

Benjamin Ducke, FU Berlin, 11. Semester

Prähist. Arch. (HF), Klass. Arch (NF), Informatik (NF)

Projektseminar Internet Learning; Klaus-Dieter Graf, Marco Rademacher

Sommersemester 2001

Educational MUDs/MOOs

Die vorliegende Arbeit entstand im Sommersemester 2001 im Rahmen des Projektseminars "Internet Learning" am Institut für Informatik der Freien Universität Berlin. Ihr Gegenstand ist der Einsatz von *MUDs* (Multi User Dungeon) als interaktive Lernumgebungen, also sogenannte *Educational MUDs*, wie sie im Internet bereits seit längerem in vielfältiger Form präsent sind.

Eine speziell für den Bereich interaktive Bildung konzipierte Weiterentwicklung des MUD-Konzeptes stellt das sogenannte MOO (MUD Object Oriented) dar. Mit dem *LambdaMOO* existiert eine frei verfügbare, leistungsfähige Implementation. Dass MUDs/MOOs, zumindest im Bildungsbereich, immer noch zu den weniger populären Bereichen des Internet gehören, dürfte nicht zuletzt dem terminalorientierten Charakter der meisten Umgebungen zuzuschreiben sein, der diese gerade für gelegentliche Besucher nicht unbedingt attraktiv erscheinen lässt. Seit einiger Zeit existieren aber ausgereifte Lösungen, die benutzerfreundliche Oberflächen anbieten, ohne die Flexibilität des Systems einzuschränken. Ziel meiner Arbeit ist es, interessierten Benutzern zunächst eine Anleitung zur Installation und Einrichtung eines MOOs zu geben und dann anhand eines Beispiels (*MaMOOth*) Anregungen zu geben, Lehr- und Lernkonzepte damit umzusetzen.

Inhaltsverzeichnis

1 Schulbank und virtuelle Identität	3
1.1 Elemente eines MOO	6
1.2 Räume, Objekte und Personen im MOO	7
2 Einrichtung und Benutzung	7
2.1 LambdaMOO	8
2.1.1 Installation des Servers	8
2.1.2 Starten des Servers	8
2.1.3 Einrichten eines Clients	9
2.2 enCore	9
2.2.1 Anmelden am MOO	10
2.2.2 Einrichten einer virtuellen Identität	10
2.2.3 Navigation im MOO	11
2.2.4 Benutzen von Objekten	12
2.2.5 Kommunikation mit Benutzern	12
2.3 Xpress	13
2.3.1 Navigation	17
2.3.2 Benutzung von Objekten	18
3 Administration	19
3.1 Einstellungen für enCore/Xpress	19
3.2 Benutzerverwaltung	19
4 Gestaltung und Programmierung	21
4.1 Objekte und Räume	21
4.1.1 Basisobjekte und -räume von enCore	22

4.1.2	Didaktische Objekte und Räume in enCore	23
4.1.3	Objektverwaltung	27
4.2	Die MOO-Programmiersprache	28
4.3	Programmieren in Xpress	30
5	MaMOOth – Steinzeit erfahren im MOO	31
6	Aspekte der Wissensvermittlung im MOO	33
7	Schlussbetrachtung	35

1 Schulbank und virtuelle Identität

MUDs (Multi User Dungeons¹) sind interaktive Umgebungen, innerhalb derer reale Benutzer eine virtuelle Identität annehmen. Ein *MUD* besteht aus miteinander verbundenen Räumen, in denen sich die Benutzer bewegen. Räume enthalten Objekte, die mit einer Vielzahl von Eigenschaften ausgestattet sind und von den Benutzern betrachtet und manipuliert werden können. Ein MUD ist ein sich ständig veränderndes, hochgradig interaktives System. Seine Attraktivität gewinnt es nicht in erster Linie durch ein möglichst perfektes, vorgefertigtes Raum- und Objekt-Design, sondern durch die Interaktion seiner Benutzer mit anderen Benutzern, Objekten und Räumen. Ein hierarchisches Rechtssystem ermöglicht es dem Administrator eines MUD, bestimmte Benutzer mit der Fähigkeit auszustatten, neue Räume und Objekte zu erschaffen. Auf diese Weise fließen Phantasie und Ehrgeiz der Benutzer in das Wachstum und die Ausgestaltung des MUD ein und schaffen eine starke Bindung der Benutzer zu ihren virtuellen Identitäten² und den Räumen und Objekten im MUD.

Wie der Name schon andeutet, handelt es sich bei MUDs in der Regel um Rollenspiel-Umgebungen. Dementsprechend findet sich in solchen Systemen eine stark hierarchische Struktur, deren

¹Der Name deutet an, dass diese Technologie ihre Wurzeln v.a. im Bereich der Online-Rollenspiele hat, wo sie sich nach wie vor großer Beliebtheit erfreut.

²Das Konzept der virtuellen Identität hat allerdings im Bereich der auf Bildung ausgerichteten MUDs/MOOs eine weniger große Bedeutung als bei den für Rollenspiele konzipierten MUDs. Für die hier besprochene Anwendung reicht es aus, darunter einen Benutzernamen mit Passwort und Benutzerrechten zu verstehen, der es einem Besucher ermöglicht, das MOO zu erforschen und u.U. selbst mitzugestalten. Im folgenden wird zumeist einfach der Begriff "Benutzer" gebraucht.

obere Ebenen Benutzern erst durch punktebringende, spielerische Erfolge zugänglich werden. Ganz anders stellt sich das vom MUD-System abgeleitete *MOO* (MUD Object Oriented) dem Benutzer dar. In einer solchen, von vorne herein zu Bildungszwecken konzipierten, Umgebung gibt es keine ausgeprägte Hierarchie unter den Benutzern und auch kein Punktesystem, auch wenn aus technisch-praktischen Gründen ein Rechteverwaltungssystem und ein Administratoren-Konzept weiterhin bestand haben. Vielmehr wird gleichberechtigte Kommunikation und Interaktion von Anfang an in den Vordergrund gestellt – eine Eigenschaft, die vielfältige Konsequenzen für die Wissensvermittlung im MOO hat.

