarbeit der Benutzer angewiesen. Deshalb war Lively noch kein kommerzielles Produkt, aber kommerzielle Komponenten waren schon in Planung. Es war möglich, aus einem Katalog Avatare, Ausrüstung für dieselben, Möbel und weitere Einrichtungsgegenstände zu bestellen. Diese Bestellung war noch kostenlos, aber es war anzunehmen, dass sich dies bald ändern würde und einzelne Komponenten bezahlt werden müssten.

Google Lively wurde am 31. Dezember 2008 eingestellt, weil Google keine finanziellen Vorteile in der Weiterführung dieser virtuellen Welt sah.²²

3.3 Web 2.0

Das Web 2.0 ist ein weites Gebiet auf dem viele Applikationen und Implementierungen existieren. Die einzelnen Web 2.0-Anwendungen wurden schon im Kapitel 2.4 behandelt. Deshalb befasst sich der nächste Abschnitt mit der Architektur von Wiki-Systemen. Weiterhin werden die Wiki-Bots, die eine automatisierte Bearbeitung der Wiki Seiten ermöglichen vorgestellt. Als Abschluß bekommt der Leser einen Überblick über verschiedene Verlinkungsmethoden, die von Media-Wiki unterstützt werden.

²²<http://www.lively.com/goodbye.html>

3.3.1 Architektur von Wiki-Systemen

Der am häufigsten eingesetzte Webserver ist der Apache Webserver. Er unterstützt alle gängigen Plattformen, wie Unix, GNU Linux und Microsoft Windows. [apa08] Dieser kann dementsprechend auf allen Servern im Internet installiert werden. Die Vielzahl der Implementierungen, die auf der offiziellen Webseite eingesehen werden können, ist ein weiterer Grund für die Verbreitung dieses Systems. Weitere Server, die gerne eingesetzt werden, sind der Cherokee²³ oder der acIIS²⁴ von Microsoft.

Wie schon in Kapitel 2.5 erwähnt, benötigen manche Wikis Datenbanken. Die Datenbanken, die hier benutzt werden, sind die meistverbreiteten im WWW. Es handelt sich um MySQL, Oracle, DB2 und weitere. Das Speichern von Daten in Datenbanken hat einige Vorteile gegenüber dem Speichern in Dateien. Die Verwaltung der Daten fällt hierbei in den Bereich der Datenbanken, die darauf ausgelegt sind, große Datenmengen zu organisieren. Dadurch ergibt sich zudem die Möglichkeit, dass mehrere Benutzer gleichzeitig ein Dokument bearbeiten, ohne dass es zu Problemen kommt, wie sie bei mehrfachen Zugriff auf Dateien auftreten.

Die Verwaltung der Daten ist aber nicht der einzige Vorteil der Datenbanken gegenüber der Speicherung in Dateien. Die Sicherung der Daten ist auch eine Funktion, die eine Datenbank sehr gut erfüllen kann. Die Datenbank kann in wenigen Handgriffen gesichert und, wenn nötig, wieder eingespielt werden.

Der Zugriff auf die Datenbanken wird durch Skripte erledigt. Die Skriptsprachen Perl, Python und PHP haben sich hierbei als besonders nützlich erwiesen. Sie sind einfach zu erlernen, aber trotzdem sehr mächtig. Außerdem wurden die erforderlichen Schnittstellen schon in die Programmiersprachen implementiert. Die Skripte haben die Aufgabe, die Daten aus den Datenbanken zu holen und sie in das HTML-Format zu übersetzen. Diese Seiten werden an den Webclient geschickt. Der Benutzer hat zu keinem Zeitpunkt einen direkten Zugriff auf die Datenbank, was Angriffe auf diese erschwert.

Für die Kommunikation wird das bewährte HTTP-Protokoll benutzt [Dro05]. HTTP ist ein mächtiges Protokoll, das auf TCP/IP [Dro05] aufsetzt. Dadurch ergibt sich eine gesicherte Übertragung, bei der keine Pakete unbemerkt verloren

²³<http://www.cherokee-project.com/>

²⁴vgl. <http://www.iis.net/>

gehen können.

Die Programmiersprache PHP ist eine Skriptsprache, die häufig im Internet eingesetzt wird. Mit ihrer Hilfe werden Webanwendungen und Webseiten dynamisch erstellt. Die MediaWiki-Software wurde in PHP geschrieben und benutzt MySQL als Datenbank. Weiterhin wird der Apache Webserver in MediaWiki eingesetzt. Wie wir hier sehen, werden in MediaWiki Komponenten eingesetzt, die große Verbreitung im Internet haben und als Standard für die Webanwendungen angesehen werden.

3.3.2 Wiki-Bots

Bots sind Programme, die häufig auftretende Aufgaben durchführen. Dabei können die Bots diese Aufgaben schneller lösen als ein Mensch. Zur Zeit existieren einige Bots, die in Wikipedia [wik09a] ihren Dienst verrichten ²⁵.

Ein Wiki Bot liest die Wiki Seiten, wie ein menschlicher Benutzer. Dabei führt er die Veränderungen dort aus, wo seine Programmierung es vorsieht. Der Bot kann dabei die Informationen schneller verarbeiten. Außerdem leidet ein Bot nicht unter Müdigkeit oder fehlender Konzentration, wie es bei menschlichen Benutzern der Fall sein kann.

Das Erstellen einer Webseite im Wiki-Format durch ein Skript ist eine besondere Herausforderung. Wikis werden wie alle anderen Web 2.0 Applikationen über den Browser bedient. Es ist nicht vorgesehen, sie anderweitig zu nutzen. Was auf der einen Seite Vorteile für den normalen Endbenutzer bringt, hat Nachteile für den Programmierer. Der Programmierer muss die Funktionen eines Browsers simulieren und die grafische Interaktion mit dem Wiki abfangen.

Wiki-Administratoren müssen manchmal ganze Serien von Seiten verschieben, bearbeiten oder sichern. In einem Wiki sind nicht selten mehrere tausend Wiki-Seiten gespeichert. Eine manuelle Administration über den Browser ist deshalb nicht praktikabel.

Die Bots haben die Aufgabe, den Administratoren die tägliche Arbeit zu erleichtern. Aus der Sicht des Wikis sind die Bots normale Benutzer mit einem eigenen Account. Je nachdem, wie sie implementiert sind, können Bots aber sehr mächtige Hilfswerkzeuge sein. Sie können in einem Schritt und in nur ein paar Sekunden tausende Seiten bearbeiten, Tippfehler ausbessern und andere häufig auftretende

²⁵<http://en.wikipedia.org/wiki/Wikipedia:Bots/status>

Aufgaben erledigen.

Gerade diese Fülle an Aufgaben macht Bots aber auch zum Ziel von Angriffe böswilliger Nutzer. Wenn ein Bot in falsche Hände gerät, kann er enormen Schaden anrichten. Die meisten Bots haben zwar keine Erlaubnis Seiten zu löschen, aber auch das Bearbeiten einer Seite kann sich verheerend auswirken. Das Ergebnis eines erfolgreichen Angriffs auf ein populäres Wiki, wie z.B. Wikipedia²⁶, wären korrupte Informationen auf den Seiten, die dem Ruf des Wikis und des Betreibers schweren Schaden zufügen.

Deshalb muss vor der Benutzung von Bots bei Wikipedia eine Reihe von Auflagen erfüllt werden.[wik09b] Der Bot muss über ein eigenes Nutzerkonto verfügen. Zudem muss der Bot eine eigene Benutzerseite haben, auf der die Aktionen und die Vorgehensweise des Bots erklärt werden. Dadurch können die Administratoren von Wikipedia darauf reagieren und notfalls den Bot sperren, wenn sie in ihm eine Gefährdung sehen. Die normale Frist bis zur Freigabe des Bots dauert mindestens eine Woche. Falls niemand Einspruch auf die Funktionsweise erhoben hat, wird der Bot freigegeben und kann seine Arbeit verrichten.

Wenn der Bot freigegeben ist, kann er trotzdem nicht alles machen, was seinem Programmierer in den Sinn kommt. Der Bot darf nur im Artikelnamensraum²⁷ arbeiten. Die Benutzerseiten darf er üblicherweise nicht bearbeiten. Diese Regel erlaubt auch Ausnahmen, aber diese müssen diskutiert werden, bevor sie freigegeben werden.²⁸

Der Bot ist sehr schnell und effizient. Gerade diese Attribute können ihn aber bei den Wiki-Betreibern unbeliebt machen. Wenn ein Bot zu viele Anfragen an den Server stellt, kann er diesen leicht an die Grenzen der Belastbarkeit bringen. Bei Wikipedia [wik09a] gilt es als geduldet, wenn der Bot maximal 5 *Edits* pro Minute durchführt [wik09b]. Wenn der Bot zu viele Anfragen stellt, kann es passieren, dass er bzw. sein Account gesperrt wird, damit der Server nicht überlastet wird. Dieses Beispiel bei Wikipedia zeigt, wie viel Macht ein Benutzer mit einem Bot haben kann. Deshalb wurden diese strikten Regeln gesetzt, deren Einhaltung streng überwacht wird. Die Bots bewegen sich durch die Linkstruktur des Wikis. Dazu werden nun auch die verschiedenen Verlinkungsarten in den Wikis vorgestellt.

²⁶vgl. <http://wikipedia.org>

²⁷vgl. <http://de.wikipedia.org/wiki/Hilfe:Namensräume>

²⁸http://en.wikipedia.org/wiki/Wikipedia:Creating_a_bot/Editing.3B_edit_tokens

3.3.3 Links

Ein Link ist ein Querverweis auf ein anderes Dokument oder eine Datei innerhalb eines Textes. Ein Link besteht immer aus einem Verweisziel und einer Beschreibung des Ziels. Wir unterscheiden zwischen internen Links und externen Links. Die internen Links verweisen auf eine Stelle innerhalb des Dokuments. Die externen Links sind Verknüpfungen zu einem anderen Dokument, welches meistens eine verwandte Thematik behandelt.²⁹

Im Wiki werden die zwei Linkarten noch weiter unterteilt. Die geläufigsten Linkarten werden hier vorgestellt.

Wikilink Ein Wikilink ist ein interner Link. Dabei wird von einer Seite auf den Titel einer anderen Seite verlinkt. MediaWiki überprüft bei diesem Vorgang automatisch, ob die Seite existiert. Falls die Seite noch nicht angelegt wurde, wird dem Benutzer angeboten, die Seite zu erstellen. Solche Links werden mit roter Farbe gekennzeichnet und nennen sich deswegen “*rote Links*”. Die Wörter in CamelCase Schreibweise werden automatisch als Links angesehen. Die meisten Wikis verwenden noch eine andere Schreibweise wie z.B. eckige Klammern bei MediaWiki `[Seitenname|Linkname]`. Der Linkname ist dabei optional.

Interwiki Link Der Interwiki Link verlinkt eine Seite mit einem Text. Das Ziel muss kein Wiki sein, obwohl es der Name vermuten lässt. Wenn die Seite kein Wiki ist, dann muss sie auf der Interwiki Map verzeichnet sein. Eine Interwiki Map wird immer für das Quellwiki spezifiziert.

Externer Link Mit externen Links kann jede Seite mit dem Wiki Text verlinkt werden. Dabei muss immer die Komplette URL angegeben werden, wie zum Beispiel: `[http://www.beispiel.org Linkname]`. Der Linkname ist auch hier optional. Die externen Links, die keinen Namen haben, werden durchnummeriert.

Sprachlink Die Sprachlinks verlinken auf ein Dokument im selben Projekt. Dabei ändert sich die Sprache des Textes. Bei der Angabe des Links muss als Präfix die Sprache angegeben werden. Beispiel: `[[en:MediaWiki]]` Dieses Beispiel verlinkt auf die englische MediaWiki Seite.

²⁹Link zum MediaWiki Buch

Kategorien Kategorien ermöglichen es, Dokumente verschiedenen Themengebieten, abhängig vom Inhalt, zuzuordnen. Die Klassifikation erfolgt, indem ein Link von dem Dokument zur Kategorie gesetzt wird. In MediaWiki werden Kategorielinks in einem speziellen Format am Ende des Gesamttextes platziert.[VKV⁺06] Ein Kategorielink ist ein ganz normaler Link mit Präfix Kategorie - Beispiel in MediaWiki: *[[Kategorie: Naturwissenschaft]]*. Befindet man sich auf der Kategorie-seite, kann man sich alle Rückverweise anzeigen lassen und erhält somit alle Dokumente, die zu dieser Kategorie gehören. Kategorie-seiten können wiederum auch einer anderen Kategorie zugeordnet werden. Dadurch entsteht eine hierarchische Struktur.[Lan05]

3.4 Konkurrenzprodukte

Das Lernen in modernen Universitäten wird zunehmend mit digitalen Komponenten angereichert, die den Informationsaustausch und die Zusammenarbeit erleichtern sollen. Mittlerweile gibt es schon einige Implementierungen, wie z.B. WebCT³⁰, Blackboard³¹ oder Moodle [moo09a], die webbasierte Lernumgebungen darstellen. Diese Umgebungen sind nicht mehr einfache Datenhäfen, sondern sie stellen online Chaträume zur Verfügung. Desweiteren gibt es z.B. auch die Möglichkeit Multiple-Choice Tests durchzuführen. Die Kommunikation im LMS ist aber immer noch durch eine fehlende Visualisierung eingeschränkt. Die Nutzer können Dokumente austauschen, aber noch nicht interaktiv erstellen. Deswegen bietet es sich an, die Lernumgebungen mit Second Life zu verbinden, um dadurch auch Präsentationen halten zu können oder anderweitig die Informationen in der 3D-Umgebung vorzustellen, um sie den Studierenden und Schülern näher zu bringen. Es gibt einige Programme, die Informationen aus Second Life herausführen und sie außerhalb von Second Life aufbereiten und präsentieren. Wir werden hier Sloodle und Virtual World Mashups beschreiben.

3.4.1 Sloodle

Sloodle ist ein freies Plug-in für Second Life, welches Werkzeuge im virtuellen Raum anbietet, die die bestehende Unterstützung der Lehrens in Second Life er-

³⁰<http://www.uni-frankfurt.de/fb/fb08/FABacht/webct/index.html>

³¹<http://e-learning.ruhr-uni-bochum.de/>

weitert. Dabei wird Second Life mit dem LMS Moodle verbunden.³²

Sloodle hat mittlerweile viele Features. Hier sind unter anderem folgende zu nennen:

- **Web Intercom** ist ein Chatraum, der den Moodle Chat und den Second Life Chat vereint. Die Studierenden können im Second Life-Chat teilnehmen, indem sie den Moodle Chatraum benutzen. Die Diskussionen, die im Chatraum geführt werden, können in der Moodle Datenbank gespeichert werden.
- **Registrierung** Die Avatare der Bewohner in Second Life werden automatisch mit den Moodle Benutzeraccounts verlinkt.
- **Test Werkzeug** ist ein weiteres Feature von Sloodle. Dabei werden verschiedene Prüfungen oder Modellierungsaufgaben im Second Life gestellt. Wenn die Aufgaben gelöst wurden, werden die Noten automatisch in Moodle eingetragen und können dort eingesehen werden.
- **Abstimmungstool** Mit diesem Tool können Studierende in Second Life und Moodle abstimmen und Ergebnisse in beiden Anwendungen sehen.
- **Sloodle Multifunktionswerkzeuge** Die Multifunktionswerkzeuge erweitern das Second Life Interface des Benutzers. Durch die Werkzeuge kann der Bewohner die Liste der anderen Avatare in der Umgebung sehen oder eigene Blogs aus Second Life schreiben.

Im nächsten Kapitel wird genauer auf die Registrierung und Anmeldung eingegangen. Dabei wird ersichtlich wie Sloodle die Anmeldung von Second Life mit der Anmeldung von Moodle verbindet.

3.4.1.1 Registrierung und Anmeldung

Die Registrierung, auch Authentifizierung genannt, ist das Setzen der Verbindung zwischen dem Second Life Avatar und dem Moodle-Account, der vom gleichen Besitzer stammt.³³ Der Avatar und der Moodle-Account müssen nur einmal miteinander verlinkt werden. Danach wird der Avatar automatisch erkannt. Es gibt

³²<http://www.sloodle.org/moodle/file.php/1/UsingSLOODLEHeatSheet1.pdf>

³³http://slisweb.sjsu.edu/sl/index.php/Registration_and_Enrolment

zwei Arten der Registrierung, die manuelle und die automatische Methode.

Die manuelle Methode kann von einigen Objekten, wie der Sloodle Toolbar, durchgeführt werden. Der Benutzer meldet sich über eine Universal Resource Locator (URL) in Second Life auf der Moodle Seite an. Das manuelle Anmelden bietet sich an, wenn der Benutzer noch keinen Second Life Account besitzt oder wenn er/sie Moodle unabhängig von Second Life nutzen möchte.

Die automatische Registrierung ist eine Besonderheit, die seit der Sloodle Version 0.3 stabil ist. Dabei muss in der Konfigurationsseite von Sloodle die automatische Registrierung freigegeben werden. Falls der Benutzer nicht über die Administratorrechte bei Moodle verfügt und diese Option nicht freigegeben ist, gibt es keine Möglichkeit, die automatische Registrierung zu benutzen.

Die Vorteile der automatischen Registrierung liegen in der Bequemlichkeit, die sie mit sich bringt. Der Avatar kann sich automatisch anmelden und mit Moodle verbinden, ohne mühevoll die manuelle Methode nutzen zu müssen.

Moodle Moodle ist eine Lernplattform auf Open-Source Basis. Die Software bietet Dozenten die Möglichkeit, die Kursräume mit ihren Lerninhalten zu füllen.³⁴ Die Software bietet die Möglichkeiten zur Unterstützung kooperativer Lernmethoden. Moodle stellt Kursräume zur Verfügung, in denen Arbeitsmaterialien und Lernaktivitäten bereitgestellt werden. Jeder Kurs kann mit einer Filterung versehen werden. Die Filterung kann so eingerichtet werden, dass nur bestimmte Teilnehmer oder Gäste an dem Kurs teilnehmen können. Zusätzlich ist eine Sicherung durch ein Passwort implementiert. Moodle bietet die Möglichkeit, sich die Inhalte wöchentlich nach Themen gegliedert anzeigen zu lassen. Als Arbeitsmaterialien dienen Texte, Links oder Dateien.