Es stellt sich nun die Frage, inwieweit das interaktive Potential der MOO-Technik für den Bildungsbereich nutzbar gemacht werden kann. Dabei sollen verschiedene Fragestellungen untersucht werden, die sowohl technischer als auch methodisch-didaktischer Natur sind:

1. Wie ausgereift ist die verfügbare Software und ist es möglich, ein MOO zu Bildungszwecken auf einem gängigen PC mit vertretbarem Zeit- und Kostenaufwand aufzubauen und zu betreiben?
2. Kann man das Potential eines MOO auch Benutzern mit geringen Kenntnissen in der PC-Benutzung zugänglich machen?
3. Wie groß ist die Medientauglichkeit eines MOO? Können Inhalte ansprechend, d.h. auch unter Einbeziehung von Bild- und Tonmaterial, aufbereitet werden?
4. Wie funktioniert die Wissensvermittlung im MOO und wo sind die Unterschiede, Vor- oder Nachteile im Vergleich zu anderen Lern- und Lehrmethoden, bzw. wo liegt das besondere Potential eines MOO?

Die erste Frage steht in engem Zusammenhang mit der Verfügbarkeit von Dokumentation zu Inbetriebnahme und Wartung eines MOO. Es ist auch ein Anliegen dieser Arbeit, hierzu verfügbare Informationen aufzubereiten und in verständlicher Form zur Verfügung zu stellen. Bildung steht heutzutage oft genug in direktem Zusammenhang mit begrenzten technischen und finanziellen Mitteln. Daher der Schwerpunkt dieser Arbeit auf der Anwendung von frei verfügbarer und auf preiswerten Systemen einsetzbarer Software. Jede im Rahmen dieser Arbeit vorgestellte Software ist kostenlos und im Quellcode erhältlich. Es steht dem Benutzer frei, sich an der Weiterentwicklung der Software zu beteiligen oder sie durch Veränderung am Quellcode auf die eigenen Bedürfnisse zuzuschneiden und sie in beliebig veränderter Form einzusetzen und weiter zu verbreiten. Die diesen Möglichkeiten zugrundeliegende Softwarelizenz ist die *GNU GPL* [1],

unter der ja auch schon so erfolgreiche Softwareprojekte wie der Web-Server *Apache* [2] oder das Betriebssystem *Linux* [3] entwickelt und verbreitet worden sind.

Die zweite und dritte Frage sind eng miteinander verbunden und beziehen sich vor allem auf das Problem der Benutzeroberfläche. Diese besteht bei einem "klassischen" MOO aus einer zeichenorientierten Telnet-Verbindung. Der Benutzer muss hier mittels Kommando-Eingabe navigieren und die Umgebung teilt sich ihm lediglich durch beschreibende Texte und einfache ASCII-Grafiken mit. Mit dem ebenfalls unter der GNU GPL stehenden MOO *enCore* [4] existiert jedoch eine ausgereifte Alternative mit grafischer Benutzeroberfläche, deren Einsatz im weiteren noch ausführlich beschrieben wird.

Im Rahmen dieser Arbeit wird eine Anleitung gegeben, um ein MOO aus frei im Internet erhältlicher Software zu installieren und soweit einzurichten, dass alle wesentlichen Aspekte verdeutlicht werden können. Die Abschnitte 2 bis 4 enthalten eine Anleitung zur Einrichtung und Verwendung des MOO. Das Anliegen dieses Teils der Arbeit ist es, die zur Installation und Benutzung bzw. Verwaltung eines MOO notwendigen Schritte vollständig und verständlich zu dokumentieren. Grundlage hierfür ist die Software *enCore* [4], die eine erweiterte Version des ursprünglichen *LambdaMOO* ist. Diese wurde ausgewählt, da sie frei erhältlich, gut dokumentiert und funktionell vollständig ist. Außerdem verfügt sie mit *Xpress* über einen ausgereiften, komfortablen Web-Client, der es den Benutzern wesentlich erleichtert, das MOO zu erschließen. Die weiteren Abschnitte (5-7) der Arbeit befassen sich mit konzeptionellen Aspekten zur Wissensvermittlung im MOO, die anhand einer kleinen Beispielanwendung verdeutlicht werden sollen.

Voraussetzung zum Verständnis der technischen Ausführungen ist eine grundlegende Vertrautheit mit Unix/Linux-Systemen im allgemeinen und speziell den zur Installation von Software auf einem solchen System notwendigen Schritten. Darüber hinaus werden ein grundsätzliches Verständnis von Netzwerkanwendungen mit Client-Server-Architektur und ein installierter Web-Server (z.B. *Apache*, s. [2]) zum Einsatz der in dieser Arbeit behandelten Anwendungen nötig sein. Ein MOO lässt sich grundsätzlich auch auf einem Windows- oder MacOS-Rechner aufbauen. Diese und weitere denkbare Plattformen bleiben im folgenden aber unberücksichtigt, da ihre Behandlung den technikorientierten Teil dieser Arbeit unnötig aufblähen würde.

▷ Informationen zum Betrieb von Netzwerkanwendungen unter Linux finden sich im *Linux-Networking-HOWTO* [5], während das *Apache-Overview-HOWTO* [6] speziell auf den Web-Server *Apache* eingeht. Das *Software-Building-HOWTO* [7] beschreibt die allgemeine Installation von Software unter Linux.

Der erste MOO-Server wurde zwar von Stephen White entwickelt, der Begriff MOO ist heutzutage jedoch meist synonym mit *LambdaMOO*, das von Pavel Curtis am Xerox-Parc entwickelt wur-

de. Curtis baute auf dem Design von White auf und erweiterte dieses zum ersten praxistauglichen MOO. Praktisch jedes heute im Internet anzutreffende MOO basiert auf Curtis' LambdaMOO-Kern, dessen Quellcode im Internet frei zugänglich ist [8].

1.1 Elemente eines MOO

Ein MOO besteht aus den drei Grundelementen *Datenbank*, *Server* und *Client* und stellt somit eine sogenannte "Client-Server-Architektur" dar. Der "Besitzer" des MOO, im folgenden als *Administrator* bezeichnet, startet auf seinem Rechner den MOO-Server mit der einzusetzenden Datenbank als Argument. Besucher (im folgenden *Benutzer*) können sich dann über das Netzwerk am Server anmelden und das MOO mit Hilfe eines Clients betreten.

Die Datenbank enthält Abbildungen aller Objekte im MOO inklusive der Programme, die deren Eigenschaften bestimmen. Die Kerndatenbank des LambdaMOO enthält einige grundlegende Objekte, Funktionen zur Administration des MOO sowie eine Programmiersprache, die es ermöglicht, die Datenbank beliebig zu erweitern. Eine MOO-Datenbank ist ein voll programmierbares System, das außer Beschreibungen und Topologie der in ihm enthaltenen Objekte praktisch alles enthält, womit der Benutzer in Berührung kommt. Dazu gehören in der internen Programmiersprache implementierte Räume und Objekte ebenso wie Administrationswerkzeuge, aber, im Falle des noch vorzustellenden enCore, auch ein vollständiger Web-Client.

Der Server verwaltet Netzwerkverbindungen und Zugriffe auf die Datenbank und führt darin enthaltene Programme aus. Bei dem für ein MOO eingesetzten Server handelt es sich praktisch immer um den Server aus dem LambdaMOO-Paket (z.B. [9]).

Die Schnittstelle zwischen dem MOO und den Benutzern bildet der Client. Der Client für ein klassisches MOO ist eine zeichenorientierte Telnet-Verbindung. Benutzer können hier mit einfachen Kommandos in der Umgebung navigieren, Objekte betrachten, untersuchen und benutzen, sowie mit anderen Benutzern kommunizieren. Dabei bekommen sie Texte angezeigt, die ihre Umgebung und die betrachteten Objekte beschreiben. Es existieren einige frei erhältliche MOO-Clients, die etwas mehr Benutzerfreundlichkeit bieten, als eine einfache Telnet-Verbindung. tkMOO-light [10] ist beispielsweise ein in der Skript-Sprache Tcl/Tk [11] implementierter Client, der immerhin ein menügeführtes Anmelden am MOO erlaubt und den Umgang mit den auf den Benutzer einströmenden Textmengen etwas erleichtert. Ein weiterer sehr beliebter Client ist MacMoose [12]. Der Xpress-Client aus dem enCore-Paket arbeitet in einem Web-Browser, wie etwa dem Netscape Navigator, und besteht aus HTML-, Java- und JavaScript-Teilen. Mit

diesem Client braucht der Benutzer seine gewohnte Web-Browser-Umgebung nicht zu verlassen und kann mit Hilfe einer grafischen Benutzerführung im MOO navigieren.

1.2 Räume, Objekte und Personen im MOO

Dem Benutzer stellt sich das MOO als eine Sammlung von *Räumen*, *Objekten* und *Personen*³ dar. Er kann sich mit Hilfe einfacher Kommandos von Raum zu Raum bewegen und die darin enthaltenen Objekte anschauen und manipulieren. Jedes Objekt verfügt über sogenannte *Verben*, die seine Interaktionseigenschaften beschreiben. Aus Benutzersicht handelt es sich dabei um Kommandos, die eingegeben werden können, um ein Objekt in vielfältiger Weise zu manipulieren. Alle in LambdaMOO vorhandenen Objekte kennen beispielsweise das Verb “examine”, mit dem sich eine detaillierte Beschreibung des Objekts anzeigen lässt: `examine Me`.

Die zweite interaktive “Komponente”, auf die ein MOO-Benutzer unweigerlich stoßen wird, sind andere Benutzer – schließlich handelt es sich bei einem MOO um ein Mehrbenutzersystem mit ausgefeilten Kommunikationsmöglichkeiten. Da sind zunächst einmal Gäste, die sich anonym, ohne Angabe ihres realen Namens oder ihrer E-Mail-Adresse im MOO bewegen. Meist handelt es sich dabei um Besucher, die zum ersten Mal das MOO betreten haben und noch nicht über einen eigenen Account verfügen. Unter den regulären, d.h. mit eigener virtueller Identität ausgestatteten, Benutzern bestehen Unterschiede im Hinblick auf deren Möglichkeiten, das MOO selbst mitzugestalten. Der Administrator kann ihnen gezielt Rechte zuweisen, um ihren Spielraum zu vergrößern. Ein zum *Builder* beförderter Benutzer kann selbst neue Objekte und Räume erschaffen, ein *Programmer* darf zusätzlich neue Verben und Eigenschaften für Objekte und Räume programmieren und im MOO zur Verfügung stellen und einem *Wizard* stehen alle Möglichkeiten offen, d.h. er darf auch administrative Tätigkeiten, wie etwa das Erstellen neuer Benutzer-Accounts, ausführen.

2 Einrichtung und Benutzung

Die hardware-technische Grundlage zur Inbetriebnahme eines MOO ist ein PC der Pentium-Klasse mit Linux als Betriebssystem. Sicherlich sind auch andere Betriebssysteme (wie etwa

³Auch Personen und Räume sind aus Sicht des MOO-Servers nichts anderes als Objekte. Trotzdem werden im folgenden, soweit es sich nicht um explizit technische Betrachtungen handelt, diese drei Gruppen sprachlich voneinander getrennt.

Windows) und Rechnerarchitekturen für den Einsatz als MOO-Server denkbar. Eine aktuelle Linux-Distribution (als CD-ISO-Images z.B. von S.u.S.E., Redhat oder Mandrake im Internet verfügbar [13]) enthält aber auf jeden Fall die gesamte für den MOO-Einsatz notwendige Software wie C-Compiler und Webserver.