3.4.2 Virtual World Mashups

Die Verbindung von Second Life und Web 2.0 nennt sich Mashup, bzw. Vernetzung dieser beiden Anwendungsgebiete. Die bekanntesten Mashups sind der Slurlmaker, der Delicious und Second Life verbindet³⁵, und Matt Biddulphs Flickr

³⁴<http://www.campussource.de/software/moodle/>, Stand 21.Juli 2008

³⁵<http://mashable.com/2006/05/30/second-life-web-20-virtual-world-mashups/>

hack³⁶. Zunächst werden die beiden Web 2.0-Anwendungen Delicious und Flickr beschrieben. Danach wird auf die Verbindung zwischen diesen Anwendungen und Second Life eingegangen.

3.4.2.1 Delicious

Delicious ist einer der populärsten Bookmarking Dienste im Web.³⁷ Das Programm, das 2003 von Joshua Schachter entwickelt wurde ist einfach zu bedienen. Die Software dient der kostenlosen Ablage der Lesezeichen im WWW. Diese Lesezeichen können von anderen Nutzern eingesehen werden und der Nutzer selbst kann sehen, was andere gespeichert haben.

Delicious kann durch die große Menge der Lesezeichen, die es speichert, eine Liste der beliebtesten Lesezeichen erstellen.³⁸ Es ist aus der Analyse ersichtlich, wie viele Benutzer eine URL zu ihren Lesezeichen hinzugefügt haben. Dadurch ergibt sich eine Bewertung der Webseiten durch den Benutzer und nicht nach der Anzahl der ausgehenden und eingehenden Links, wie es heute bei Suchmaschinen[Dro05] üblich ist. Diese Beliebtheitslisten können auch nach Interessengebieten sortiert werden.

Durch die Online-Speicherung der eigenen Lesezeichen kann der Nutzer des Dienstes jederzeit und von jedem Rechner aus auf die Lesezeichen zugreifen. Neben dem Zugriff von überall, organisiert Delicious auch die Lesezeichen. Dazu benutzt der Dienst Schlagwörter. Die Schlagwörter sind Wörter, die einen Link beschreiben. Alle Lesezeichen, die z.B. mit "Lustig" und mit "Video" versehen sind, landen automatisch in der "Lustig"- und in der "Video"-Kollektion.

3.4.2.2 Flickr

Flickr ist zufällig entstanden. Es wurde 2002 als Nebenprodukt des Spiels "Neverending" der Firma *Ludicorp*³⁹ entwickelt.⁴⁰ Das Programm sollte nur das Hochladen der Bilder vereinfachen. Mit der Zeit wurde es klar, dass Flickr erfolgversprechender war als das eigentliche Spiel. Deswegen verlagerte die Firma, ihre Anstrengungen von dem Spiel auf Flickr.

³⁶<http://www.hackdiary.com/2006/05/29/alas-second-life-web-20-in-a-virtual-world/>

³⁷vgl. [Alb07]

³⁸<http://delicious.com/help/learn>

³⁹<http://ludicorp.com/>

⁴⁰vgl. [Alb07]

Wie bereits erwähnt, ist Flickr eine Software für das Hochladen von Bildern. Dafür bietet Flickr Tools an. Wenn man keine Tools verwenden will, gibt es auch die Möglichkeit, die Bilder direkt über die Flickr-Webseite hochzuladen. Eine weitere Möglichkeit ist das Hochladen der Fotos über E-Mails.

Die hochgeladenen Bilder werden in Alben gespeichert und können mit einem Schlagwort versehen werden. Flickr bietet zusätzliche Funktionen, die auch ein Bearbeiten der Bilder ermöglichen.⁴¹

Wie bei vielen anderen Web 2.0 Programmen ist auch hier der Community Gedanke sehr stark vertreten. Die Bilder, die ein Benutzer hochlädt, können von anderen Benutzern eingesehen, bewertet und kommentiert werden. Diese Möglichkeit ist aber nur optional. Die eigenen Alben kann jeder Benutzer entweder öffentlich machen, lediglich einer Gruppe die Einsicht gewähren oder sie als privat einstufen.

Flickr selbst ist auch mit anderen Web 2.0-Anwendungen verknüpft. Die Benutzer können nämlich unter Flickr über die eigenen oder fremden Bilder bloggen.

Flickr finanziert sich vorwiegend mit Partnerschaften zu kostenpflichtigen Diensten, die Digitalfotos drucken⁴².

3.4.2.3 Mashups

Slurlmarker verbindet Delicious und Second Life, und erlaubt dadurch Lesezeichen zu Gebieten, die sich in der virtuellen Welt befinden, zu setzen. Diese Lesezeichen werden als SLurl oder Second Life URL gespeichert und erscheinen bei Delicious/tag/slurlmarker. Wenn auf dem Rechner Second Life installiert ist, kann der Nutzer mit einem Klick auf den Link sich direkt auf die Stelle teleportieren, die mit dem Second Life Lesezeichen vermerkt wurde.

Der zweite Mashup ist der Flickr hack von Matt Biddulph. Diese Anwendung erlaubt das Anzeigen der Fotos aus Flickr in der virtuellen Welt von Second Life.

3.5 Anforderungen

Es soll eine Schnittstelle zwischen Second Life und einer Web 2.0-Anwendung aufgebaut werden. Diese Schnittstelle soll eine Zwei-Wege-Verbindung darstellen.

⁴¹<http://www.flickr.com/tour/>

⁴²<http://www.flickr.com/help/printing/>

Dabei müssen die Daten hauptsächlich von Second Life zu Web 2.0-Anwendungen fließen. Die zweite Richtung dient den Fehlermeldungen, die von den Web 2.0-Anwendungen gesendet werden. Sonst wird diese Verbindung nicht genutzt, um die Ressourcen der Server zu schonen.

Das Skript muss von mehr als einer Person benutzbar sein. Das Programm muss dabei unterscheiden können, von welchem Bewohner die Daten verschickt werden, um die Daten richtig zu verarbeiten.

Das Programm soll die Gespräche in Second Life aufnehmen und archivieren. Dabei werden die Daten in einer Web 2.0-Anwendung gespeichert. Das Programm soll lediglich die Gespräche im Textchat übermitteln. Für den Voice-Chat ist eine Protokollierung nicht vorgesehen.

Das Programm soll weiterhin E-Mails über den entfernten Server verschicken können. Dadurch wird es möglich, E-Mails an Personen zu verschicken, die keine Bewohner von Second Life sind. Die E-Mails, die auf diesem Weg verschickt werden, sollen archiviert werden, damit sie später einsehbar sind. Die Archivierung wird in einer Web 2.0-Anwendung geschehen.

Weiterhin soll es möglich sein, neue Artikel anzulegen und bestehende Artikel in der Web 2.0-Anwendung zu erweitern. Diese Funktion beinhaltet nur das Anhängen von neuen Inhalten an einen bestehenden oder neuen Artikel. Das Bearbeiten von Artikeln ist wegen der fehlenden Visualisierung nicht vorgesehen.

Ein Terminplaner wird zudem implementiert. Der Kalender befindet sich in der Web 2.0-Anwendung und wird von Second Life aus mit neuen Terminen beschrieben. Dabei sollen die Termine in der Web 2.0-Anwendung chronologisch richtig angeordnet werden.

Das Programm soll weiterhin Notecards, also Notizen, aus Second Life in die Web 2.0-Anwendung übertragen. Die zu übertragenden Notecards befinden sich im Objekt.

Nur berechnigte Nutzer dürfen diese Dienste in Anspruch nehmen. Deswegen muss auch eine Authentifizierung eingebaut werden. Da das Objekt in Second Life über keine solche Authentifizierung verfügt, ist es notwendig, dass die Überprüfung in der Web 2.0-Anwendung stattfindet. Das System soll auf der einen Seite im Second Life und auf der anderen Seite auf einem entfernten Server funktionieren. Diese beiden Bereiche sollen sich über eine gesicherte, aber nicht verschlüsselte Verbindung austauschen. Das System soll die Daten aus Second Life übertragen.

Wenn das nicht möglich ist, muss über die verschiedenen Fehlercodes, die HTTP zurückgibt, der Fehler ermittelt werden und dementsprechend soll auch das Programm bzw. der Benutzer sich darauf anpassen können. Die Austauschformate sind E-Mails und verschiedene andere Daten, die in einem Wiki veröffentlicht werden. Das System soll sich auf alle Web 2.0-Anwendungen anpassen können. Die Web 2.0-Anwendungen sind vielseitig und allein schon die Wiki-Derivate haben zum Teil signifikante Unterschiede vorzuweisen. Deshalb muss das System allgemein gehalten werden, damit relativ leicht auch weitere Schnittstellen eingefügt werden können.

Kapitel 4

Konzept

Auf Grundlage der definierten Anforderungen in Abschnitt 3.5 wird in diesem Kapitel das Konzept vorgestellt. Der Abschnitt 4.1 bezieht sich vor allem auf den Aufbau des Prototyps. Den ersten Teil bilden die Second Life Funktionen, die implementiert werden sollen ¹.

Die Aufgaben, die das System ausführen soll, müssen so aufgeteilt werden, dass einzelne Eingabe- und Ausgabekomponenten des Systems austauschbar sind. Deswegen ist eine Zwischenschicht notwendig, die die Kommunikation zwischen Second Life und Web 2.0 regelt. ² Der HTTP-Supervisor wird nicht nur den Datenverkehr regeln sondern auch die Sicherungskopien erstellen, die für den Fall eines Ausfalls des Systems benötigt werden.

Abschließend wird in Abschnitt 4.1.1 das Schreiben der verschiedenen Wikiseiten genau erläutert.

Für die Umsetzung der Anforderungen werden verschiedene vorhandene Kommunikationsprotokolle benutzt. Diese Protokolle werden im Kapitel 4.2 genauer beschrieben. Das besondere Augenmerk wird hierbei auf HTTP ³ gerichtet. Am Rande wird auch XMLRPC als eine weitere wichtige Kommunikationsmöglichkeit erwähnt ⁴.

In dieser Arbeit wurden öfter so genannte Client/Server Modelle erwähnt. Das Programm selbst bildet eine solche Architektur. Deswegen wird auf das Client/-Server Modell im Kapitel 4.3 eingegangen.

Abschließend ist es wichtig zu sagen, welchen Mehrwert diese Arbeit für die Ver-

¹vgl. Abschnitt 4.1.1

²vgl. Kapitel 4.1.2

³vgl. Kapitel 4.2.1

⁴vgl. Kapitel 4.2.2

anstaltung “Einführung in das IT-Projektmanagement” [itp05] hat, weil die Grundidee aus dieser Vorlesung stammt. Die Auseinandersetzung mit den Vorteilen, die diese Arbeit bringt, findet im Kapitel 4.4 statt.

4.1 Umsetzung der Anforderungen

Damit die Anforderungen umgesetzt werden können, muss das Programm in verschiedene Bereiche unterteilt werden. Der erste Teil des Programms befindet sich in Second Life. Dieser Teil wird in LSL geschrieben und sendet die erforderlichen Daten zum entfernten Server, der diese Informationen weiter verarbeitet.

Der zweite Teil des Programms nimmt diese Daten entgegen. Dieser Teil des Programms wird in PHP geschrieben und sorgt dafür, dass die Informationen an den Bot, der zum dritten Bereich gehört, geschickt werden. Der Server stellt in diesem Fall eine Weiche dar, die Informationen weiterleiten soll.

Der Bot läuft auf dem gleichen Server und bildet zusammen mit der Web 2.0 Anwendung den dritten Teil des Programms. Die Programmierung wird in diesem Teil der Implementierung auch in PHP geschrieben. Die weiteren Bots können auch in anderen Sprachen verfasst werden.

Es gibt viele virtuelle Welten und Web 2.0 Anwendungen, die auf unterschiedlichen Programmen basieren. In dieser Arbeit wurde als virtuelle Welt Second Life ausgewählt und die Web 2.0 Anwendung ist das MediaWiki. Die Abbildung 4.1 gibt das Konzept wieder.

Die Kommunikation bildet den vierten Teil der Implementierung. Da sich die Kommunikation zwischen dem ersten und dem zweiten, bzw. zwischen dem zweiten und dem dritten Bereich befindet, wird sie teilweise in LSL und teilweise in PHP implementiert.

4.1.1 Teil 1: Second Life

Der erste Teil des Programms ruft die einzelnen Funktionen auf. Diese Funktionen sind:

- Protokollierung von Gesprächen
- Verschicken von E-Mails

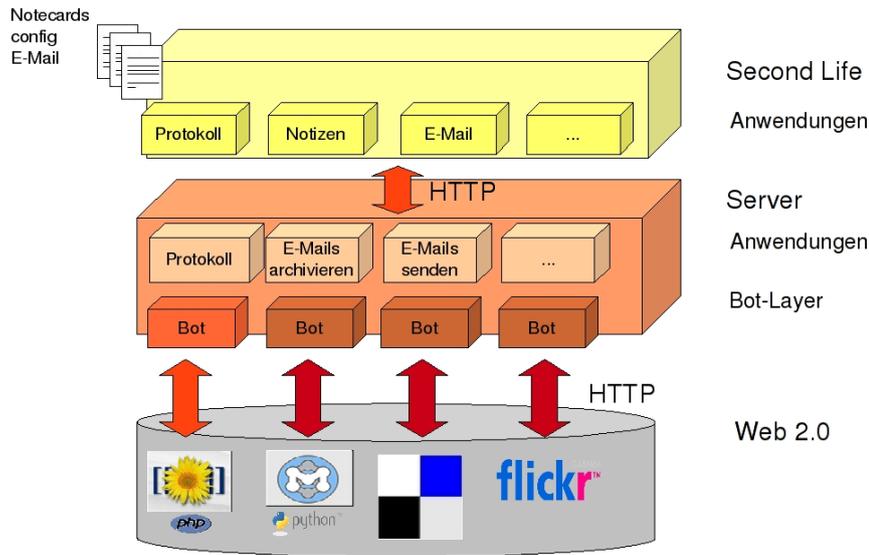


Abbildung 4.1: Schematische Darstellung des Konzepts der Arbeit

- Anlegen eines Artikels in MediaWiki
- Verschicken von Notecards
- Anlegen von neuen Terminen in einer Wiki Seite

4.1.1.1 Protokollierung von Gesprächen

Die Gründe für die Protokollierung der Gespräche haben wir schon in Kapitel 2 erklärt. An dieser Stelle ist noch eine Warnung angebracht. Das Ausspionieren von Gesprächen ist in Second Life ausdrücklich untersagt!⁵ Die Second Life Regeln sind aber so ausgelegt, dass erst eine Beschwerde bei der Second Life Administration zum Ausschluss aus Second Life führt. Deswegen müssen alle Personen, deren Gespräche aufgezeichnet werden sollen, damit einverstanden sein. Sicherheitshalber erscheint eine Warnung, wenn das Geschriebene aufgezeichnet wird. Das Protokoll benutzt die HTTP-Verbindung und schickt die Daten weiter an den Server. Das Programm zum Senden der Daten versieht diese automatisch mit dem Befehl *protokoll* und schickt sie als Rohdaten weiter.

Das Protokoll kann über den Dialog oder über den Chat gestartet bzw. beendet werden. Beim Starten des Protokolls über die Dialogfunktion berührt der Benut-

⁵<http://secondlife.com/corporate/tos.php>

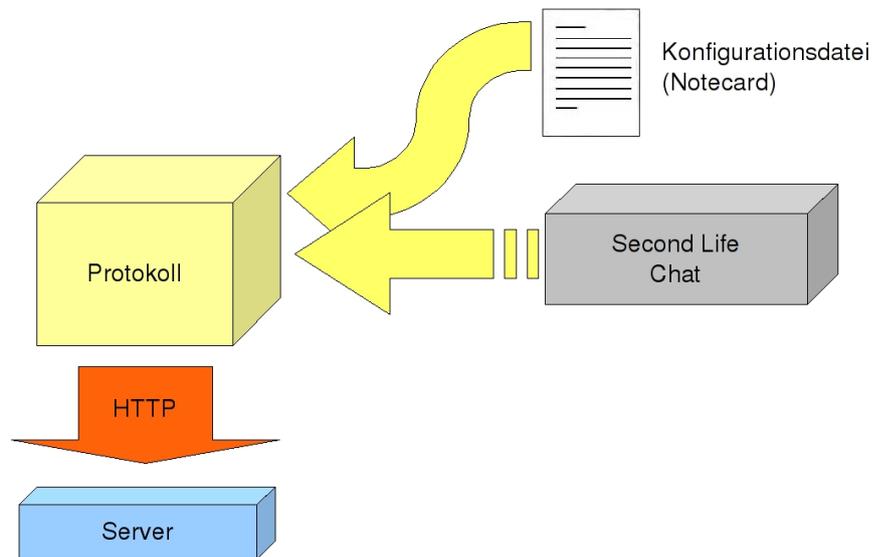


Abbildung 4.2: Protokollierung von Gesprächen

zer das Objekt und bekommt einen Dialog in der rechten oberen Ecke angezeigt. Hier muss nun auf „Protokoll“ geklickt werden, um die Protokollierung zu starten. Die zweite Möglichkeit ist das Starten über den Chat. Hierbei wird im Chatfenster *„protokoll“* eingegeben. Bei beiden Arten wird eine Warnung ausgegeben, dass die Protokollierung gestartet wurde. Analog funktioniert auch das Beenden des Protokolls.

Wenn die Protokollierung gestartet wurde, wechselt das Objekt die Farbe von grün auf rot. Dies ist ein weiteres Indiz für alle in der Nähe stehenden Avatare, dass die Protokollierung gestartet wurde. Beim Beenden wechselt das Objekt wieder die Farbe in grün und verschickt den entsprechenden Befehl an den Server, damit dieser weitere Schritte zur Speicherung unternehmen kann.