2.1 LambdaMOO

Grundlage aller MOOs ist der Original LambdaMOO von Pavel Curtis. LambdaMOO ist weit verbreitet, entsprechend ausgereift und zudem unter der GNU GPL frei im Quellcode erhältlich. Da das System recht kompakt ist, bedeutet es keinen großen Aufwand, die nur wenige MB umfassenden Dateien vom FTP-Server herunterzuladen.

2.1.1 Installation des Servers

Die Installation auf einem lokalen Linux-System ist einfach. Zur Einrichtung der Software benötigt man die folgenden Dateien vom FTP-Server [8]:

LambdaMOO-latest.tar.gz

LambdaCore-latest.db.gz

*ProgrammersManual.*⁴*

HelpSystem-02Jun93.txt

Nach dem Aufruf von `./configure` im Quellenverzeichnis ermittelt das Konfigurationsskript, welche Netzwerkmodi vom eigenen System unterstützt werden. Das vorgeschlagene TCP/IP sollte i.d.R. funktionieren. Danach reicht ein einfacher Aufruf von `make` und nach kurzer Zeit steht das ausführbare `moO` Programm bereit.

2.1.2 Starten des Servers

Zunächst wird die Datei *LambdaCore-latest.db.gz* mit `uncompress` dekomprimiert. Diese enthält die Basisdatenbank mit den grundlegenden Objekten und Funktionen. Dann reicht ein Aufruf des mitgelieferten `restart` Skripts mit dem Namen der Datenbank als Parameter (ohne die Endung *.db!*):

⁴Das Programmierhandbuch liegt in mehreren, inhaltlich identischen Versionen auf dem FTP-Server. Die Versionen unterscheiden sich lediglich in ihrem Dateiformat, etwa HTML oder Postscript.


```
./restart LambdaCore-latest
```

Der Server setzt sich nun kommentarlos als Hintergrundprozess ab und wartet per Voreinstellung am Port 7777 auf die Anmeldung von Clients.

2.1.3 Einrichten eines Clients

Um mit dem Server Kontakt aufzunehmen, reicht ein `telnet localhost 7777`. Schon ist man mit dem MOO verbunden und eine Begrüßungsmeldung erscheint. Mit `connect wizard` bekommt man als Besitzer der Datenbank die notwendigen Rechte, um beliebige Änderungen vorzunehmen. Von diesem Punkt an ist das System vollständig und bietet bereits alle Möglichkeiten eines MOO. Allerdings handelt es sich bei der installierten Kern-Datenbank wirklich nur um ein Grundgerüst aus den elementaren Benutzer- Und Administrationskommandos. Dieses System aus im ASCII-Code repräsentierten Objekten und Telnet-basiertem Client ist zweckmäßig, aber nicht sonderlich attraktiv. Es legt durch die etwas dürftige Dokumentation nicht nur unerfahrenen Benutzern, sondern auch dem Administrator einige Hürden für eine problemlose Benutzung in den Weg. Daher wollen wir für die weitere Arbeit eine hinsichtlich ihrer Benutzerfreundlichkeit ausgereifere, ebenfalls unter der GNU GPL erhältliche, Lösung verwenden.

2.2 enCore

EnCore [4] ist eine erweiterte LambdaMOO-Datenbank. Es enthält neben einem leicht zu bedienenden Web-Client (Xpress), mit dem es Besuchern möglich ist, sich über eine grafische Benutzerführung im MOO zu bewegen, auch einen Satz von Konfigurationswerkzeugen, welche die Administrationsarbeit erleichtern, sowie viele hilfreiche vorgefertigte Objekte und Räume, die als didaktische Werkzeuge eingesetzt werden können. Die Dateien finden sich unter [4]. Unbedingt notwendig ist eigentlich nur die Datei *enCore-latest.tar.gz*. Diese enthält die MOO-Datenbank und den HTML-Baum für den Xpress-Client.

Nachdem man die Datei *enCore-latest.tar.gz* dekomprimiert hat, findet man zwei Komponenten: Die eigentliche Datenbank (*enCore.db*) und einen Baum von HTML- und Java-Dateien, die für die Web-Client-Funktionalität notwendig sind. Die Installationsanweisung sieht vor, die Datenbank im Verzeichnis *~/moo* zu installieren. Von dort lässt sie sich problemlos mit dem Lambda-MOO-Server starten (`./restart ~/moo/enCore`).

Der größte Vorteil des enCore-Systems ist sicherlich der integrierte Browser-Client Xpress. Um dessen Funktionalität zur Verfügung zu stellen, benötigt man natürlich einen Web-Server, der die

Client-Komponenten ins Netzwerk exportieren kann. Hierfür eignet z.B. sich der ebenfalls unter der GNU GPL erhältliche Apache Web-Server [2] in der Version 1.3. Dazu kopiert man den HTML-Baum aus dem enCore-Paket ins Standard-HTML-Verzeichnis des Apache-Servers (häufig ist dies *\$HOME/public_html*) und verbindet sich zunächst mittels `telnet localhost 7777` “ganz normal” mit dem MOO-Server. Nachdem man sich mittels `connect wizard` Administrator-Rechte angeeignet hat, startet man das enCore-Konfigurationswerkzeug mittels `@configure`. Die einzigen Optionen, die an dieser Stelle verändert werden müssen, sind normalerweise 8) Domain Name und 18) Base URL. Die erste Option gibt an, wo die Xpress-Dateien über den Webserver zu erreichen sind (normalerweise ein Pfad der Form *www.site.edu/~foo*), die zweite bezeichnet den Aufbewahrungsort der Dateien im Dateisystem (z.B. *~/public_html/encore*).

Im folgenden wird davon ausgegangen, dass der MOO-Server mit der enCore-Datenbank, wie im vorangegangenen Abschnitt beschrieben, eingerichtet und lokal gestartet worden ist. Direkt nach der Installation existiert im MOO nur ein einziger Raum mit einigen wenigen Objekten. Die Verbindung zum MOO-Server wird über ein `telnet localhost 7777` hergestellt.

2.2.1 Anmelden am MOO

Da bis jetzt noch keine weiteren Benutzer eingerichtet sind, kann sich zunächst nur der Administrator lokal am MOO anmelden. Dies geschieht mit `connect wizard`. Als nächstes erscheint der Hinweis `*** Connected ***` und eine kurze Begrüßungsmeldung. Ab dieser Stelle stehen alle Kommandos zur Benutzung, Veränderung und Verwaltung des MOOs zur Verfügung. Als erstes sollte man sich mit `news 1` die einführenden Informationen des Systems zur Verwaltung anzeigen lassen. Mit `news 2` erhält man Informationen zur Benutzung des MOO. Danach kann man bei Bedarf das Kommando `@help` ausprobieren, mit dem sich detailliertere Hilfetexte zu weiteren Kommandos und Themen abrufen lassen.

2.2.2 Einrichten einer virtuellen Identität

Wer sich häufig im gleichen MOO aufhält, sollte sich einen eigenen Account mit einem Benutzernamen und einem Passwort vom Administrator einrichten lassen (s. hierzu Ausführungen in 3), um zukünftig mit einer eigenen virtuellen Identität im MOO vertreten zu sein. Die Beschreibung der eigenen Person, das Passwort und andere Einstellungen können dann vom Benutzer selbst geändert werden. Das Kommando `@help players` gibt eine Liste mit weiteren Kommandos zu diesem Thema aus.

Zur weiteren Arbeit mit dem MOO ist es sinnvoll, zunächst einen Gastzugang einzurichten. Dieser kann von beliebigen Benutzern ohne Passwort benutzt werden, um sich mit beschränkten Rechten im MOO zu bewegen. Hierzu dient das Kommando `@make_guest <Adjektiv>`, wobei `<Adjektiv>` eine Bezeichnung ist, die dem Wort "Guest" vorangestellt wird:

`@make_guest First` erstellt demnach einen Gastzugang namens "First_Guest". Dieser wird einem beliebigen Besucher zugewiesen, der sich mittels `@connect Guest` am MOO-Server anmeldet. Einem regulären Benutzer, der permanent in der Datenbank gespeichert bleibt und dem man dann auch Rechte, z.B. zum Erstellen von neuen Objekten, übertragen kann, erstellt man mittels

```
@make_player <Name> <E-Mail> <echter Name>
```

einen permanenten Account. Wir benötigen zunächst aber einen mit allen Rechten ausgestatteten Benutzer-Account zur weiteren Arbeit. Also befördern wir unseren eben erschaffenen Benutzer zunächst zum Programmierer:

```
@programmer <Benutzername>
```

Der so beförderte Benutzer `<Benutzername>` kann nun neue Räume und Objekte erstellen. Wir wollen ihm aber noch weitere Rechte übertragen, so dass wir als Administrator arbeiten können. Die Ebene, auf der ein Benutzer alle Rechte besitzt, ist "Wizard". Einem Benutzer mit diesem Status stehen alle Möglichkeiten zur Veränderung des MOO zur Verfügung. Hierzu gehören auch die Kommandos, um Objekte und Benutzer im MOO unwiederbringlich zu löschen. Daher sollte man einen solchen Status nur Personen gewähren, zu denen man beträchtliches Vertrauen hat. Um einen Benutzer `<Benutzername>` zum Wizard zu befördern, erstellt man zunächst einen neuen Benutzer `<Benutzername2>`, wie oben beschrieben, und benutzt dann die folgenden Kommandos:

```
@chparent <Benutzername2> to $wiz
@set <Benutzername2>.wizard to 1
@set <Benutzername2>.public_identity to <Benutzername>
```

2.2.3 Navigation im MOO

Ein MOO besteht aus untereinander verbundenen Räumen. Jeder Raum besitzt Ausgänge zu einem oder mehreren weiteren Räumen. Um sich von Raum zu Raum bewegen, muss man lediglich die Bezeichnung des Ausgangs (wird in eckigen Klammern angezeigt) eingeben.

Um in einen entfernteren Raum zu gelangen, muss man aber nicht immer alle dazwischenliegenden Räume durchqueren. Zur effizienteren Bewegung im MOO existieren weitere Kommandos. Mit `@go <Raum>` kann man sich von jedem Ort des MOO direkt in den durch `<Raum>` bezeichneten Raum begeben. Das Kommando `@join <Benutzer>` befördert einen direkt in den Raum, in dem sich auch `<Benutzer>` aufhält. Schließlich dient `@home` dazu, einen Benutzer direkt in dessen "Heimatraum" zu befördern. Zu Beginn handelt es sich dabei normalerweise um den Startpunkt des MOO, an dem auch Gäste das MOO betreten. Mit dem Kommando `@sethome` kann ein Benutzer den Raum, in dem er sich gerade befindet, als neuen "Heimatraum" setzen.

▷ Weitere Informationen zu diesem Thema erhält man mit `@help movement`.

2.2.4 Benutzen von Objekten

Normalerweise verfügt ein Objekt zumindest über die Verben "look", "examine", "take" und "drop". Damit ist es möglich, ein Objekt zu betrachten, aufzunehmen und wieder abzulegen. Ein `look <thing>` genügt dann, um das Objekt mit Namen `<thing>` zu betrachten, d.h. die mit ihm assoziierte Beschreibung anzeigen zu lassen. Viele Objekte lassen sich auch lesen. Im Raum, in dem man sich direkt nach dem Anmelden wiederfindet, existiert z.B. ein Objekt namens "news", das nach Eingabe von `read news` einen Willkommenstext und einführende Hinweise anzeigt. Mit `examine <thing>` lassen sich detaillierte Informationen über das Objekt `<thing>` inklusive aller definierten Verben anzeigen (mehr zu diesem Thema im Abschnitt 4, der in die Objektprogrammierung einführt).