4.1.1.2 Verschicken von E-Mails

Die E-Mails werden, wie in Abbildung 4.3 ersichtlich, auf zwei Arten verschickt. Die erste ist das Verschicken von E-Mails über die Second Life Funktion [Mel08]. Der Befehl zum Verschicken einzelner E-Mails wird im Chat eingegeben. Der Befehl lautet *„/mail“*. Danach wird der Benutzer gebeten, den Sender, den Empfänger, den Betreff und den Inhalt einzugeben. Beim Sender und Empfänger wird

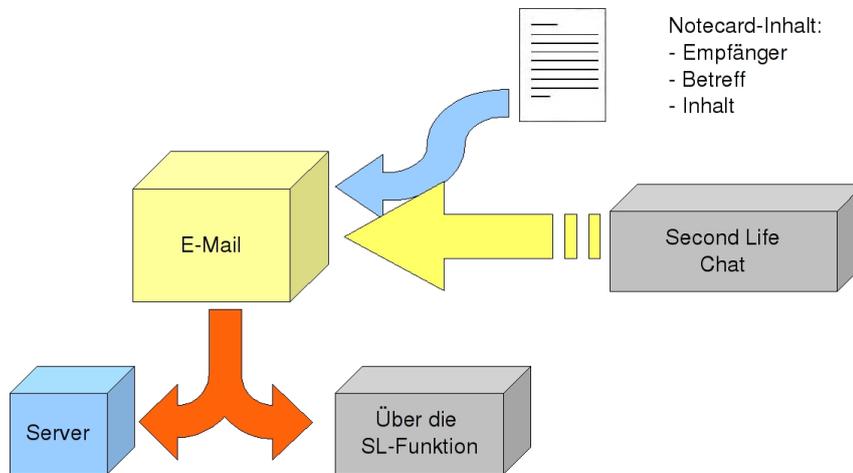


Abbildung 4.3: Verschicken von E-Mails

überprüft, ob die E-Mail Adresse die erforderlichen Zeichen beinhaltet. Erst nachdem eine gültige E-Mail-Adresse eingegeben wurde, wird der nächste Schritt eingeleitet. Dieser beinhaltet das Schreiben des Inhalts. Wenn der Inhalt eingegeben wurde, wird die E-Mail über die Second Life Funktion verschickt.

Das Verschicken von E-Mails an Gruppen wird einem Skript in Second Life nicht gestattet. Die Gründe wurden schon im Kapitel 3.1.3 erörtert, deshalb wird hier nicht näher darauf eingegangen. Für das Verschicken von E-Mails an mehrere Empfänger wird eine zweite Funktion implementiert. Diese beinhaltet das Versenden von E-Mails über einen entfernten Server. Wie schon beim Versenden der E-Mails über Second Life, wird auch hier zuerst eine Abfrage nach dem Absender, Empfänger, Betreff und Inhalt getätigt. Wenn eine Eingabe korrekt ausgeführt wurde, wird sie sofort an den Server geschickt. Dadurch wird gewährleistet, dass keine großen Datenmengen in Second Life gehalten werden und das Skript unnötig belasten. Das Verschicken von E-Mails an den Server wird im Detail in Kapitel 4.1.1.7 vorgestellt.

4.1.1.3 Anlegen eines Artikels in MediaWiki

Das Anlegen der Artikel in MediaWiki ist eine weitere Funktion, die Teil des Programms ist. Das Programm fragt hierbei nach dem Thema des Meetings und nach dem Artikelnamen. Der Artikelname ist der Name der Seite und gleichzeitig

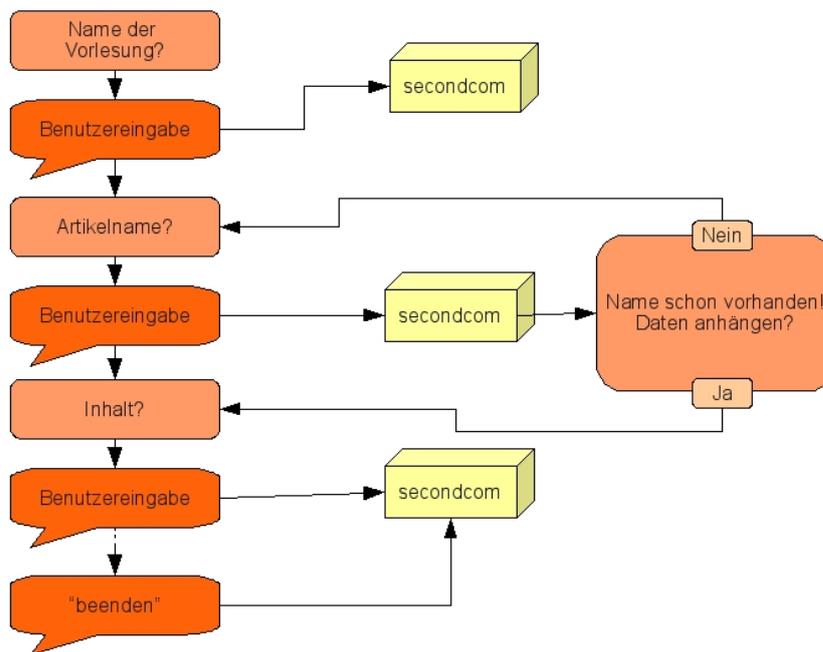


Abbildung 4.4: Grafische Darstellung des Anlegens der Artikel

auch die Überschrift der Seite. Bei der Überschrift findet eine Überprüfung statt, ob diese bereits existiert. Falls die Überschrift schon existiert, wird nachgefragt, ob der neue Inhalt angehängt werden soll oder ob eine neue Überschrift gewählt wird.

Danach wird der Inhalt eingegeben. Der Inhalt kann entweder in einzelnen Abschnitten oder als ein Block geschrieben werden. Damit nimmt der Autor schon in Second Life Einfluss auf die Formatierung der Seite. Außerdem muss die Formatierung der jeweiligen Web 2.0 Anwendung berücksichtigt werden. Im Fall dieses Prototyps handelt es sich bei der Anwendung um MediaWiki. Die Formatierung von MediaWiki wird im Kapitel 4.1.3.8 genauer erörtert. Nach der Eingabe des Inhalts, wird im Textchat *“beenden”* eingegeben. Erst zu diesem Zeitpunkt wird der Artikel erstellt bzw. erweitert.

Falls ein Fehler auftritt, wird keine neue Seite erstellt. Die möglichen Fehler sind:

1. Fehlender Name der Vorlesung
2. Fehlender Name des Artikels

3. Fehlender Inhalt.

Die ersten beiden Fehler treten bei der Überlastung oder anderen Problemen des Second Life Servers auf. Dann wird das Programm neu gestartet, weil die Konsistenz des Programms nicht mehr gewährleistet ist.

4.1.1.4 Öffnen eines Browsers

Diese Funktion wird als ein Zusatzfeature der Protokollierung angesehen. Wenn die Protokollierung beendet und die Wiki-Seite erstellt wurde, öffnet sich ein Browser und zeigt die entsprechende Seite an. Damit weiß die für die Protokollierung zuständige Person, dass die Protokollierung richtig durchgeführt wurde und kann bei einem Fehler die Protokollierung, falls nötig, manuell durchführen.

4.1.1.5 Zugriff auf die Kalenderfunktion

Die Funktion "Termine" schreibt einen neuen Termin in die entsprechende MediaWiki Seite. Dabei wird in das Textchat-Fenster das Datum, die Uhrzeit und eine kurze Beschreibung des Termins eingegeben.

Das Datum wird im folgenden Format eingegeben: Der Tag ist eine zweistellige Zahl. Falls nur eine Stelle eingegeben wird, wird der Zahl eine führende Null vorangestellt. Der Monat muss ausgeschrieben werden. Das Jahr hat die gleiche Konvention wie der Tag. Beispiel: „31. Dezember 09“

Bezüglich der Uhrzeit legt die Konvention fest, dass zwischen der Stunde und der Minute ein Doppelpunkt stehen muss. Beispiel: „00:00 “

Die Beschreibung des Termins kann beliebig eingegeben werden. Dabei ist zu beachten, dass der Text nicht zu lang wird, damit der Terminkalender übersichtlich bleibt. Wenn die Formatierung richtig eingegeben wurde, werden die Daten verschickt.

4.1.1.6 Verschicken von Notecards

Das Verschicken der Notecards ist eine weitere Funktion in diesem Programm. Diese Funktion beschränkt sich auf das Verschicken solcher Notecards, die sich im Objekt befinden. Das dient vor allem zur Sicherung der Konfigurationsdaten, die sich innerhalb des Objekts befinden.

Das Skript fragt zuerst nach dem Namen der Vorlesung. Danach wird der Name

der Notecard abgefragt. Falls die Notecard nicht existieren sollte, wird gebeten die Eingabe noch einmal zu wiederholen.

Wenn der Name korrekt abgefragt wurde, wird in MediaWiki überprüft, ob eine Wiki Seite mit dem Namen existiert. Im Fall, dass eine solche Seite existiert, wird der Name mit einem Zeitstempel versehen, damit nicht versehentlich wichtige Daten überschrieben werden. Danach wird die Notecard ausgelesen und verschickt.

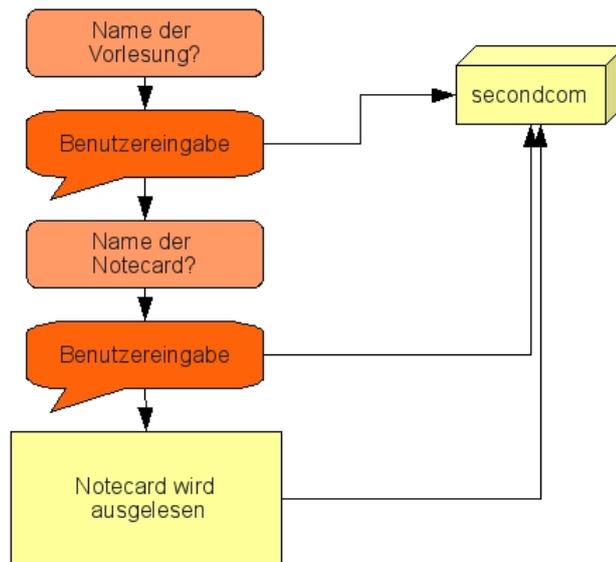


Abbildung 4.5: Grafische Darstellung des Verschickens einer Notecard

4.1.1.7 Kommunikation mit Second Life

Durch die Aufteilung des Programms in drei Bereiche, die örtlich voneinander getrennt sind, ist eine stabile und zuverlässige Kommunikation eines der wichtigsten Anforderungen an das Programm. Das Programm braucht eine Verbindung über die es die Daten vom Second Life Server verschicken kann. Hierfür werden zwei Verbindungen aufgebaut. Die erste ist die E-Mail Verbindung. Bei der zweiten handelt es sich um die HyperText Transfer Protocol (HTTP)-Verbindung.

Außerdem wird noch ein XMLRPC Kanal geöffnet, damit der HTTP-Supervisor unerwartete Meldungen von MediaWiki unverzüglich an den Second Life Server weiterleiten kann.

E-Mail Verbindung Die E-Mail Verbindung wird von der E-Mail Funktion verwendet. Diese sendet damit E-Mails an einzelne Personen. Für weitere Anwendungen eignet sich die E-Mail Funktion nicht, da sie alle zwanzig Sekunden nur einmal aufgerufen werden darf. Für alle anderen Verbindungen von Second Life aus, wird die HTTP Verbindung verwendet.

HTTP Verbindung Über die HTTP Verbindung wird der meiste Verkehr aus Second Life getätigt. Die einzelnen Pakete werden von hier an den HTTP Supervisor geschickt. Damit der Supervisor weiß, wie er die Pakete weiterleiten soll, wird im Prototyp jedem Paket der Befehl "mediawiki" vorangestellt. Dadurch erkennt der HTTP Supervisor, dass er die Pakete an den MediaWiki Bot senden muss. Die Befehle aus der Tabelle Table 4.1 werden direkt nach dem mediawiki Befehl geschrieben.

protocol	Protokollierung von Gesprächen
mail	E-Mail senden und archivieren
notecard	Verschicken von Notecards
article	Anlegen von neuen Wikiseiten
articleatt	Anhängen von Kapiteln an bestehende Wikiseiten
calendar	Kalenderfunktion

Tabelle 4.1: Die Befehle in einem Datenpaket, das an den HTTP Supervisor geschickt wird.

Die Second Life Funktion beinhaltet eine Einschränkung bezüglich der Länge der HTTP Nachricht. Deshalb wird das Programm nur eine Nachricht mit höchstens 200 Zeichen verschicken. Falls eine Nachricht mehr als 200 Zeichen beinhaltet, wird sie in mehrere Nachrichten aufgespalten und diese werden nacheinander versendet.

XMLRPC Verbindung Die XMLRPC Verbindung ist die einzige Möglichkeit, wie ein Server außerhalb von Second Life Nachrichten in Second Life verschicken kann. Dafür muss das Programm, das Nachrichten verschicken möchte, den richtigen Kanal kennen, über den das Skript erreichbar ist. Bei der Initialisierung der

XMLRPC Verbindung wird deswegen dem externen Programm mitgeteilt, über welchen Kanal das Skript zu erreichen ist.

4.1.1.8 Sensoren

Die Sensoren sind ein wichtiger Teil des Programms. Wie in Kapitel 4.1.1.1 erwähnt, ist die Protokollierung der Gespräche in Second Life nicht erlaubt. Deswegen müssen die Sicherheitsvorkehrungen bei der Protokollierung so gesetzt sein, dass sich kein Bewohner belästigt fühlt. Die Bewohner sollen rechtzeitig informiert werden, wenn sie in eine Zone gehen oder fliegen, die überwacht wird.

4.1.2 Teil 2: HTTP Supervisor

Der HTTP-Supervisor ist für das Weiterleiten der Daten von Second Life und Meldungen von MediaWiki zuständig. Außerdem überprüft das Programm bei der Initialisierung, ob die MediaWiki Adresse gültig ist und ob der Benutzername und das Passwort korrekt sind. Falls die Adresse ungültig sein sollte, sendet der HTTP-Supervisor einen Abbruchbefehl an die Second Life Funktion, die das Programm sofort beendet. Ansonsten sendet der HTTP-Supervisor die Meldungen unverändert von MediaWiki zu Second Life.

Die Daten in diesem Prototyp werden nur von Second Life zu MediaWiki weitergeleitet. Der HTTP-Supervisor, soll aber so frei konfiguriert werden, dass die Daten von Second Life zu beliebigen Web 2.0 Anwendungen weitergeleitet werden können.

Wenn die Daten ankommen, wird das Programm zunächst überprüfen, ob die Daten zum MediaWiki Bot geschickt werden sollen. Der HTTP Supervisor ist darauf ausgelegt, die Daten an verschiedene Bots weiterzuleiten. Zur Zeit ist aber nur der MediaWiki Bot implementiert. Sollen die Daten an eine andere Web 2.0 Anwendung verschickt werden, wird eine Fehlermeldung ausgegeben, die besagt, dass der zuständige Bot noch nicht implementiert wurde.

Datensicherheit und temporäre Dateien

Die Anforderungen sehen vor, eine absturzsichere Methode zu implementieren. Die Absturzsicherheit bezieht sich auf den plötzlichen Ausfall des Second Life Viewers, des HTTP Supervisors oder des MediaWiki Servers. Die Daten sollen auch bei einem plötzlichen Abbruch der Verbindung erhalten bleiben. Dafür werden die Daten zunächst in Dateien zwischengespeichert. Der HTTP Supervisor als von Web 2.0 unabhängige Komponente ist für diese Aufgabe zuständig. Die ankommenden Daten sind entweder schon formatiert oder werden erst vom Bot verarbeitet.

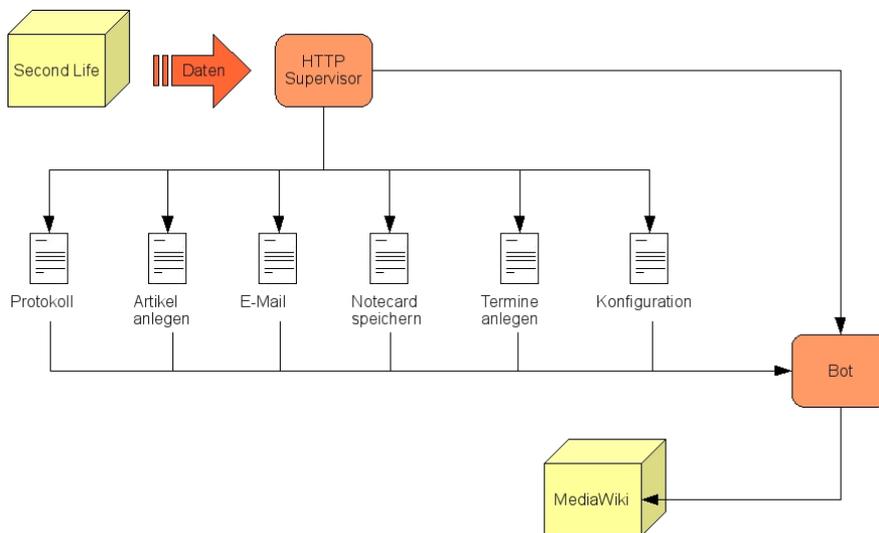


Abbildung 4.6: Speicherung in temporäre Dateien

Der HTTP Supervisor unterscheidet hierbei zwischen der Protokollierung, der Artikel Erstellung, der E-Mail Funktion, der Notecardspeicherung und der Kalenderfunktion. Bei der E-Mail Funktion überprüft der HTTP Supervisor zusätzlich, dass Absender, Empfänger, Betreff und Inhalt richtig voneinander getrennt sind. Außerdem überprüft er, ob alle erforderlichen Komponenten vorhanden sind, damit eine E-Mail verschickt werden kann.

Weiterhin erstellt der HTTP Supervisor eine Konfigurationsdatei. In dieser werden die Daten gespeichert, die für die weitere Verarbeitung durch den Bot wichtig sind. Für den MediaWiki Bot sind dabei folgende Daten erforderlich:

Address	Webadresse unter der auf MediaWiki zugegriffen werden kann
Username	Der Benutzername für die MediaWiki Anmeldung
Password	Das Passwort für die MediaWiki Anmeldung
XMLRPC Channel	Der Kanal unter dem das Second Life Skript erreicht werden kann

Tabelle 4.2: Erforderliche Daten für die Kommunikation mit dem MediaWiki und dem Second Life Server

Es ist wichtig, dass die Konfigurationsdatei sofort nach der Verarbeitung durch den Bot gelöscht wird, weil die Daten in dieser Datei im Klartext gespeichert werden. Dadurch wäre es einem Angreifer möglich, die Benutzerdaten zu bekommen und dadurch die Daten zu verändern. Deswegen löscht der Bot die Datei, nachdem er die Daten extrahiert hat. Dadurch wird diese Sicherheitslücke auf ein Minimum beschränkt. Alle anderen Dateien werden erst gelöscht, wenn der Bot eine Bestätigung bekommen hat, dass die Daten auch richtig in MediaWiki eingepflegt wurden.