▷ Weitere Informationen zu diesem Thema erhält man mit `@help manipulation`.

2.2.5 Kommunikation mit Benutzern

Die Kommunikation mit anderen Benutzern lässt sich sehr flexibel handhaben. Das Kommando `say` dient dazu, etwas laut, d.h. für alle im Raum Anwesenden "hörbar" (bzw. lesbar), zu äußern. Mit `whisper` kann man sich gezielt mit einer anderen Person unterhalten. Befindet diese sich in einem anderen Raum, kann `page` verwendet werden, um sie trotzdem zu erreichen. Mittels `gag` lassen sich die Äußerungen bestimmter Personen gezielt stummschalten, falls man sich an deren Gesprächen nicht beteiligen will; `ungag` bewirkt das Gegenteil. Mittels `emote` kann ein Benutzer Äußerungen in der dritten Person machen, die gern genutzt werden, um Gemütszustände anzuzeigen.

Beliebige andere Medien, wie etwa HTML-Dokumente, lassen sich in enCore auf den Bildschirmen aller in einem Raum befindlichen Benutzer mit dem @URL Kommando anzeigen.

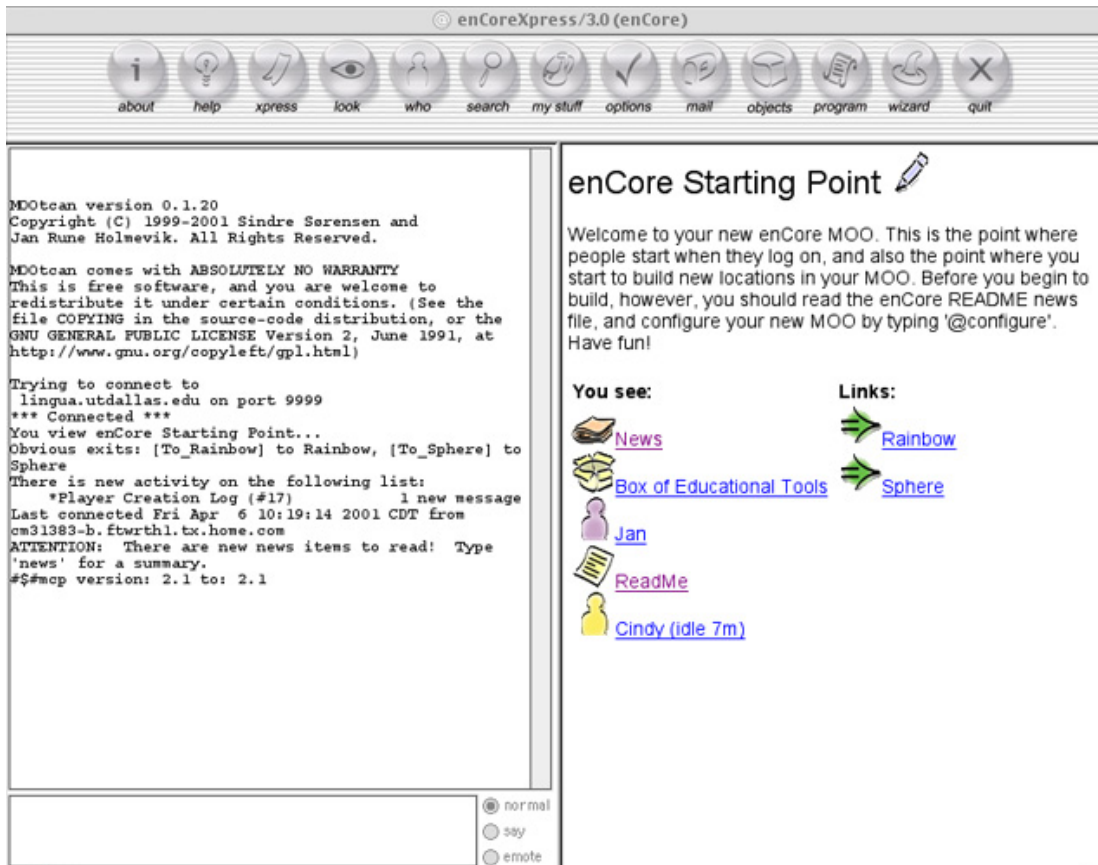
▷ Weitere Informationen zu diesem Thema erhält man mit @help communication.

2.3 Xpress

Der folgende Abschnitt stellt eine kurze Einführung in die Benutzung des im enCore-Paket enthaltenen, grafischen MOO-Client Xpress dar, die nur die grundlegenden Bedienelemente der Software und die Navigation im MOO erläutert. Details zu den administrativen Funktionen der Software finden sich in Abschnitt 3.

Voraussetzung für die Benutzung von Xpress ist, wie in Abschnitt 2.2 beschrieben, die Einrichtung eines Web-Servers. Nachdem man einen JavaScript-fähigen Browser (z.B. Netscape oder Internet Explorer ab Version 4) gestartet hat, öffnet man die entsprechende Web-Server-URL auf Port 7000 (bezogen auf die Beispieleinrichtung aus Abschnitt 2.2 wäre dies *www.site.edu/~foo*). Es erscheint der Login-Bildschirm, wo man sich unter Angabe von User-ID und Passwort am MOO anmelden kann. Insofern ein Gast-Account besteht (s. hierzu Abschnitt 3.2), ist es auch möglich, sich als “Guest” und ohne die Eingabe eines Passwortes ins MOO zu begeben.

Nach der erfolgreichen Anmeldung erscheint der Xpress-Client in einem separaten Browserfenster. Die Anzeige ist dreigeteilt.



Die linke Hälfte der Anzeige wird von einem Telnet-Client eingenommen, mit dem man die aus LambdaMOO bekannten Kommandos zur Navigation im MOO (wenn Button “normal” aktiviert) und zur Kommunikation mit anderen Benutzern (sofern Button “say” oder “emote” aktiviert”) verwenden kann (s. 2.2.5).