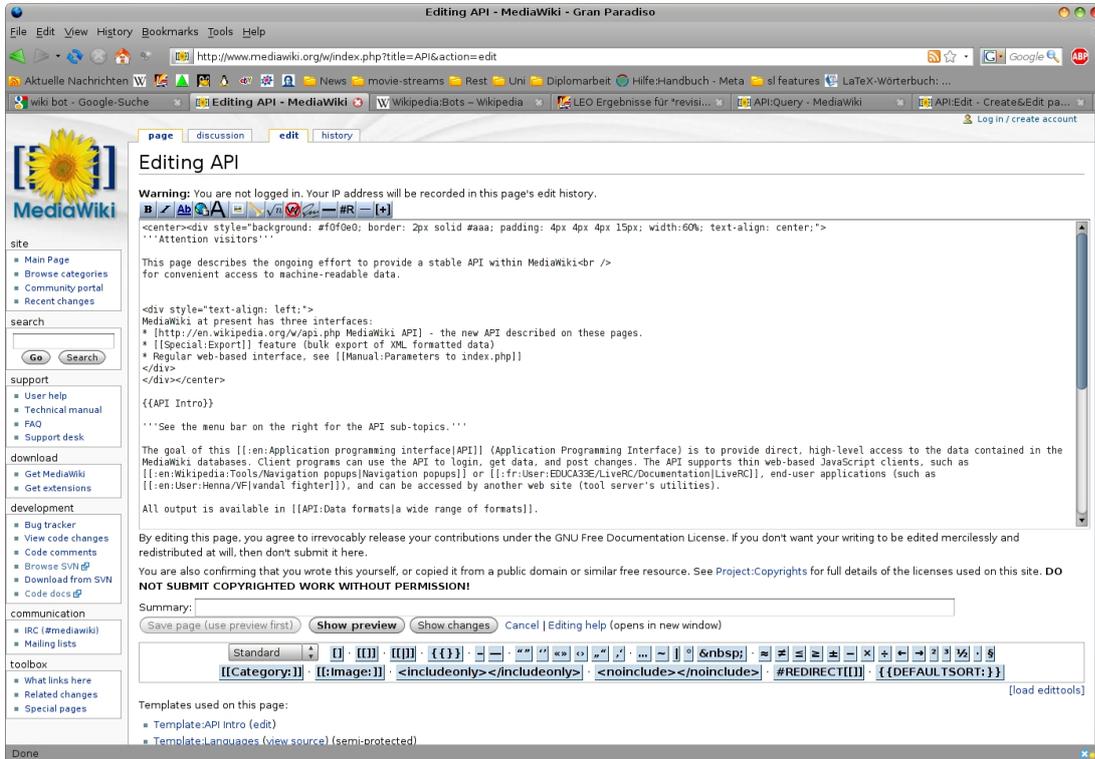


Abbildung 4.7: Oberfläche zum Bearbeiten des Inhalts

4.1.3 Teil 3: MediaWiki

4.1.3.1 Automatischer Zugriff auf die MediaWiki-Inhalte

Der Zugriff auf die MediaWiki-Inhalte geschieht im Normalfall über den Browser. MediaWiki stellt die Oberfläche zur Verfügung, über die der Nutzer die Inhalte einsehen und bearbeiten kann. Die Abbildung 4.7 zeigt die Oberfläche zum Bearbeiten des Inhalts.

Der automatische Zugriff durch ein Skript erfolgt über das Application Programming Interface (API). Die API gibt dem Programm direkten Zugang zu den Daten, die sich in den MediaWiki Datenbanken befinden.

Die Programme können die API zum Anmelden, zum Auslesen oder zum Bearbeiten der Inhalte benutzen. Die API unterstützt webbasierte Java-Clients und kann von jeder Webseite erreicht werden. Der Zugriff auf die API Funktion erfolgt über die URL des MediaWiki Servers mit der Endung `api.php`. Dabei können verschie-



Abbildung 4.8: Die Anmeldung in MediaWiki mit der anschließenden Antwort.

dene Aktionen gestartet werden. Die erste Aktion, auf die der Nutzer zugreift, ist `action=login`. Diese Aktion hilft dem MediaWiki Server, den Benutzer zu erkennen, damit der Server die nachfolgenden Schritte dem Nutzer zuweisen kann. Außerdem wird bei der Anmeldung ein Cookie erstellt, der bei jedem weiteren Request mitgesendet werden muss. Die Anmeldedaten müssen im HTTP-POST Format gesendet werden. Ein gesendeter GET-Request führt zu einer Fehlermeldung.

Die Parameter, die bei einer Anmeldung übergeben werden, sind in der folgenden Abbildung aufgezeigt:

<code>action=login</code>	Die Aktion zum Anmelden des Benutzers
<code>lgname</code>	Benutzername
<code>lgpassword</code>	Benutzerpasswort

Tabelle 4.3: Parameter der Anmeldung

Dabei können verschiedene Fehler auftreten. Wenn die API über den Browser ausgeführt wird, werden die Fehler im Browserfenster angezeigt. Die möglichen Fehler werden in der folgenden Tabelle aufgelistet:

NoName	Der Benutzername wurde nicht übergeben
Illegal	Ein unzulässiger Benutzername wurde übergeben
NotExists	Der Benutzername wurde noch nicht angelegt
EmptyPass	Das Passwort wurde nicht übergeben
WrongPass	Das Passwort ist falsch
WrongPluginPass	Die gleiche Fehlermeldung wie WrongPass. Dieses Mal hatte aber das Authentifizierungsplug-in das Passwort abgelehnt.
CreateBlocked	Das Wiki hat versucht automatisch ein neues Konto zu errichten, aber die IP Adresse des Benutzers ist nicht zulässig.
Throttled	Der Benutzer hat zu oft versucht sich einzuloggen

Tabelle 4.4: Fehlermeldungen bei der Anmeldung

Die nächste Funktion ist das Abmelden des Benutzers im MediaWiki. Die Abmeldeprozedur löscht alle Token und Cookies, die bei der Anmeldung erstellt wurden. Der Request sieht folgendermaßen aus:



Abbildung 4.9: Die Abmeldung in MediaWiki mit anschließender Antwort.

Hierbei können keine Fehler auftreten. Deshalb gibt es auch keine Fehlerausgabe. Neben dem Anmelde- und dem Abmeldevorgang kann der Benutzer auch verschiedene Anfragen (Queries) durchführen. Mit Hilfe der Queries kann der Benutzer verschiedene Informationen erhalten. Das Query-Modul hat viele Untermodule, die verschiedene Funktionen enthalten. Zur Zeit existieren drei Arten der Query-Module:

1. Meta-Informationen über das Wiki und den angemeldeten Benutzer

2. Seiteneigenschaften
3. Liste der Seiten, die verschiedene Kriterien erfüllen.

Eine oft benutzte Methode ist das Abfragen des edit-Tokens. Das Token ist für alle Seiten gleich, ändert sich aber bei jeder Anmeldung. Deswegen muss sich der Benutzer zuerst anmelden und erst dann das Token erfragen. Gleichzeitig kann es ratsam sein, den Zeitstempel der letzten Änderung anzufordern, damit Kollisionen bei der Bearbeitung der Seite vermieden werden. Das Token und der Zeitstempel können auf folgende Weise beantragt werden:



```

Obtaining an edit token
api.php ? action=query & prop=info|revisions & intoken=edit & titles=Main%20Page

<?xml version="1.0" encoding="utf-8"?>
<api>
  <query>
    <pages>
      <page pageid="15580374" ns="0" title="Main Page" touched="2008-03-27T21:15:39Z" lastrevid="199631190" counter="0" length="47" />
    </pages>
    <revisions>
      <rev revid="199631190" user="David Levy" timestamp="2008-03-20T17:26:39Z" comment="Have you tested it in every major browser" />
    </revisions>
  </query>
</api>

```

Abbildung 4.10: Query zum Erfragen des *edit*-Tokens und des Zeitstempels.

Wenn das Token und der Zeitstempel eingeholt wurden, kann mit der Bearbeitung des Textes angefangen werden.

Die Aktion zum Bearbeiten von Inhalten nennt sich *action=edit*. Vor dieser Aktion muss die Anmelde- und die Edittoken-Aktion beendet sein, sonst wird eine Fehlermeldung ausgegeben.



```

Adding a new section to en:Talk:Main Page
api.php ? action=edit & title=Talk:Main_Page & section=new & summary=Hello%20World & text=Hello%20everyone! & watch & basetimestamp=2008-03-20T17:26:39Z & token=cecded1f35005d22904a35cc7b736e18%2B%5C

<?xml version="1.0" encoding="utf-8"?>
<api>
  <edit result="Success" pageid="12" title="Talk:Main Page" oldrevid="465" newrevid="471" />
</api>

```

Abbildung 4.11: Aktion zum Bearbeiten des Inhalts mit der anschließenden Antwort.

Diese Aktion braucht mehr Parameter als die vorangegangenen. Die einzelnen Parameter werden in der folgenden Tabelle genauer erläutert.

title	Name der Seite, die bearbeitet werden soll.
section	Das Kapitel, das bearbeitet werden soll. 0 = erstes Kapitel, new = neues Kapitel.
text	Der Inhalt der neuen Seite oder des neuen Kapitels.
token	Edit Token.
summary	Zusammenfassung. Wenn section=new benutzt wird, ist das die Überschrift des Kapitels.
minor	Wenn gesetzt, wurde das Bearbeiten als geringfügig eingestuft.
notminor	Wenn gesetzt, wurde das Bearbeiten als nicht geringfügig eingestuft.
bot	Die Bearbeitung wurde vom Bot durchgeführt.
basetimestamp	Zeitstempel der letzten Änderung. Wird benutzt, um Bearbeitungskonflikte zu ermitteln.
recreate	Wenn gesetzt, wird die Seite neu erstellt. Die Fehlermeldung wird ignoriert.
createonly	Wenn gesetzt, wird ein Fehler ausgegeben, falls die Seite bereits existiert.
nocreate	Ein Fehler wird ausgegeben, wenn die Seite noch nicht existiert.
watch	Seite wird zu der Beobachtungsliste hinzugefügt.
unwatch	Seite wird von der Beobachtungsliste entfernt.
md5	MD5-Hash des <i>text</i> Parameters. Wenn die Hashwerte nicht übereinstimmen, wird die Seite nicht verändert.

Tabelle 4.5: Parameter zum Bearbeiten einer WikiSeite mit API.

Bei der Erstellung einer WikiSeite können auch Fehler entstehen. Die Fehler werden in der Tabelle 4.6 aufgelistet.

notitle	Die Überschrift wurde nicht gesetzt.
notext	Inhalt wurde nicht übergeben.
notoken	Das Token wurde nicht gesetzt.
invalidsection	Der <i>section</i> Parameter muss ganzzahlig oder <i>new</i> sein.
protectedtitle	Die Überschrift ist geschützt und kann nicht erzeugt werden.
cantcreate	Der Benutzer hat keine Erlaubnis neue Seiten zu erzeugen.
cantcreate-anon	Anonyme Benutzer dürfen keine neuen Seiten erzeugen.
articleexists	Die Wiki Seite existiert schon.
noimageredirect	Der Benutzer hat nicht die Erlaubnis die Bilder umzuadressieren.
spamdetected	Die Bearbeitung wurde abgelehnt, weil sie ein Spam Fragment enthielt.
filtered	Ein Filter hat die Bearbeitung abgelehnt.
contenttoobig	Der Inhalt übersteigt die zulässige Größe in Bytes.
noedit-anon	Anonyme Benutzer dürfen die Seite nicht bearbeiten.
pagedeleted	Die Seite wurde seit dem Erzeugen des Zeitstempels gelöscht.
emptypage	Das Erstellen einer leeren Seite ist nicht erlaubt.
emptynewsection	Das Erstellen neuer, leerer Kapitel ist nicht erlaubt.
editconflict	Beim Bearbeiten wurde eine Kollision festgestellt.

Tabelle 4.6: Fehlermeldungen, die beim Bearbeiten der Seite auftreten können

In diesem Unterkapitel wurden die wichtigsten Aktionen und ihre Benutzung erklärt. Im nächsten Unterkapitel werden die Wiki Bots vorgestellt.

4.1.3.2 Wiki-Bots

Ein Bot kann auf verschiedene Arten die Seiten anfordern. Die MediaWiki API wurde schon im Kapitel 4.1.3.1 erörtert. Die zweite Art nennt sich *Screen Scra-*

ping. Beim *Screen Scraping* wird eine Seite angefordert. Dabei erhält der Bot den Quellcode der Seite, den er bearbeiten kann. Diese Methode ist mittlerweile veraltet. Das Problem bei diesem Ansatz ist eine mögliche Änderung des Quelltextes der Wikiseite, der beim Erscheinen einer neuen Version vorkommen könnte. Dabei müsste der Bot, der *Screen Scraping* benutzt, umständlich angepasst werden oder im schlimmsten Fall neu geschrieben werden. Deswegen wird diese Art der Bearbeitung nicht mehr angewandt.

Die nächste Methode ist der *Special:Export*. Dabei wird der Inhalt einer Wikiseite im eXtensible Markup Language (XML)-Format ausgegeben. Dieses Feature gehört zur normalen MediaWiki Installation und existiert dementsprechend auf allen MediaWiki Servern.

Die letzte Möglichkeit beruht darauf, den Quellcode in Rohdaten zu erhalten. Dabei wird nur der Inhalt der Wikiseite ausgegeben, der verändert werden kann. Beim *Screen Scraping* bekommt der Bot die gesamte *Edit*-Seite im Quellcode und muss sich den Text der Wikiseite anhand der regulären Ausdrücke ausschneiden. Der Bot muss sich immer anmelden, bevor er eine Seite bearbeiten darf. Wenn ein Bot eine Seite nur auslesen möchte, dann ist eine Anmeldung zwar nicht erforderlich, aber ratsam. Denn nur im Falle einer Anmeldung können die einzelnen Schritte des Bots protokolliert werden. Die Anmeldeprozedur wurde schon im Kapitel 4.1.3.1 genau erörtert und gilt sowohl für Bots als auch für die menschlichen Benutzer, wenn sie sich über die API anmelden.

Nach der Anmeldung kann der Bot die Wikiseite bearbeiten. Die Bearbeitung erfordert zwei HTTP Anfragen. Die erste Anfrage ist die nach dem *Edittoken*. Die andere bezieht sich auf die Aktion zum Bearbeiten der Wikiseite. Diese Aktionen wurden ebenfalls schon im Kapitel 4.1.3.1 beschrieben.

Ein Bot muss auch mit Fehlern, die ein Wiki ausgeben kann, umgehen. Der am häufigsten auftretende Fehler, ist der Konflikt beim Bearbeiten, der im MediaWiki *editconflict* genannt wird. Beim *editconflict* gibt es eine Kollision mit einem anderen Benutzer, der ebenfalls diese Seite bearbeiten wollte oder gerade bearbeitet. Der Bot muss in diesem Fall überprüfen, wie die Veränderungen aussehen und ob es noch notwendig ist, die Seite weiter zu verändern.

In dieser Arbeit hat der Bot verschiedene Aufgaben, die er verrichtet. Diese Aufgaben werden nun vorgestellt.

4.1.3.3 Speicherung der Protokollierung

Bei der Speicherung der Protokollierung werden die erforderlichen Daten aus der temporären Datei in MediaWiki geschrieben. Diese Vorgehensweise ist notwendig, weil es sonst zu Problemen kommen kann, wenn die Daten nacheinander in MediaWiki geschrieben werden.⁶ Außerdem wären die Daten bei einem plötzlichen Serverausfall an mehreren Stellen gespeichert, was eine Wiederherstellung erschweren würde.

Da die Daten schon in Second Life formatiert wurden, muss dies nicht erneut geschehen. Die Datei wird unter dem Verzeichnis *Veranstaltung/Protokolle* in MediaWiki abgelegt. Der Name des Verzeichnisses *Veranstaltung* bezeichnet den Namen der Veranstaltung, unter der das protokollierte Meeting stattgefunden hat. Der Dateiname setzt sich aus dem Thema des Meetings und einem Zeitstempel zusammen. Der Zeitstempel dient der Eindeutigkeit, weil der Name bei der Erstellung der Seite nicht geprüft wird. Falls eine Wikiseite unter dem gleichen Namen auf dem Server existiert, wird sie in diesem Schritt überschrieben. Es ist in der Praxis aber nicht möglich, dass so eine Kollision stattfindet. Der Zeitstempel setzt sich aus dem Datum und der Uhrzeit zusammen. Die Uhrzeit selbst besteht aus Stunden und Minuten. Der Zeitstempel bezieht sich also auf den Zeitpunkt der Erstellung der Wikiseite durch den Bot. Dadurch lässt sich nicht vorhersagen, wie der genaue Titel der Seite lauten wird. Außerdem sind die Protokolle dadurch zeitlich besser zuzuordnen.

Der Inhalt wird unverändert ausgelesen und in die Wikiseite eingepflegt. Die betreffende temporäre Datei wird anschließend gelöscht.

4.1.3.4 Anlegen eines Artikels in MediaWiki

Das Anlegen eines Artikels in MediaWiki ähnelt in einigen Punkten dem Anlegen einer Wikiseite, die für die Protokollierung zuständig ist. Der größte Unterschied ist, dass der Bot prüfen muss, ob die betreffende Seite schon existiert. Die vorhandene Seite darf in diesem Fall nicht gelöscht werden. Der Bot muss eine Anfrage an das Second Life Skript und dadurch also auch an den Benutzer senden, wie weiter verfahren werden soll. Danach schaltet sich der Bot ab und wartet bis er wieder vom HTTP Supervisor angefordert wird.

⁶Diese Probleme werden genauer im Kapitel 5.2 erörtert.

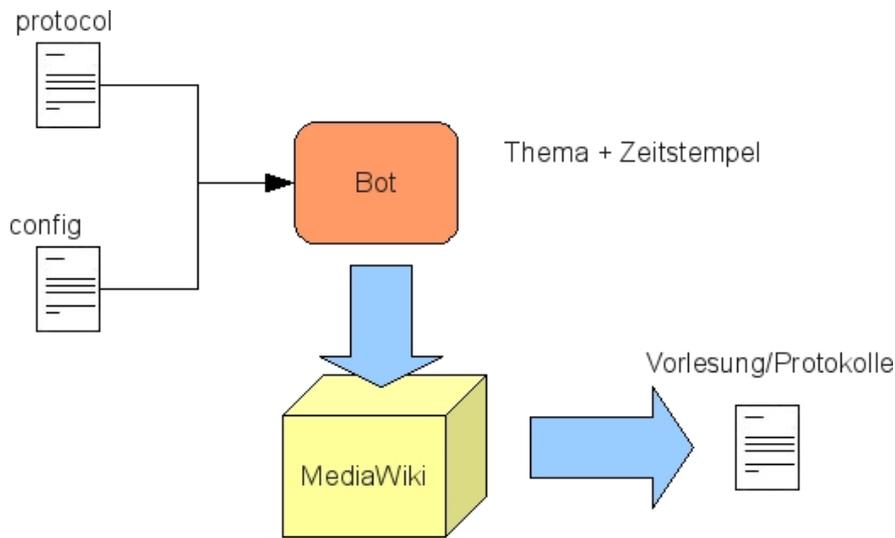


Abbildung 4.12: Speicherung der Protokollierung in MediaWiki

Der Benutzer hat nun zwei Möglichkeiten zur Auswahl. Entweder kann er bestimmen, dass der neue Inhalt an den alten angehängt wird oder er kann einen neuen Namen für die Webseite wählen.

Danach wird der Bot wieder gestartet. Wenn die Auswahl auf das Anhängen des Inhalts gefallen ist, wird der Inhalt angehängt und die temporäre Datei gelöscht. Falls eine neue Seite angelegt werden soll, muss wieder überprüft werden, ob die Wikiseite mit dem Namen existiert und der Vorgang wiederholt sich.

4.1.3.5 Speicherung einer Notecard

Der Unterschied zum Anlegen einer Wikiseite für Notecards zu den anderen Speichermethoden liegt in der Überschrift. Da es sich beim Speichern einer Notecard meist um die Sicherung der Konfigurationsdaten handelt, wird es mit einer hohen Wahrscheinlichkeit zu Kollisionen bei der Namensgebung kommen. Falls es nicht zu einer Kollision kommt, wird der Name unverändert gespeichert. Bei einer Kollision wird dem Namen der in Kapitel 4.1.3.3 beschriebene Zeitstempel angehängt. Die Notecards werden im Verzeichnis Veranstaltung/Notecards gesichert.

4.1.3.6 E-Mail Versand und Archivierung

Bei der Verarbeitung der E-Mails werden zwei verschiedene Funktionen ausgeführt: E-Mails senden und E-Mails archivieren.

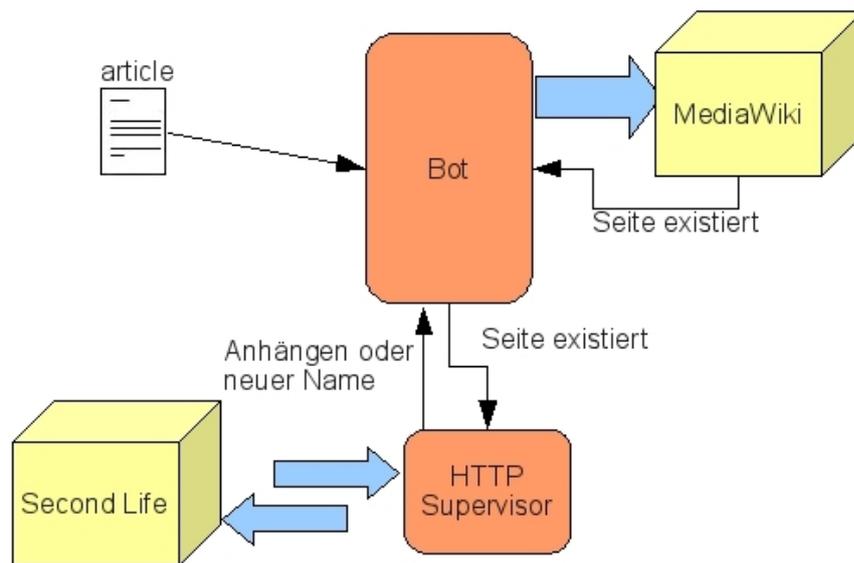


Abbildung 4.13: Anlegen eines neuen Artikels oder Anhängen des Inhalts

Beim Senden der E-Mails werden die Daten übertragen. Diese werden nicht überprüft, weil diese Überprüfung schon in Second Life und noch einmal im HTTP Supervisor stattgefunden hat.⁷

Das Archivieren von E-Mails ist die wichtigste Neuerung in dieser Funktion. Beim Senden der E-Mails werden diese nämlich automatisch auch in MediaWiki eingepflegt. Dabei werden alle E-Mails im Ordner Vorlesung/E-Mails abgelegt. Jede E-Mail wird als eine einzelne Wikiseite in MediaWiki geschrieben. Die Wikiseite *"E-Mail Archiv"* beinhaltet das Inhaltsverzeichnis, welches den Überblick über die E-Mails erleichtert.

4.1.3.7 Kalenderfunktion

In der Kalenderfunktion werden die Termine einer Vorlesung festgehalten. Die Termine werden chronologisch angeordnet. Dabei werden die Monate als Sektionen angelegt, um Übersichtlichkeit zu gewährleisten.

⁷Vergleiche dazu Kapitel 4.1.1.2 und Kapitel 4.1.2

4.1.3.8 Formatierung in MediaWiki

4.1.3.8.1 Namensgebung Die Namen einer Seite in MediaWiki dürfen keine Sonderzeichen enthalten. Sonderzeichen sind [,], {, }, |, #, < oder >. Diese Sonderzeichen finden bereits in MediaWiki Verwendung und würden die Ausgabe verfälschen. Die erlaubten Zeichen sind: A-Z, 0-9, %, , ?, :, , &, \$, _, -, +, !, *, (,) und ,.

Dabei sollten die Namensräume nicht als Namen für Seiten benutzt werden. Die Namen für die Namensräume sind reserviert und die doppelte Besetzung würde zu Fehlern in der Funktionsweise von MediaWiki führen. Die Namen dürfen außerdem nicht denen der Pseudo-Namensräume gleichen. Pseudo-Namensräume sind *Special:* oder *Media:*. Diese Namensräume existieren nicht. Sie bilden nur Verknüpfungen zu den oft besuchten Seiten im Wiki. *Media:* ist z.B. ein Pseudo-Namensraum für Bilder und andere Medien.

4.1.3.8.2 Formatierung einer Seite Die Formatierung des Textes wurde in MediaWiki im Vergleich zum HTML stark vereinfacht. Diese besteht aus Sternen, einfachen Anführungszeichen und Gleichheitszeichen. Die Formatierungszeichen können verschiedene Bedeutungen haben. Wenn z.B. ein Wort kursiv geschrieben werden soll, wird das Wort von zwei einfachen Anführungsstrichen auf jeder Seite umrandet ("kursiv"). Es ist hilfreich zu wissen, dass ein Artikel mit vier oder mehr Überschriften automatisch mit einer Inhaltsangabe ausgestattet wird. Desweiteren müssen die Absätze explizit angegeben werden, weil MediaWiki die normalen Zeilenumbrüche ignoriert. Wenn ein neuer Absatz gestartet werden soll, muss eine Zeile frei gelassen werden. Eine andere Möglichkeit sind die HTML-Tags
 oder
.

In MediaWiki sind weiter HTML-Tags erlaubt. Einige Beispiele von funktionierenden Tags sind <code>, <div>, und . Wenn alle Benutzer als vertrauenswürdig erachtet worden sind, kann auch das gesamte HTML frei geschaltet werden. [Rah07]

4.2 Kommunikationsprotokolle

4.2.1 HTTP

Das Protokoll, das in diesem Prototyp benutzt wird, ist HTTP. [Dro05] Deshalb wird hier eine kurze Einführung in das HTTP Protokoll gegeben. HTTP ist ein zustandsloses, generisches Anwendungsprotokoll. Es basiert auf dem TCP/IP Protokoll. Ein HTTP Protokoll besteht aus zwei Programmen. Das erste ist das Clientprogramm. Beim zweiten handelt es sich um das Serverprogramm. Diese beiden Programme kommunizieren untereinander mittels Nachrichtenaustausch. Diesen Nachrichtenaustausch nennt man das Request/Response Interaktionsmodell. In diesem Modell fordert der Client Daten vom Server durch Versenden einer HTTP-Request-Nachricht an. Der Server antwortet mit einem HTTP Response. Die Version 1.1 des HTTP Protokolls kennt 8 verschiedene Request Methoden. Diese sind CONNECT, DELETE, GET, HEAD, OPTIONS, POST, PUT und TRACE. Es würde den Rahmen dieser Arbeit sprengen, diese zu erklären. Deswegen wird hier nur die POST Request Methode kurz erklärt. Die anderen können in [Dro05] nachgelesen werden.

Beim POST-Request handelt es sich um eine Methode, mit welcher der Client Informationen an ein Objekt übermitteln kann. Der Unterschied zwischen dem GET- und dem POST-Request liegt in der Übermittlung der Parameter. Die Parameter des Requests werden im Message Body übermittelt und nicht wie beim GET-Request in der URL. Dadurch können auch Passwörter sicher übertragen werden, weil sie nicht einfach ausgelesen werden können.

Jeder Request fordert eine Response Nachricht. Die Response Nachricht selbst besteht aus drei Teilen:

1. Status-Line
2. Sechs Header
3. Entity-Body, das die Daten enthält

Der *Entity Body* nimmt im Prototyp eine wichtige Stellung ein. Über ihn werden die Meldungen von MediaWiki an den HTTP Supervisor und dann weiter an das Second Life Skript übermittelt. Damit wird gewährleistet, dass die drei Komponenten immer informiert sind.

Neben dem *Entity Body*, ist die *Status-Line* für den Prototyp sehr wichtig. Die *Status-Line* selbst, besteht wieder aus drei Teilen:

1. Protokoll-Versionsfeld
2. Status-Code
3. Status-Message

Der Prototyp legt großen Wert auf die Status Codes. Die Status Codes liefern dem Second Life Skript Meldungen über Erreichbarkeit der beiden anderen Server des Systems. Auf diese Meldungen muss er entsprechend reagieren.

Die möglichen Statuscodes sind in der Tabelle 4.7 aufgelistet:

2xx	Erfolg	Die Daten wurden erfolgreich übermittelt.
3xx	Umleitung	Das Objekt, der Ort des angesprochenen Objekts hat sich verändert.
4xx	Client Error	Aufgefordertes Objekt ist nicht vorhanden, unautorisierter Zugriff, usw.
5xx	Server Error	Auf dem Server ist ein interner Fehler aufgetreten.

Tabelle 4.7: Die Familien der Statuscodes in dem HTTP Anwendungsprotokoll

Folgende Meldungen sind für den Prototyp wichtig:

200	OK	Request war erfolgreich
400	Bad Request	Server hat den Request nicht verstanden
404	Not Found	Server hat die erforderliche Seite nicht gefunden
408	Request Timeout	Request wurde nicht rechtzeitig beantwortet
501	Internal Server Error	Unerwarteter Server Fehler. Request konnte nicht erfüllt werden
503	Service Unavailable	Server kann zur Zeit den Request nicht verarbeiten
505	HTTP Version Not Supported	Server unterstützt die Version des HTTP Requests nicht

Tabelle 4.8: Meldungen, auf die der Prototyp reagieren muss

4.2.2 XMLRPC

XMLRPC ist ein entfernter Prozeduraufruf, welcher über das Internet arbeitet. Die XMLRPC-Nachricht ist ein HTTP-POST Request. Die gesamte Kommunikation vom Client zum Server und zurück wird in XML kodiert.

Die Prozedurparameter können dabei Nummern, Strings, Daten usw sein. Außerdem kann es sich bei den Parametern um Listen handeln.⁸

4.3 Client/Server Modell

Diese Arbeit beruht auf dem Client/Server Modell. Deswegen wird das Modell hier kurz erläutert. Das Client/Server Modell ist ein Software-Architekturmodell. Das Grundschema, auf welchen das Modell basiert, beschreibt die Kommunikation, die zwischen dem Client und dem Server stattfindet. Der Client startet die Kommunikation mit einer Anfrage. Er formuliert die Aufträge und schickt sie an einen Dienstanbieter, den Server. Dieser stellt bestimmte Dienste zur Verfügung und nimmt die dazugehörigen Aufträge entgegen. Der Server bearbeitet nur die

⁸vgl. <http://www.xmlrpc.com/spec>

```
POST /RPC2 HTTP/1.0
User-Agent: Frontier/5.1.2 (WinNT)
Host: betty.userland.com
Content-Type: text/xml
Content-length: 181
```

```
<?xml version="1.0"?>
<methodCall>
  <methodName>examples.getStateName</methodName>
  <params>
    <param>
      <value><i4>41</i4></value>
    </param>
  </params>
</methodCall>
```

Abbildung 4.14: Ein Beispielcode für ein XMLRPC Aufruf

Dienste, die er Verfügung stellt. Andere Anfragen werden entweder weitergeleitet, ignoriert oder der Server sendet eine Fehlermeldung zurück.

Im Client/Server Modell sind die Rollen der Beteiligten und die zeitliche Abfolge der Interaktionsschritte genau festgelegt.

Ein Client kann im Laufe der Verarbeitung auf mehrere Server zugreifen; der Server kann auch verschiedene Clients bedienen. Ein Server kann auch ein Client werden, falls er im Laufe der Bearbeitung der Anfrage auf einen anderen Server zugreifen muss, wie in der Abbildung 4.16 ersichtlich ist.

Im Client/Server Modell werden alltägliche Aufgaben wie z.B. der E-Mail Verkehr oder der Webzugriff bearbeitet. Es kann aber auch sein, dass spezifische Aufgaben, wie z.B. Routing oder Paketfilterung in diesem Modell vorgenommen werden.

4.3.1 Kommunikation im Client/Server Modell

Die Kommunikationsinfrastruktur wird nicht explizit vorgegeben. Das Modell beschreibt nur eine logische Verteilung zwischen beiden Systemen. Der Client und der Server können sowohl lokal innerhalb eines Rechners arbeiten, die beiden Komponenten können aber auch über das Internet miteinander kommunizieren.

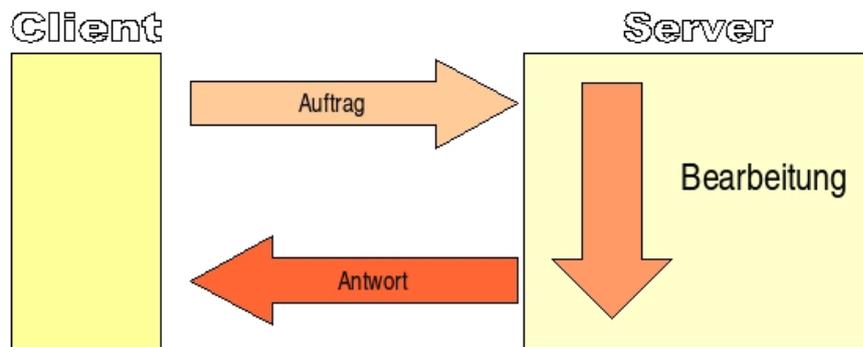


Abbildung 4.15: Client-Server Modell

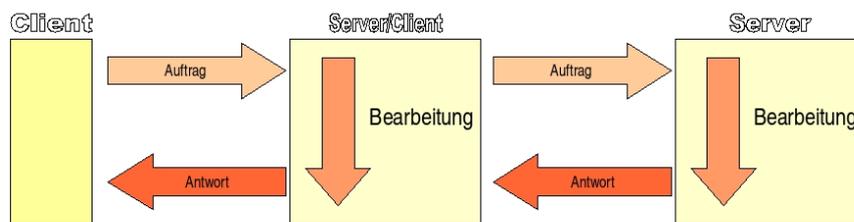


Abbildung 4.16: Client-Server Modell mit mehreren Servern

[Gei95]

Client/Server Systeme kennen zwei Arten der Kooperation:

1. Nachrichtenaustausch (Message Passing) und
2. Entfernten Prozeduraufruf (Remote Procedure Call (RPC)).

Wegen der besonderen Kommunikationseinschränkungen von Second Life ⁹ werden sowohl der Nachrichtenaustausch als auch RPC verwendet.

Beim Nachrichtenaustausch unterscheiden wir zwei verschiedene Nachrichten:

1. Senden (Send) und
2. Empfangen (Receive)

Der Nachrichtenaustausch kann vereinfacht so dargestellt werden:

⁹vgl. Kapitel 3.1.3

1. Der Client sendet die Nachrichten an den Server. Dabei muss er den zu erbringenden Dienst spezifizieren.
2. Der Server nimmt die Nachricht entgegen und führt den Dienst aus.
3. Wenn kein Fehler aufgetreten ist, wird das Ergebnis in einer Nachricht an den Client zurückgeschickt. Falls ein Fehler aufgetreten ist, wird nur der Fehlerindikator zurückgeschickt.
4. Der Client nimmt das Ergebnis entgegen und setzt seine Verarbeitung fort.

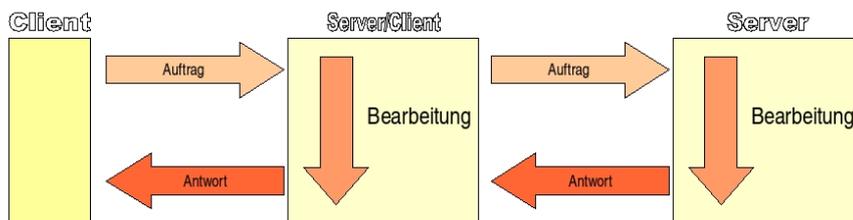


Abbildung 4.17: Der Nachrichtenaustausch in einer Client-Server-Server Umgebung

Der entfernte Prozeduraufruf ist ein anderer Ansatz zur Realisierung verteilter Systeme. Der Ablauf des RPC geschieht auf folgende Art und Weise:

1. Ein Anwendungsprogramm will eine Prozedur auf einem anderen Rechner starten.
2. Das Lokale System kodiert den Aufruf mit dem Client *Stub*, der den Server kennt.
3. Der Aufruf besteht aus:
 - a) Spezifikation der aufgerufenen Prozedur
 - b) Adresse des Zielrechners
 - c) eindeutige Aufrufkennung und
 - d) dem aktuellen Parameter
4. Der Aufruf wird übertragen

5. Übergabe an den Server Stub
 - Dekodierung des Aufrufs und der Parameter
 - Feststellen der Aufruferadresse
 - Aufruf der Prozedur und
 - Ausführung der Prozedur
6. Nach der Ausführung werden die Rückgabeparameter und die Aufrufkennung kodiert und die Antwort wird zurückgeschickt.
7. Die Stub Komponente dekodiert die Antwort, übergibt das Ergebnis und deblockiert das Programm.

In dieser Arbeit wird ein hybrides Modell verwendet. Die Gründe hierfür sind folgende: Second Life erlaubt eine Verbindung mittels HTTP-Requests von Second Life aus. Der Aufruf von einem entfernten Server zu Second Life kann nur mittels XMLRPC geschehen. Deswegen wird die Anfrage von Second Life mittels Nachrichtenaustausch und die Anfrage von einem anderen Server zu Second Life mittels RPC gelöst.