Im oberen Bereich des Hauptfensters befindet sich eine Icon-Leiste mit den wichtigsten Funktionen zur Benutzung des MOO. Die hier angezeigten Icons unterscheiden sich in Abhängigkeit davon, mit welchen Rechten der Benutzer angemeldet ist (zur Rechteverwaltung und dem Anlegen neuer Benutzer siehe Abschnitt 3.2):



about Gibt Versions- und Statistik-Informationen über den laufenden enCore-Server aus.

help Startet den Help-Browser von enCore. Dieser bietet Texte zu verschiedenen Themen, die

LambdaMOO ebenso wie die enCore-spezifischen Erweiterungen abdecken. Zur einfacheren Erschließung der Texte existiert auch eine Suchfunktion.

xpress Der “Xpress Navigator” vereinfacht die Navigation im MOO, indem häufig besuchte Räume als “Bookmarks” gespeichert werden können. Eine Notizbuchfunktion ist ebenfalls über diesen Menüpunkt erreichbar.



look Zeigt Bilder, Texte und Objekte des Raumes an, in dem sich der Besucher momentan befindet. Diese Funktion kann benutzt werden, um beispielsweise von einer Objekte- zur Raumbeschreibung zurück zu schalten.

who Der “Who Browser” zeigt alle im MOO befindlichen Personen mit den Räumen, in denen sie sich befinden, an. Ein Klick auf den Namen der Person zeigt die entsprechende Beschreibung inklusive E-Mail-Adresse an. Klickt man auf den Link unter dem Eintrag “Location”, so wird man direkt in den Raum befördert, in dem sich der jeweilige Benutzer befindet.














































search Mit dieser Funktion lässt sich nach Objekten oder Personen im MOO suchen.

Inventory for ducke-wiz

Refresh inventory Close inventory

Quota usage:  450,384 bytes free.  49,616 bytes used. Total objects: 35.

1 2 3 4 5 [next >>]

Object	Location	Actions
 To_MaMOOth_Startpunkt (#205)	Nowhere	
 To_Ausstellung (#204)	Nowhere	
 Ausstellung (#203)	Nowhere	  
 Steinwerkzeuge (#202)	Im Inneren der Höhle (#171)	     
 Vitrine mit Steinwerkzeugen (#201)	Ausstellung (#203)	    
 Handabdruck (#200)	Kultraum (#191)	    
 Tierskulptur (#199)	Kultraum (#191)	    
 Frauen-Skulptur (#198)	Kultraum (#191)	    
 Steinwerkzeug (#197)	Im Inneren der Höhle (#171)	    

my stuff Diese Funktion zeigt alle im Inventar eines Benutzers befindlichen Gegenstände an (s. Abbildung). Dazu gehören solche, die sich mit dem Benutzer bewegen (Eintrag “Nowhere” in der Spalte “Location”) ebenso wie in einem Raum abgelegte Objekte eines Benutzers. Die auf ein Objekt anwendbaren Funktionen sind in der Spalte “Actions” als Icons repräsentiert (siehe Beschreibung der Icons weiter oben in diesem Abschnitt).

options Über dieses Icon sind alle Einstellungen zugänglich, um die Eigenschaften der eigenen virtuellen Identität (Name, E-Mail, Geschlecht...) festzulegen.

mail Bietet Zugang zum MOO-internen Mail-System, das ein Verschicken von Nachrichten zwischen Benutzern eines MOOs erlaubt.

Die folgenden Funktionen benötigen einen bestimmten Benutzer-Status. Andernfalls wird das

entsprechende Icon gar nicht erst angezeigt (siehe Abschnitt 3 zum Einrichten von Benutzer-Accounts und zur Rechtevergabe).

objects Erlaubt das Editieren bestehender und das Anlegen neuer Objekte und Räume. Die Benutzung dieser Funktion erfordert mindestens Builder-Status.

program Insofern man über einen Programmier-Status verfügt, kann man diese Funktion nutzen, um in einem einfach zu bedienenden Editor Programme für das MOO zu schreiben (siehe Abschnitte 4 und 4.3).

wizard Über dieses Icon, das nur einem Benutzer mit Wizard-Status zugänglich ist, lassen sich zahlreiche Aspekte der MOO-Administration erreichen. Dazu gehört eine Funktion zur Erstellung von Benutzer-Accounts und zur Zuweisung von Benutzerrechten ebenso wie Einstellmöglichkeiten für das grundsätzliche Erscheinungsbild des MOO und eine Möglichkeit neue Basisobjekte zu kreieren, die von Benutzern (mit mindestens Builder-Status) erstellt und verändert werden können.

quit Meldet den Benutzer am MOO ab. Die virtuelle Identität verlässt das MOO.

2.3.1 Navigation

Die rechte Hälfte des Bildschirms zeigt eine grafische Repräsentation des aktuellen Raumes mit allen darin befindlichen Objekten, Personen und Übergängen zu anderen Räumen. Durch Anklicken der entsprechenden Icons kann der Benutzer Objekte betrachten und manipulieren und sich durch die Räume bewegen.



(Person) Dieses Icon repräsentiert eine Person. Ein Klick darauf zeigt weitere Informationen, wie etwa den realen Namen und die E-Mail Adresse, an.



(Ausgang) Dieses Symbol repräsentiert einen Zugang zu einem anderen Raum. Ein Klick auf dieses Icon befördert den Benutzer in den entsprechenden Raum.



... **(Objekte)** In den einzelnen Räumen befinden sich Objekte unterschiedlicher Art. Sie werden durch verschiedene Icons repräsentiert (im einzelnen, s. Abschnitt 4.1). Ein Klick auf eines dieser Icons fördert Text- und Bildbeschreibung sowie evtl. Manipulationsmöglichkeiten (s. Abschnitt 2.3.2) hervor.