4.4 Bedeutung für die Veranstaltung

Die Bedeutung für die Veranstaltung "Einführung in das IT-Projektmanagement" lässt sich nur dann richtig bewerten, wenn betrachtet wird, wie die Durchführung der Arbeit ohne das Programm aussieht. Ohne das Programm sind viele Schritte notwendig, um eine Protokollierung durchzuführen. Das Protokoll muss aus Second Life in die Wiki Seite kopiert werden. Diese Wiki Seite muss dann manuell erstellt werden. Falls es einen Ausfall gibt, sind die Daten unwiederbringlich verloren. Das Gleiche gilt, wenn der Moderator vergisst, die Protokollierung zu kopieren.

Die Protokollierung sollte automatisiert werden. Dadurch können die Protokolle sofort nach der Sitzung eingesehen werden. Aus der ersten Idee ist ein Programm entstanden, welches neben der Protokollierung noch weitere Funktionen beinhaltet. Die Dozenten und die Studierende haben ein Werkzeug zur Verfügung, welches ihnen das Arbeiten innerhalb von Second Life erleichtert. Zu keinem Zeitpunkt muss die virtuelle Welt verlassen werden. Dadurch ist das Arbeiten

effektiver und einfacher. Die Studierenden können sich mit Hilfe dieses Prototyps mehr auf die anderen Aufgaben im Second Life konzentrieren, da der Großteil der administrativen Arbeit vom Programm erledigt wird.

Kapitel 5

Implementierung

In diesem Kapitel wird der Prototyp auf Basis des Konzepts vorgestellt. Die Umsetzung erfolgte auf der Basis von Second Life und des MediaWikis. Das Unterkapitel 5.1 beschreibt dabei die Funktionen, die implementiert werden und die Technologien, die dabei helfen. Abschließend werden im Kapitel 5.2 die verschiedenen Probleme und Lösungsansätze besprochen, die während dieser Arbeit aufgetreten sind.

5.1 Umsetzung

Beim Konzept wurde das Programm schon in drei Teile aufgespalten. Dieser Ansatz wird auch hier weiter verfolgt. Folgende Technologien wurden verwendet:

Anwendung	Second Life Viewer Version 1.21.6.99587
SL-Skriptsprache	Linden Scripting Language (LSL)
Verbindung	HTTP Version 1.1 POST Request
Bot-Skriptsprache	PHP Version 5.2.8
Wiki	MediaWiki Version 1.13.0rc2

Tabelle 5.1: Verwendete Technologien in Second Life

Second Life verwendet die Sprache LSL. Diese Skriptsprache wird von Second Life vorgegeben, weil es nämlich keine Compiler außer dem LSL-Compiler gibt, die im Second Life eingesetzt werden können. Für den Bot wird PHP ausgewählt,

weil MediaWiki auch in PHP implementiert ist.

Die Verbindung zwischen den einzelnen Bereichen des Programms wird mit HTTP-Paketen bewerkstelligt. MediaWiki erlaubt nur POST-Requests, um die Sicherheit der Übertragung zu gewährleisten. Zur Vereinheitlichung sind zwischen Second Life und dem HTTP Supervisor auch nur POST-Pakete zugelassen. In der Abbildung 5.1 wird eine grobe Gliederung der Implementierung gezeigt.

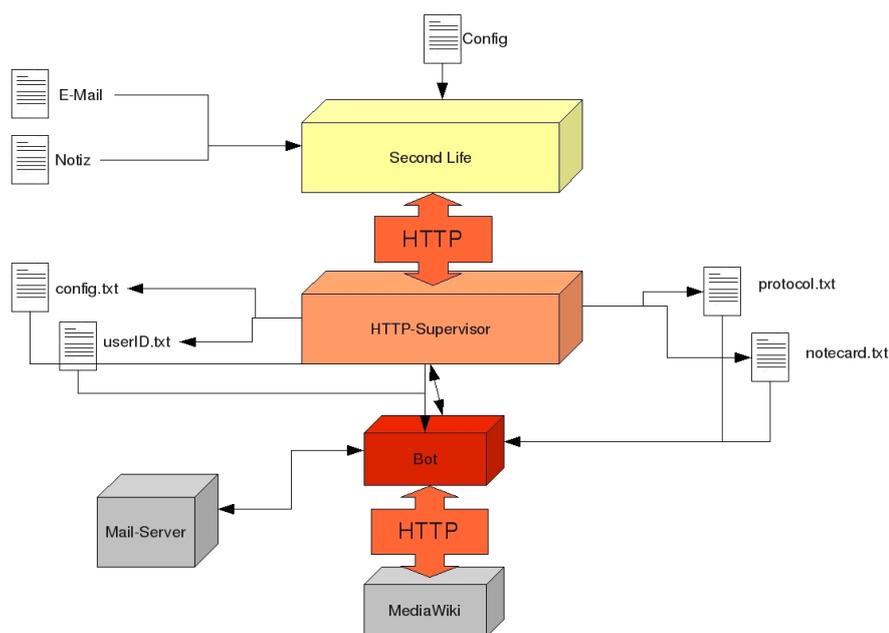


Abbildung 5.1: Implementierung der Anforderungen

5.1.1 Die Funktionen

5.1.1.1 Second Life Implementierung

Die Second Life Implementierung besteht aus acht Skripten. Diese sind:

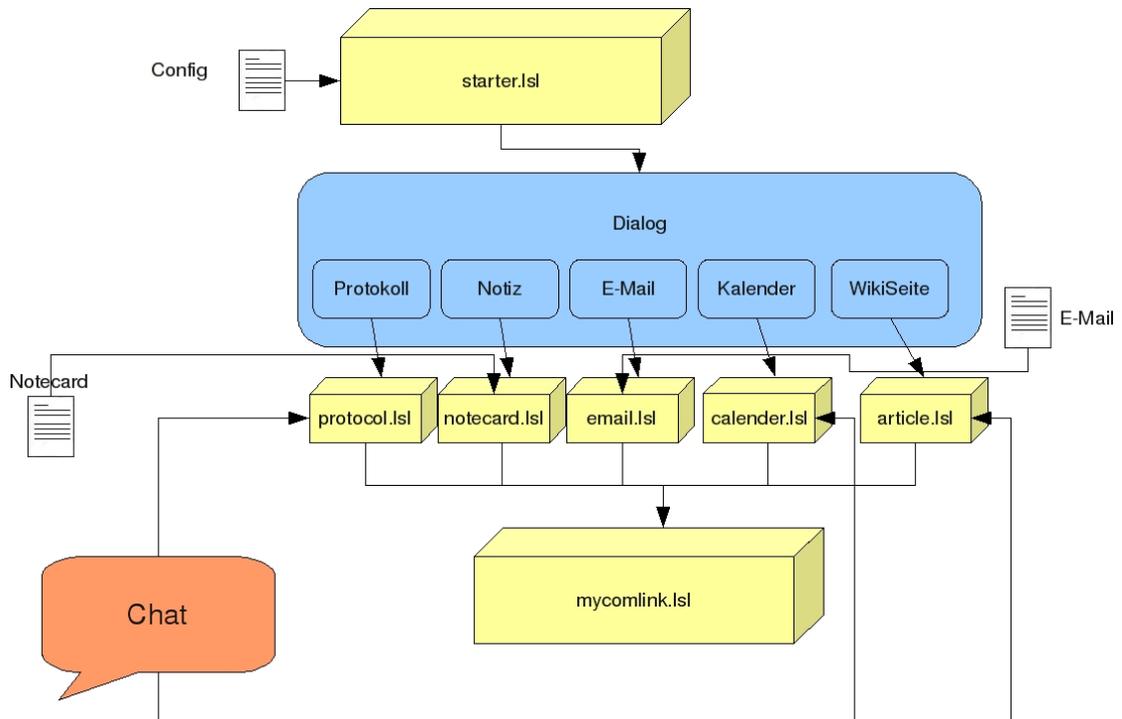


Abbildung 5.2: Implementierung aus der Sicht des Moderators.

<i>starter.lsl</i>	erste Steuerungseinheit (bedient den Moderator)
<i>userstarter.lsl</i>	zweite Steuerungseinheit (bedient die restlichen Benutzer)
<i>protocol.lsl</i>	Skript zur Steuerung der Protokollierung
<i>article.lsl</i>	Skript zur Steuerung der Erstellung der Wiki-Seiten
<i>email.lsl</i>	Skript zur Steuerung der E-Mail Erstellung
<i>notecard.lsl</i>	Skript zur Steuerung des Auslesens der Notizen
<i>sensors.lsl</i>	Skript zur Steuerung der Sensoren
<i>mycomlink.lsl</i>	Skript zur Kommunikation über HTTP und XMLRPC

Tabelle 5.2: Die acht Skripte der Second Life Implementierung.

starter.lsl Dieses Skript ist für das Starten und Stoppen des Programms zuständig. Zudem befindet sich in diesem Skript die Dialogfunktion, die für den Moderator den Hauptdialog darstellt. Alle anderen Benutzer werden direkt zu

userstarter.lsl weitergeleitet. Wenn eine Funktion durch den Moderator ausgewählt wird, schickt das Skript eine Nachricht, auch *Message Link* [Mel08] genannt. Diese Nachricht wird von allen Skripten empfangen und ausgewertet.

userstarter.lsl Das Skript *userstarter.lsl* bedient die Teilnehmer. Dabei stellt es, wie *starter.lsl*, einen Dialog zur Verfügung und wartet auf die Auswahl des Teilnehmers. Wenn die Auswahl getroffen wurde, wird die Nachricht wie beim vorherigen Skript an alle Skripte weitergeleitet. Das Skript, das für die Ausführung der Auswahl zuständig ist, kümmert sich um die weitere Abarbeitung.

mycomlink.lsl Dieses Skript bildet die zentrale Schnittstelle für die Kommunikation mit den entfernten Servern. Dabei öffnet das Skript einen XMLRPC-Kanal, sendet und empfängt HTTP-Nachrichten und empfängt die XMLRPC-Mitteilungen, die vom HTTP Supervisor kommen. Außerdem formatiert das Programm die Nachrichten so, dass sie vom HTTP Supervisor ausgewertet werden können.

protocol.lsl Die Anfrage der Protokollierung wird hier ausgewertet. Das Skript kümmert sich um die Farbveränderung des Objekts und die verschiedenen Warnungen, die verschickt werden müssen, wenn eine Nachricht bei aktiver Protokollierung gespeichert wird. Außerdem leitet das Skript die Nachrichten an *mycomlink.lsl* weiter.

notecard.lsl Wenn eine Notecard ausgelesen und verschickt werden soll, dann wird dieses Skript aufgerufen. Das Skript liest die Notecard aus und leitet den Inhalt ebenfalls an *mycomlink.lsl* weiter.

article.lsl Beim Anlegen einer Wiki-Seite wird dieses Skript aufgerufen. Es führt den Benutzer durch die erforderlichen Angaben.

email.lsl Das Skript *email.lsl* wird gestartet, wenn E-Mails verschickt und archiviert werden sollen.

sensors.lsl Das Sensors-Skript arbeitet unabhängig von anderen Skripten. Dieses Skript sendet Warnungen an Personen, die sich dem abgehörten Bereich nähern.

5.1.1.2 Die Server-Implementierung

Die Server-Implementierung besteht aus sieben PHP-Skripten: *article.php*, *calendar.php*, *email.php*, *httpsupervisor.php*, *notecard.php*, *protocol.php* und *wikibot.php*. Die Skripte *article.php*, *calendar.php*, *email.php*, *protocol.php* und *notecard.php* sind die Gegenstücke der im vorherigen Abschnitt erklärten LSL-Skripte. Das Skript *httpsupervisor.php* ist die Datenweiche, welche die Daten in temporären Dateien sichern und den richtigen Bot starten soll.

Das Skript *wikibot.php* beinhaltet die Klasse zur Kommunikation mit MediaWiki.

5.1.2 Benutzerarten

Der Prototyp unterscheidet zwischen zwei Arten von Benutzern: Moderator und Teilnehmer. Der Moderator ist die einzige Person, die sich anmelden muss, damit er/sie einige Funktionen exklusiv steuern darf. Die Teilnehmer hingegen haben nur eingeschränkte Rechte, d.h. ihnen ist die Protokollierung und das Verschicken von Notecards nicht gestattet. Die Benutzerdaten der Teilnehmer werden nicht im Objekt gespeichert. Deshalb müssen sie sich beim Aufruf jeder Funktion erneut anmelden.

5.1.3 Pseudo Zufallszahlen-Generator

Der Prototyp benutzt einen Zufallszahlengenerator. Der Generator bedient dabei die Teile des Prototyps, die sich mit dem Benutzer unterhalten und von ihm eine Antwort erwarten. Deshalb wird bei jedem Aufruf einer Funktion, die eine Antwort erwartet, durch den Generator der Kanal zufällig ausgewählt. Weil sich der Kanal bei jeder neuen Frage ändert, antwortet der Benutzer bei einer Funktion über mehrere Kanäle.

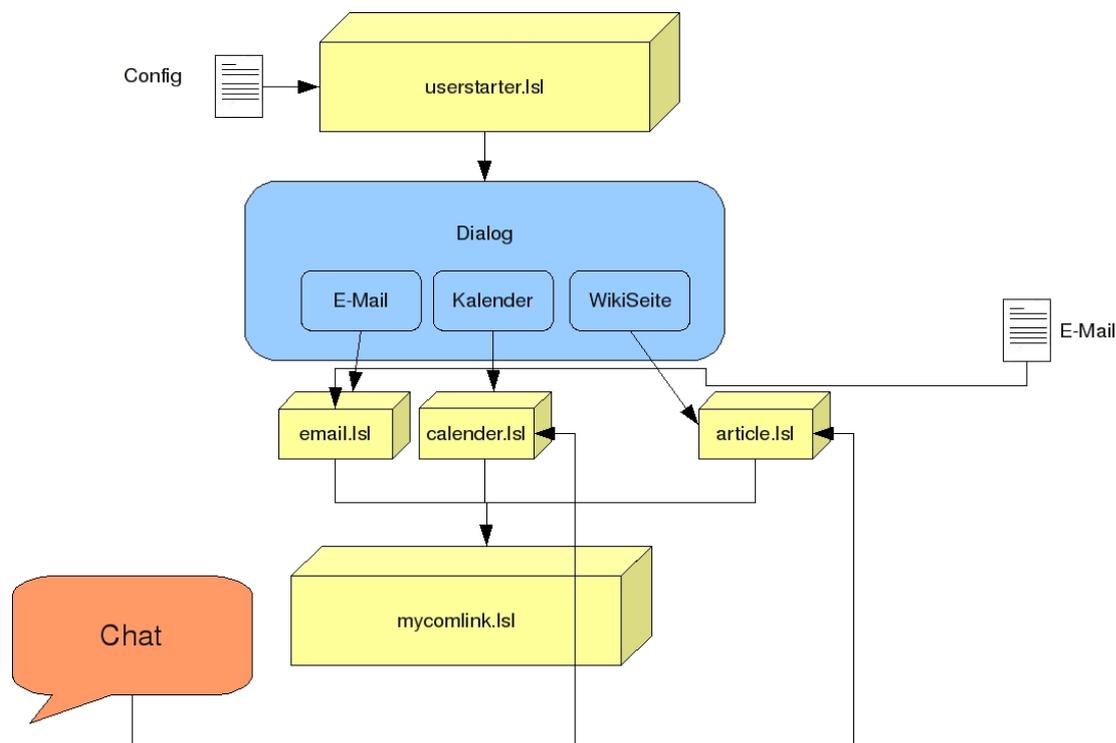


Abbildung 5.3: Implementierung aus der Sicht des normalen Benutzers

5.1.4 Nachrichtenübermittlung

Die einzelnen Nachrichten in Second Life werden mit Hilfe von *llMessageLinked* [Mel08] verschickt. Diese Funktion kann eine Nachricht nur an ein Objekt übermitteln. Dadurch bekommen alle Skripte in diesem Objekt die gleiche Nachricht. Damit die Skripte nicht auf jede Nachricht reagieren, wurden in dem Prototyp Befehle für jedes Skript implementiert. Die Befehle werden in der Tabelle 5.3 aufgeführt.

protocol	protocol.lsl
email	email.lsl
notecard	notecard.lsl
article	article.lsl
calender	calender.lsl
comlink	mycomlink.lsl

Tabelle 5.3: Befehle in Second Life

5.1.5 Wiki-Bot

Jede der Funktionen benötigt den Wiki-Bot, um die Informationen als eine Wiki-Seite zu speichern. Die einzelnen Schritte sind in allen Funktionen, bis auf die Kalenderfunktion und das Anhängen der Inhalte, identisch. Deswegen werden sie hier kurz erläutert.

Der Wiki-Bot ist eine PHP Klasse, die von jeder Funktion gestartet wird. Alle Funktionen zur Interaktion mit MediaWiki sind hier implementiert. Es handelt sich hierbei um das An- und Abmelden im MediaWiki, Tokenanforderung, Anfordern, Schreiben, sowie das Verändern einer Wiki-Seite.

Das Skript benutzt den Browsersimulator *Snoopy*¹ für die Kommunikation mit MediaWiki. Snoopy ist ein Anwendung, die unter der GNU Lesser General Public License (LGPL)² steht. Sie ist in PHP geschrieben und kann wegen der LGPL sowohl in freien als auch in kommerziellen Produkten eingesetzt werden. Mit Snoopy ist es möglich den HTTP-Request zu automatisieren. Dadurch kann der Bot auf seine grundlegenden Funktionen beschränkt werden und muss sich nicht um die Browserfunktionen kümmern. Die Funktionen des Bots sind im Skript `wikibot.php` gespeichert. Sie werden von jeder Funktion beansprucht. Die erforderlichen Informationen, die für jeden einzelnen Schritt übermittelt werden müssen, wurden bereits in Kapitel 4.1.3.1 erklärt.

Im Gegensatz zum Zugriff über den Browser, muss sich der Bot anmelden und ein *Edit*-Token anfordern. Die Anmeldung wird bei jedem Zugriff durchgeführt, damit sichergestellt wird, dass die Erstellung der Seite nicht an einem fehlerhaften

¹<http://m-software.de/snoopy.php>

²<http://www.gnu.org/licenses/lgpl-3.0.html>

Cookie scheitert. Falls die Anmeldung erfolgreich war, wird das Token angefordert. Nach der Anforderung des Tokens wird der Inhalt übermittelt. Der Wikibot wartet auf eine Meldung, ob der Inhalt richtig übertragen wurde. Wenn die Meldung über die erfolgreiche Einbettung des Inhalts erfolgte, wird an die Klasse, die den Wiki-Bot gestartet hat ein "OK" gesendet. Bei einer Fehlermeldung wird diese unverfälscht zurückgegeben. Der Benutzer wird danach abgemeldet und der Wiki-Bot wird gestoppt.

5.1.6 Initialisierung

Das Programm wird durch die Berührung des Objekts, in dem sich die Second Life-Skripte befinden, initialisiert. Während der Initialisierung werden alle Skripte gestartet. Das "*starter.lsl*" Skript dient als Kommandozentrale des Programms und nimmt nach der Berührung die Arbeit auf. Als erstes wird die Farbe des Objekts auf grün umgeschaltet. Die grüne Farbe signalisiert den Avataren in der Umgebung, dass die Protokollierung zur Zeit nicht ausgeführt wird und dass sie sich ungestört unterhalten können.

Der nächste Schritt betrifft die Konfiguration des Systems. Dazu wird eine Notecard mit dem Namen "*config*" ausgelesen. Diese Notecard muss sich im Objekt befinden, weil das Programm sonst mit einer Fehlermeldung abbricht.

Die "*config*" Notecard beinhaltet drei Parameter: Bot, Adresse und Art der Web 2.0 Anwendung. Der erste Parameter "*Bot*" beinhaltet die Botadresse. Auf dieser Adresse befindet sich der HTTP Supervisor, der die Informationen weiterleitet. Der zweite Parameter beinhaltet die Adresse der Web 2.0 Anwendung. Diese beiden Parameter werden vorher nicht überprüft. Die Fehlermeldung wird erst dann auftreten, wenn die ersten Daten an die beiden Adressen verschickt werden.

Der dritte Parameter wird dagegen sofort geprüft. Das Skript gleicht ihn mit einer Liste der Anwendungen ab, zu der eine Implementierung existiert. Zur Zeit beinhaltet diese Liste nur eine Anwendung: MediaWiki. Alle anderen Anwendungen werden abgelehnt und das Programm schaltet sich ab.

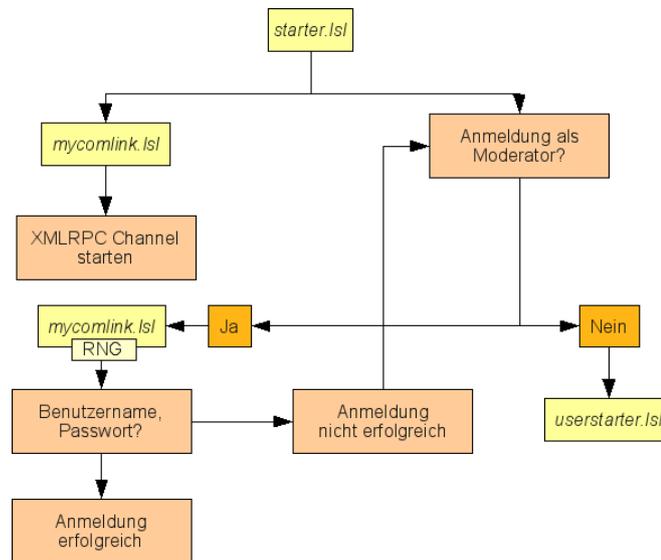


Abbildung 5.4: Initialisierung des Programms und Anmeldung des Moderators

5.1.7 Aufbau der Verbindung und Anmeldung des Moderators

Als nächstes wird das Skript "mycomlink.isl" ausgeführt. Dieses Skript kümmert sich um die HTTP-, E-Mail- und die XMLRPC-Verbindungen. Zuerst wird ein XMLRPC Kanal geöffnet. Wenn der Kanal offen ist, wird der Moderator nach dem MediaWiki-Benutzernamen und dem MediaWiki-Passwort gefragt. Diese Einträge werden im Chatfenster eingegeben. Die Formatierung der Eingabe wird im Textchat kurz erläutert. Der Benutzername und das Passwort werden gleichzeitig abgefragt. Die beiden Angaben werden durch ein Komma getrennt. Zudem werden beide Nachrichten in einem privaten Kanal übermittelt, damit die umstehenden Avatare diese nicht sehen können. Die Formatierung lautet wie folgt:

<Kanal Nr.> <Benutzername>,<Passwort>

Der Moderator wird danach nicht mehr nach den Benutzerdaten gefragt. Diese werden bis zur Abmeldung im System gespeichert. Falls der Benutzername und das Passwort richtig sind, bleibt der Benutzer als Moderator angemeldet. Falls nicht, wird der Benutzer abgemeldet und das Programm wartet auf einen neuen Moderator. Ein Abspeichern der Benutzerdaten aller Nutzer ist zu aufwendig. Deshalb werden alle anderen Benutzer erst kurz vor der Ausführung einer Funktion durch den Bot nach dem Benutzernamen und dem Passwort gefragt.

5.1.8 Touch - Berühren des Objekts

Nach der Berührung des Objekts öffnet sich ein Dialogfenster, in dem die gewünschte Funktion ausgeführt werden kann. Der Moderator hat alle sechs Funktionen zur Auswahl. Die Funktionsliste für die Teilnehmer beinhaltet nicht die Abmeldung, die Protokollierung und das Verschicken der Notecards. Die einzelnen Funktionen werden nun aus der Sicht des Moderators erklärt. Die Vorgehensweise des Programms bei gemeinsamen Funktionen ist bei einem Teilnehmer und beim Moderator die gleiche.

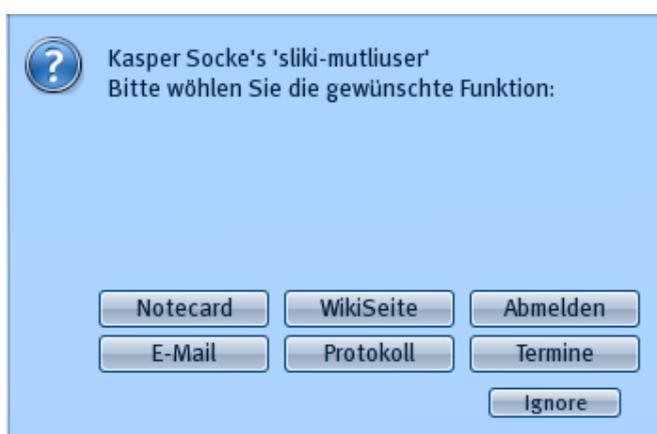


Abbildung 5.5: Auswahlfunktion aus der Sicht des Moderators

5.1.9 Abmeldung

Der Moderator muss sich abmelden, um Anderen die Steuerung der Zusatzfunktionen zu ermöglichen. Beim Abmelden werden alle Funktionen, die der Moderator gestartet hat, abgeschaltet. An den HTTP Supervisor wird eine Nachricht gesendet, dass er alle ausstehenden Funktionen des Benutzers beenden soll. Die Abmeldung findet in den Skripten *mycomlink.lsl* und *starter.lsl* statt. Im ersten werden die Benutzerdaten des Moderators gelöscht. Im zweiten Skript wird die Benutzer-ID in der Variable gelöscht. Danach wird jeder neue Nutzer gefragt, ob er sich als Moderator anmelden möchte.

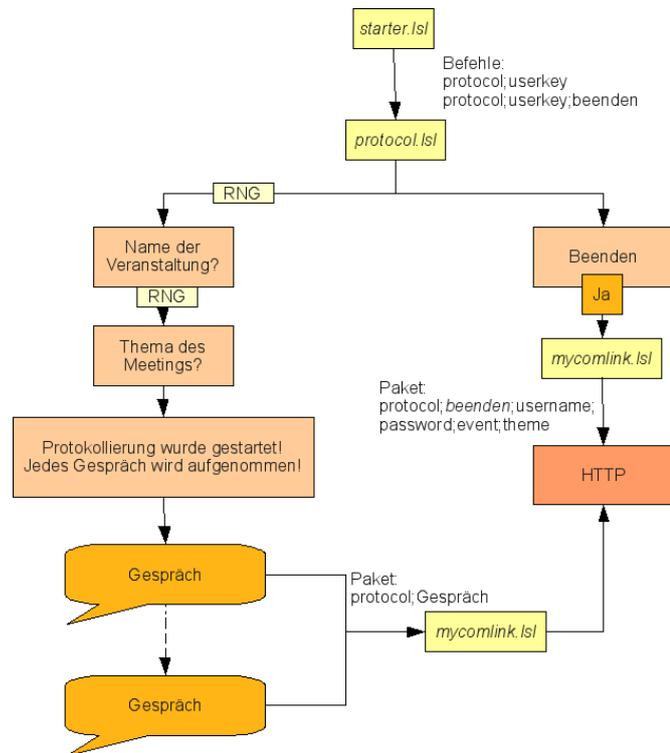


Abbildung 5.6: Die Protokollfunktion in Second Life

5.1.10 Protokollierung

Wenn der Moderator den Knopf „Protokoll“ auswählt, startet das Programm die Protokollierung. Wenn die Protokollierung bereits läuft, wird der Moderator gefragt, ob die Protokollierung abgeschaltet werden soll. Für die Protokollierung in Second Life ist das Skript *protocol.lsl* zuständig.

Bei der Initialisierung der Protokoll-Funktion wird zunächst nach dem Veranstaltungsnamen gefragt. Danach wird nach einem Thema für das Meeting verlangt. Vor beiden Anfragen wird mit dem Zufallszahlengenerator ein Kanal gewählt, auf dem der Moderator antworten muss. Nachdem beide Abfragen beantwortet wurden, startet die Protokollierung. Im Hauptchat wird eine Warnung angezeigt, damit alle Bewohner im Umkreis über die Protokollierung informiert sind. Danach wird der *llListen*-Befehl abgesetzt, der den Kanal 0 abhört. *llListen* ist ein Befehl zum Auslesen des Textchats durch ein Skript. Das Programm wartet nun auf ankommende Gespräche im Hauptchat. Wenn ein Bewohner etwas schreibt, wird dieses Gespräch sofort an das Skript *mycomlink.lsl* gesendet. Der Befehl lau-

tet:

comlink;protocol;userkey;<Gespräch>

Das Skript *mycomlink.lsl* erstellt einen HTTP-Request und verschickt die Daten an den HTTP Supervisor. Dieser speichert die Daten in die temporäre Datei *protocol*.

Die Protokollierung wird über das Dialogfenster beendet. Nach der erneuten Bestätigung, dass die Protokollierung beendet werden soll, sendet *mycomlink.lsl* ein Paket an den HTTP Supervisor. Das Paket sieht folgendermaßen aus:

comlink;protocol;userkey;beenden;username;password

Der HTTP Supervisor aktiviert nun das Skript *protocol.php*, die mit Hilfe des im Kapitel 5.1.5 erwähnten Wiki-Bots die Wiki-Seite erzeugt.

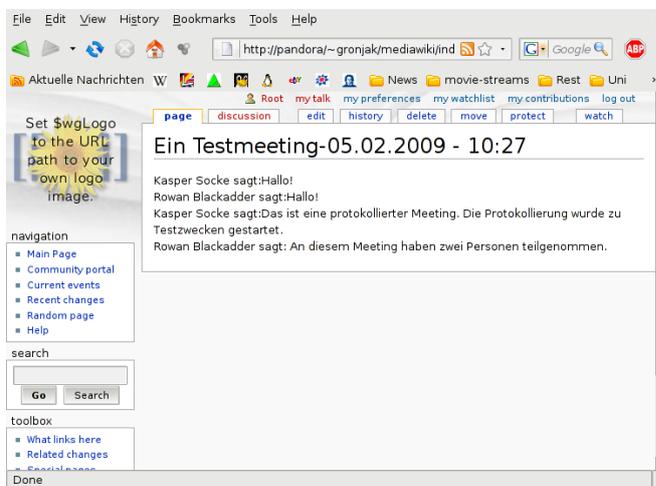


Abbildung 5.7: Die Protokollseite Testmeeting

5.1.11 Notecard verschicken

Für das Verschicken der Notecards ist *notecard.lsl* zuständig. Der Benutzer wird gefragt, um welche Veranstaltung es sich handelt und wie der Name der Notecard lautet. Falls der Name der Notecard falsch ist, wird der Moderator gebeten den

Namen zu wiederholen. Wenn der Name der Notecard im Objekt gefunden wurde, wird die Notecard mit dem Befehl *llGetNotecardLine* zeilenweise ausgelesen. Das Paket, das über *mycomlink.lsl* verschickt wird lautet:

notecard;userkey;<Zeile der Notecard>

Jede Zeile wird einzeln über den *llHTTPRequest* verschickt und in der temporären Datei *notecard* zwischengespeichert.

Nachdem die Notecard vollständig ausgelesen und verschickt wurde, wird automatisch der Befehl zum Erstellen der Wiki-Seite an den HTTP Supervisor verschickt. Das Paket, das dieses Mal über den HTTP-Request gesendet wird, beinhaltet hierbei den Benutzernamen, das Passwort, den Veranstaltungsnamen und den Namen der Notecard. Die Notecard wird unter dem selben Namen abgelegt, den sie auch in Second Life hatte. Die Speicherung übernimmt dieses Mal das Skript *notecard.php*.

5.1.12 Wiki-Seite anlegen

Das Anlegen einer Wiki-Seite verläuft ähnlich wie das Anlegen eines Protokolls. Der Unterschied hierbei ist, dass jeder Benutzer die Wiki-Seiten anlegen kann. Für diese Funktion ist das Skript *article.lsl* zuständig. Auch hier wird zuerst der Schlüssel des Benutzers (*die BenutzerID*) im Befehl verschickt. Dieses Mal wird allerdings die BenutzerID dazu benutzt, die temporäre Datei auf dem Server anzulegen. Diese Datei bekommt die Endung *.alt*, damit ersichtlich ist, dass in dieser Datei eine Wiki-Seite zwischengespeichert wurde. Als erstes wird wie immer nach dem Namen der Veranstaltung und dem Artikelnamen gefragt. Danach wird der Inhalt eingegeben. Alle Eingaben erfolgen über verschiedene Kanäle, die von einem Zufallszahlen-Generator ausgewählt werden.

Der Inhalt wird wieder sofort übertragen, der Artikel- und der Veranstaltungsname werden zunächst zwischengespeichert. Der Nutzer muss dieses Mal manuell *beenden* in den Textchat eingeben, damit die Seite erstellt wird.

Auch dieses Mal startet der HTTP Supervisor zunächst das Skript *article.php*, welches den Wiki-Bot benutzt. Das Skript überprüft zunächst über *action=query*, ob diese Seite existiert. Falls die Seite existiert, wird über den HTTP Supervisor die Anfrage verschickt, wie weiter verfahren werden soll. Der HTTP Supervisor

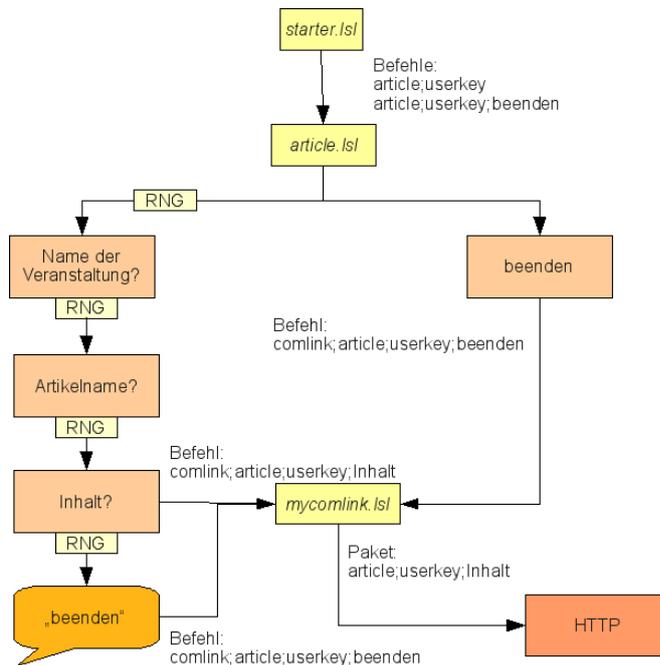


Abbildung 5.8: Anlegen einer Wiki-Seite. Ansicht der Second Life-Funktion, ohne das Anhängen der Inhalte an eine bestehende Seite.

benutzt dafür die Funktion HTTP-Response, die einen Rückkanal zu Second Life bildet. Der Nutzer hat nun die Möglichkeit, die Daten anzuhängen oder eine neue Wiki-Seite anzulegen. Die Entscheidung wird auf die gleiche Art und Weise verschickt, wie der erste Befehl zum Anlegen der Wiki-Seite.

5.1.13 E-Mail

Wird „E-Mail“ gedrückt, startet eine Funktion zum Erstellen und Senden einer E-Mail. Diese Funktion steht, wie das Anlegen der Wiki-Seite, jedem Benutzer zur Verfügung. Das Skript startet eine Abfrage über den Textchat. Die Anfragen werden über Instant Messages gestellt und über zufällig gewählte Textchat-Kanäle beantwortet. Das Prinzip wurde bereits in 5.1.3 erläutert.

Der Empfänger und der Absender können auch in einer Notecard im Objekt gespeichert sein. Ob das der Fall ist, wird in der ersten Anfrage geprüft. Wenn es zutrifft, wird der Name der Notecard abgefragt und diese wird ausgelesen. Jede Zeile der Notecard wird gesondert abgeschickt. Dadurch wird gewährleistet, dass die Daten nicht zu groß sind und vom *mycomlink.lsl* aufgespalten werden

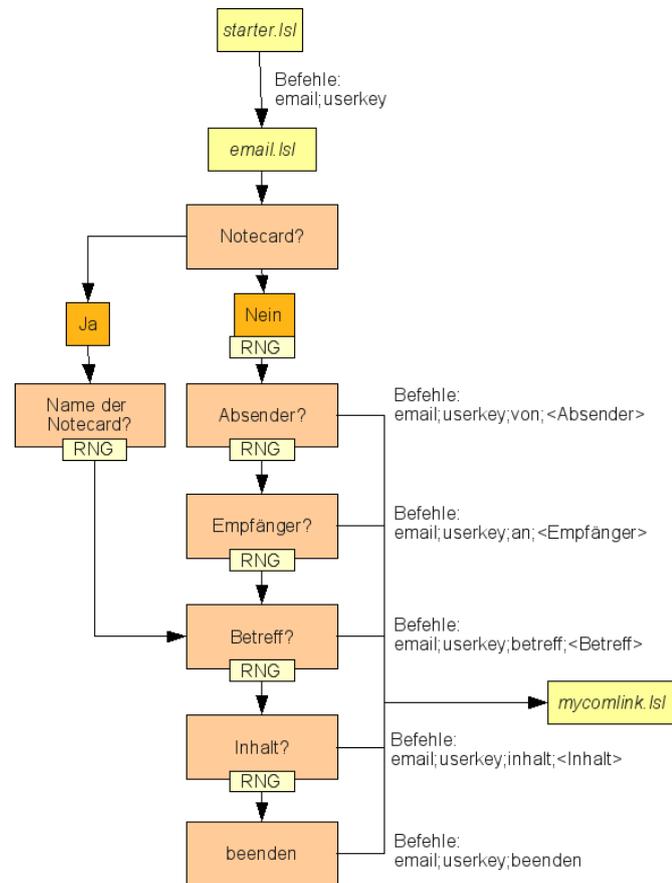


Abbildung 5.9: Die Schritte zur Erstellung einer E-Mail

müssen. Die ankommenden Daten werden alle vom HTTP Supervisor entgegen genommen. Danach wird der Betreff und der Inhalt der E-Mail abgefragt.

Wenn der Inhalt übergeben wurde, bekommt der HTTP Supervisor den Befehl zum Verschicken der E-Mail. Die E-Mail wird nun aus der temporären Datei ausgelesen. Das Senden übernimmt hierbei ein PHP-Befehl. PHP hat eine eingebaute Funktion zum Senden der E-Mails. Dafür muss auf dem Server, der die PHP-Skripte verarbeitet, ein E-Mail-Server arbeiten. Sollte das nicht der Fall sein, funktioniert die PHP-Funktion nicht.³

Das Archivieren der E-Mail übernimmt wieder der Bot. Dabei speichert der Bot die E-Mail in einer eigenen Seite, die unter <Veranstaltungsname>/E-Mails/ verlinkt wird.

³<http://de.php.net/mail>

5.1.14 Kalenderfunktion

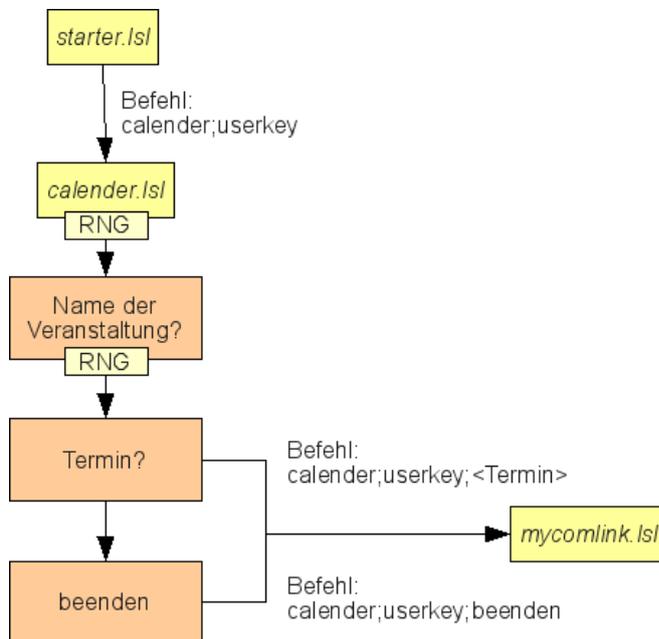


Abbildung 5.10: Die Kalenderfunktion und die Schritte in Second Life

Die Kalenderfunktion ist auch eine Funktion, die allen Nutzern zur Verfügung steht. In dieser Funktion wird der Nutzer über einen zufälligen Kanal nach dem Namen der Veranstaltung und nach dem Termin gefragt. Danach wird die Funktion beendet und die Wiki-Seite beschrieben. Die Besonderheit bei dieser Funktion ist, dass die Seite nicht überschrieben werden darf. Es reicht auch nicht, den Termin anzuhängen. Der Wiki-Bot erfragt mit Hilfe des HTTP-GET-Befehls und der im Kapitel 4.1.3.2 beschriebenen Funktion die Wiki-Seite und erhält die Rohdaten, die er verarbeiten muss. Die Wiki-Seite ist so aufgebaut, dass die einzelnen Absätze die Monate bilden. In jedem Absatz stehen die Termine für den jeweiligen Monat. Die Termine werden mit „:“ eingerückt. Am Ende jedes Termins steht der HTML-Befehl `
`, der das Ende der Zeile anzeigt. Ein neuer Termin hat das Format `01.Dezember.09,00:00:<Termin>`. Zuerst wird der Monat vom Rest getrennt, damit der richtige Absatz gewählt werden kann. Danach wird in diesem Absatz nach dem richtigen Tag gesucht. Falls unter dem gleichen Datum schon ein Termin steht, muss nach Stunden und gegebenenfalls Minuten der Termin eingefügt werden.

Wenn die richtige Stelle gefunden wurde, werden die Daten in einem Array ge-

speichert. An der Stelle `Array[0]` wird der erste Teil der Seite gespeichert, der später vor dem Termin stehen soll. Die Stelle `Array[2]` bildet den zweiten Teil der Seite, der dem Termin folgen soll. Der neue Termin wird an die Stelle `Array[1]` gespeichert. Davor muss der Termin noch formatiert werden. Die Formatierung beinhaltet das Voranstellen der beiden Doppelpunkte für die Einrückung und den HTML-Befehl `
` am Ende des Termins. Danach wird die Seite wieder zusammengesetzt, indem die drei Teile des Arrays in einen String gespeichert werden. Die Wiki-Seite wird danach wieder mit Hilfe des Wiki-Bots aktualisiert, wobei der alte Inhalt durch den neuen überschrieben wird.

5.2 Probleme und Lösungsansätze

LSL Bei der Implementierung in Second Life stellte die Skriptsprache Linden Scripting Language (kurz LSL) immer wieder ein Problem dar. Die Sprache ist nämlich auf größtmögliche Sicherheit ausgelegt. Deswegen sind viele Funktionen nicht implementiert (z.B. Exportieren von Bildern) oder werden vom Server für Skripte nicht freigegeben (z.B. `llSetNotecardLine`).⁴

HTTP Die negative Seite von HTTP ist, dass es mißbraucht werden kann. Ein mögliches Szenario ist das Verschicken einer Menge von Anfragen an einen Server, die diesen überlasten, so dass der Server aus dem Netz genommen werden muss. Dies ist ein sogenannter *DoS* (=Denial of Service) Angriff. Die übliche Vorgehensweise hierbei ist, den Rechner, von dem der Angriff gestartet wurde, auf eine Blacklist zu setzen. Danach werden die Pakete, die von diesem Rechner ausgehen, nicht mehr verarbeitet. Für Second Life, das auf das Internet angewiesen ist, wäre das das schlimmst mögliche Szenario. Deswegen wurden Vorkehrungen getroffen, um eine solche Gefahr erst gar nicht aufkommen zu lassen. Um das Problem zu lösen, wurde die Anzahl der HTTP-Requests aus Second Life heraus beschränkt. Der Benutzer darf innerhalb von 20 Sekunden höchstens 25 HTTP-Requests verschicken. Der Prototyp ist darauf ausgelegt, maximal einen Request pro Sekunde zu verschicken. Die Daten, die innerhalb dieser Sekunde anfallen, müssen deshalb gesammelt werden.

Beschränkung der Größe der Skripte Die Skripte in Second Life haben eine Beschränkung auf 16 Kilobyte. Diese Beschränkung erlaubt es nicht, komplexe Programme in einem Skript unterzubringen. Die großen Programme, wie der hier vorgestellte Prototyp, müssen deswegen in mehrere Skripte untergebracht werden. Ein weiteres Problem ist, dass sich die Größenbeschränkung nicht nur auf den Quellcode und die Konstanten erstreckt. Die Variablen im Skript werden auch hinzugezählt. Durch die Variablen, die in der Lage sind, eine beliebige Größe anzunehmen, kann die Grenze von 16 Kilobyte überschritten werden.

Die Lösung dieses Problems ist, das Programm in mehrere kleinere Skripte zu unterteilen. Je kleiner die Skripte sind, desto kleiner ist die Wahrscheinlichkeit, dass sie größer als 16 Kilobyte werden.

⁴vgl. 3.1.1

Erstellen von Notecards Die Notecards sind kleine Notizen, die man sich im Second Life schreiben kann und die man im persönlichen Inventarordner ablegt. Diese Notizen können als Konfigurationsdateien verwendet werden. Das Problem hierbei ist, dass das Erstellen von Notecards aus einem Skript heraus nicht gestattet wird. Dieses Mal sind es keine Sicherheitsbedenken, die die Programmierer davon abhalten, diese Funktion freizuschalten, sondern die Verantwortlichen befürchten, dass dadurch ein Overhead entstehen könnte, der die Server erheblich verlangsamen würde. In diversen Foren werden Diskussionen für und wider dieser Funktion geführt.⁵ Die Erstellung der Notecards würde die Erstellung von komplexeren Programmen in Second Life erheblich erleichtern, weil dadurch die Variablen ausgelagert werden könnten.

Eine Lösung für dieses Problem konnten wir nicht finden. Alle Variablen werden im Programm gesichert und die Werte für diese werden bei jedem Neustart des Skripts zurückgesetzt. Ein weiteres Problem wurde schon bei der Beschränkung der Größe der Skripte erwähnt.

Verbindungsprobleme Bei Verbindungen sind sowohl Verbindungen zwischen Objekten, wie auch Verbindungen zu entfernten Servern gemeint. Die Verbindungen können Pakete verlieren, wenn das Skript zu langsam reagiert. Dadurch kann es zu Beeinträchtigungen im Programmablauf kommen. Ein weiteres Problem ist, dass sich Datenpakete überholen können. Damit ist gemeint, dass ein Paket zwar als erstes abgeschickt wurde, aber am anderen Ende als zweites ankommt. Die Lösung hierbei ist, die Pakete mehrfach zu verschicken. Bei HTTP-Requests ist diese Möglichkeit wegen (5.2) nicht ratsam. Bei der Kommunikation kann sie problemlos angewandt werden.

Verzögerung durch Überlastung Das vorherige Problem ist eng mit dem aktuellen verbunden. Wenn nämlich zu viele Skripte oder zu viele Avatare in einem Grid befinden, kann es zu einer Störung durch Überlastung am Server kommen. Die Skripte reagieren in diesem Fall verzögert und es ist nicht sichergestellt, dass die Informationen richtig verarbeitet werden.

Es gibt zu diesem Problem leider keine Lösung, aber da sich meistens in Second Life nur relativ wenige Personen gleichzeitig in einem Grid befinden, muss der

⁵vgl. 3.1.3

Nutzer nur darauf achten, nicht zu viele Skripte gleichzeitig zu starten.

Serverausfälle und intensive Wartungsarbeiten Im November und Dezember 2008 kam es zu Ausfällen im Betrieb der Second Life Server. Durch die Ausfälle war es nicht möglich eine konsistentes Programm zu schreiben, weil der Server unberechenbar reagierte und in manchen Fällen Programmteile nicht ausführte. Außerdem wurden gelegentlich auch die Objekte, die die Skripte beherbergten gelöscht.

Kapitel 6

Zusammenfassung und Ausblick

Im Rahmen dieser Arbeit haben wir ein Konzept und einen Prototypen zur Kommunikation zwischen Second Life und Web 2.0 entwickelt.

Im Rahmen dieser Diplomarbeit wird ein Konzept und ein Prototyp zur Kommunikation zwischen Second Life und Web 2.0-Anwendungen entwickelt. In der Übung zur Veranstaltung "Einführung in das Projektmanagement" wurden Meetings in Second Life abgehalten. Dabei haben die Studierenden im Rahmen der Übung Protokolle erstellt, die sie im Internet veröffentlichten. Die Protokollierung musste immer manuell durchgeführt werden und war dadurch fehleranfällig und nicht ausfallsicher. Hier entstand der Wunsch, die Protokollierung zu automatisieren und somit den administrativen Aufwand zu reduzieren.

Im Kapitel 2 werden die Grundlagen behandelt, die für das Verständnis der Arbeit notwendig sind. Hierbei werden Second Life, sowie Blog und Wiki als Repräsentanten von Web 2.0 vorgestellt. Außerdem wird eine klare Abgrenzung zwischen diesen Technologien aufgezeigt.

Die Analyse dieser Diplomarbeit umfaßt unter anderem die verschiedenen Möglichkeiten der Übertragung der Informationen zwischen Second Life und den Web 2.0-Anwendungen.

Im Konzept ist die Gesamtarchitektur zur Kommunikation zwischen Second Life und den Web 2.0 Anwendungen enthalten. Die Hauptsystemkomponenten, die dafür notwendig sind, stellen eine Ansammlung aus Second Life Skripten, dem HTTP Supervisor und den Wiki Bot Skripten dar. Die Second Life und die Wiki Bot Skripten sind eine Ansammlung von Unterprogrammen, die jeweils auf eine Aufgabe spezialisiert sind. Die genaue Erklärung wird im Kapitel 4

Aufgrund der Unterschiede der Web 2.0 Anwendungen wurde ein Bot eingesetzt,

der eine Web 2.0 Anwendung bedient. Der HTTP Supervisor dient der Vermittlung dient der Vermittlung der Daten zwischen Second Life und der Web 2.0 Anwendung. Außerdem speichert er die Daten in temporäre Dateien. Der Second Life Teil des Programms dient der Steuerung des gesamten Sysyem.

Durch die prototypische Umsetzung ist die Durchführbarkeit bewiesen, die Daten aus Second Life herauszuführen und im MediaWiki zu speichern.

Literaturverzeichnis

- [Alb07] Tom Alby. *Web 2.0 - Konzepte, Anwendungen, Technologien*. Hanser Verlag München, 2007.
- [apa08] Apache http server project, 01.12.2008.
- [Bal98] Helmut Balzert. *Lehrbuch der Softwaretechnik: Softwaremanagement, Software Qualitätssicherung*. Spektrum Akad. Verlag Heidelberg Berlin, 1998.
- [cms09] Content-management-system, February 01.02.2009.
- [Dro05] Prof. Dr. Oswald Drobnik. Kapitel 5-9. In *Vorlesungsskript: Verteilte Systeme und Telematik II*, WS 2004/05.
- [GB02] A. Petersen G. Bente, N. C. Krämer. *Virtuelle Realitäten*. Hogrefe, Göttingen, 2002.
- [gdv08] It-projektmanagement, 01.12.2008.
- [Gei95] K. Geihs. *Client/Server Systeme*. International Thomson Publishing, Bonn, 1995.
- [GF05] Jürgen Fr Wolff von Gudenberg Gregor Fischer. *Programmieren in Java 1.5. Ein kompaktes, interaktives Tutorial*. Springer Verlag Berlin, 2005.
- [itp05] Einführung in das it-projektmanagement, 13.10.2005.
- [kem08] Putting a second life “metaverse” skin on learning management systems, February 01.02.2008.
- [Kra08] Jörg Krause. *PHP 5: Webserver Programmierung unter Windows und Linux*. Carl Hanser Verlag München, 2008.

- [Lan05] Chistoph Lange. *Wiki Planen, Einrichten, Verwalten*. 1. Computer und Literaturverlag GmbH, May 2005.
- [Lex08] T. Lex. *Möglichkeiten virtueller 3D-Welten für Bildung und E-Learning am Beispiel von Second Life*. Diplomica Verlag, Hamburg, 2008.
- [Lit95] Hans D. Litke. *Projektmanagement: Methoden, Techniken, Verhaltensweisen*. Hanser Verlag, München, Wien, 1995.
- [liv08a] Google lively - kostenlose 3d-chaträume für webseiten, July 09.07.2008.
- [liv08b] Google startet eigenes second life, July 09.07.2008.
- [lsl09] Homepage von lsl wiki, February 01.02.2009.
- [Mel08] Matthias Melzer. *Second Life Programmierung*. 1. Carl Hanser Verlag München, May 2008.
- [ML07] Antje Müller and Martin Leidl. Virtuelle welten - second life in der lehre, 04.12.2007.
- [moi09] Moinmoinwiki homepage, February 01.02.2009.
- [moo09a] Moodle.org: open-source community-based tools for learning, February 01.02.2009.
- [moo09b] Was ist moodle?, February 01.02.2009.
- [Mün05] Stefan Münker. *Medienphilosophie der Virtual Reality In: M. Sandbothe und L. Nagl (Hrsg.) Systematische Medienphilosophie*. Akademie Verlag Berlin, 2005.
- [mw009] Mediawiki, February 01.02.2009.
- [mw209] Mediawiki: Editing pages, February 01.02.2009.
- [mw309] Mediawiki: Forum, February 01.02.2009.
- [mw409] Mediawiki: Faq, February 01.02.2009.

-
- [Neu04] B. Neuhausen. *Bildung in der Digitale: Zur Bildungsrelevanz virtueller Welten*. Lang, Peter, Frankfurt, 2004.
- [php09] php-homepage.de - die deutschsprachige ressource für php und mysql, February 01.02.2009.
- [PZ02] F. W. Hesse P. Zentel, U. Cress. *Virtueller Campus: Szenarien - Strategien - Studium*. Waxmann, Münster, 2002.
- [Rah07] Miznaur Rahman. *MediaWiki Administrators' Tutorial Guide*. Packt Publishing Ltd. Brimingham, 2007.
- [sl101] Homepage second life, February 01.
- [sl309] Second life ide, February 01.02.2009.
- [sli05] Das institut für wissensmedien in second life, 13.10.2005.
- [slo09a] sloodle - virtual environment learning system, February 01.02.2009.
- [slo09b] Sloodleuserdocs, February 01.02.2009.
- [slo09c] What is sloodle?, February 01.02.2009.
- [slo09d] What is sloodle?, February 01.02.2009.
- [Sti07] S. Stillich. *Second Life: Wie virtuelle Welten unser Leben verändern*. Ullstein TB Verlag, Berlin, 2007.
- [VKV+06] M. Völkel, M. Krötzsch, D. Vrandecic, H. Haller, and R. Studer. Semantic wikipedia. In: Proceedings of the 15th international conference on World Wide Web, May 2006.
- [vr008] Virtuelle realität, 19.06.2008.
- [wik09a] Wikipedia, February 01.02.2009.
- [wik09b] Wikipedia:bots, February 01.02.2009.
- [Wun] Patrick Wunderland. Vier erfolgskfaktoren für second life, web 2.0 und web 3d.
- [Wun09] Patrick Wunderland. Eclipse plugin für lsl, February 01.02.2009.