2.3.2 Benutzung von Objekten

Die am häufigsten zur Objektmanipulation benötigten Funktionen sind in Xpress auch über die grafische Benutzeroberfläche erreichbar. Sie werden durch Icons repräsentiert, die in verschiedenen Zusammenhängen (z.B. in der Inventarliste) vorkommen. Zahlreiche Objekte mit besonderen Eigenschaften bieten komplexere Manipulations- und Benutzungsmöglichkeiten (s. z.B. die in Abschnitt 4.1 besprochenen enCore-Objekte), die nur über Kommando-Eingabe, wie in 2.2.4 besprochen, verfügbar sind.



examine Zeigt Informationen über ein Objekt an, sowie eventuelle Manipulationsmöglichkeiten. Die angezeigten Verben können auf das Objekt angewendet werden, indem man sie im Telnet-Fenster (linke Seite des enCore-Hauptfensters), gefolgt vom Objektnamen, eingibt.



take Befördert ein Objekt aus einem der Räume ins persönliche Inventar des Benutzers. Das persönliche Inventar kann nur eine begrenzte Anzahl von Objekten aufnehmen (siehe 3.2) und ist über die Icon-Leiste (s. weiter unten) zugänglich.



drop Entfernt ein Objekt aus dem Inventar (s.o.) des Benutzers und plaziert es in dem Raum, in dem dieser sich gerade befindet.



view Öffnet ein neues Fenster mit Funktionen zur Manipulation eines Raumes oder Objektes. Sofern entsprechende Berechtigung besteht, können hier z.B. die Textbeschreibung und die zugeordnete Grafik für ein Objekt angepasst werden.



lock Verschließt ein Objekt oder einen Raum vor dem Zugriff anderer Benutzer. Auf ein Objekt angewendet bedeutet dies, dass kein Benutzer außer dem Besitzer des Objekts dieses mittels "take" aufnehmen darf. Angewendet auf einen Raum bedeutet es, dass dieser von keinem Benutzer, außer dem Besitzer des Raumes, betreten werden darf.



unlock Gibt ein zuvor verschlossenes Objekt wieder frei.



recycle Entfernt ein Objekt vollständig aus dem MOO. Damit wird es unwiederbringlich gelöscht und der zugeordnete Speicherplatz in der Datenbank wieder frei gegeben.



email Mit dieser Funktion kann man sich die Beschreibung eines Objekts an die eigene E-Mail-Adresse zuschicken lassen. Dies funktioniert mit Objekten, die lesbar sind, wie etwa solchen vom Typ *Note* (s. Beschreibungen in 4.1).

3 Administration

Der Begriff “Administrator” bezeichnet die Person, die für Installation und Wartung des MOO zuständig ist. Der Administrator besitzt alle Rechte, um das MOO beliebig zu verändern. Im Gegensatz dazu sind mit “Benutzer” die Besucher des MOO gemeint, die sich in der Regel über das Internet am MOO anmelden und dort mit beschränkten Rechten agieren. Ein Benutzer kann in der Regel nicht einfach beliebige Änderungen am MOO vornehmen. Er kann aber unter Umständen das MOO um eigene Räume und Objekte erweitern. Ob ihm solche Möglichkeiten offenstehen, entscheidet der Administrator von Fall zu Fall.

3.1 Einstellungen für enCore/Xpress

Das optische Erscheinungsbild des enCore-Xpress-Clients lässt sich in Grenzen anpassen. Die hierfür zur Verfügung stehenden Einstellungen finden sich unter dem Icon “xpress” in der Iconleiste des Xpress-Clients.

3.2 Benutzerverwaltung

Xpress bietet einen Dialog zum Erstellen und Befördern von Benutzern, der die in Abschnitt 2.2.2 besprochenen Schritte erheblich vereinfacht. Bei der Beförderung eines Benutzers zum Wizard ist aus den dort bereits erwähnten Gründen größte Vorsicht geboten!

The screenshot shows the 'Xpress MOO Administration' window. At the top, there is a navigation bar with buttons for 'Accounts', 'Bookmarks', 'Generics', 'MOTD', 'enCore Settings', and 'Close'. Below this is the 'Character Account Administration' section, which includes a 'Single Account' button, a 'Multiple Accounts' button, and an 'Update Account:' label. A search field with a 'Find' button is present. The 'Character Class:' section has radio buttons for 'Guest', 'Player', 'Builder', 'Programmer', and 'Wizard'. To the right, there is a text box with instructions: 'Select a character class and enter name, email address and real name for the new account to be created below. If you are creating a new guest account, you only need to fill in the name field.' Below the instructions are three input fields labeled 'Name', 'Email', and 'Real Name'. A 'Create New Account' button is at the bottom.

Ein Benutzer kann einige Einstellungen bezüglich seines Accounts (E-Mail Adresse, Passwort, etc.) mit dem Kommando `@preferences` selbst einstellen (dieselben Einstellungen lassen sich auch über das Icon “options” im Xpress erreichen – s. Abschnitt 2.3).

Wenn ein Benutzer ein Objekt (bzw. einen Raum) erstellt, so benötigt dieses natürlich Speicherplatz in der Datenbank. Bei vielen im MOO aktiven Benutzern und begrenzten Ressourcen kann dieser knapp werden. Daher verfügt LambdaMOO über einen einfachen, “Quota” genannten, Mechanismus, der den einem Benutzer zur Erstellung eigener Objekte zur Verfügung stehenden Speicherplatz begrenzt. Während sich diese Quota bei LambdaMOO in Objekten ausdrückt, arbeitet enCore-MOO mit einem Byte-basierten System. Die voreingestellte Quota von ca. 50.000 Bytes zur Erstellung von Objekten ist schnell erreicht. Mehr erhält ein Account mit folgendem Kommando:

```
@quota <player> is [public] [+]<number> [<reason>]
```

Um also beispielsweise den Benutzer “Fooward” mit einem halben MB Speicherplatz auszustatten, genügt folgendes Kommando:

```
@quota Fooward is 524288
```

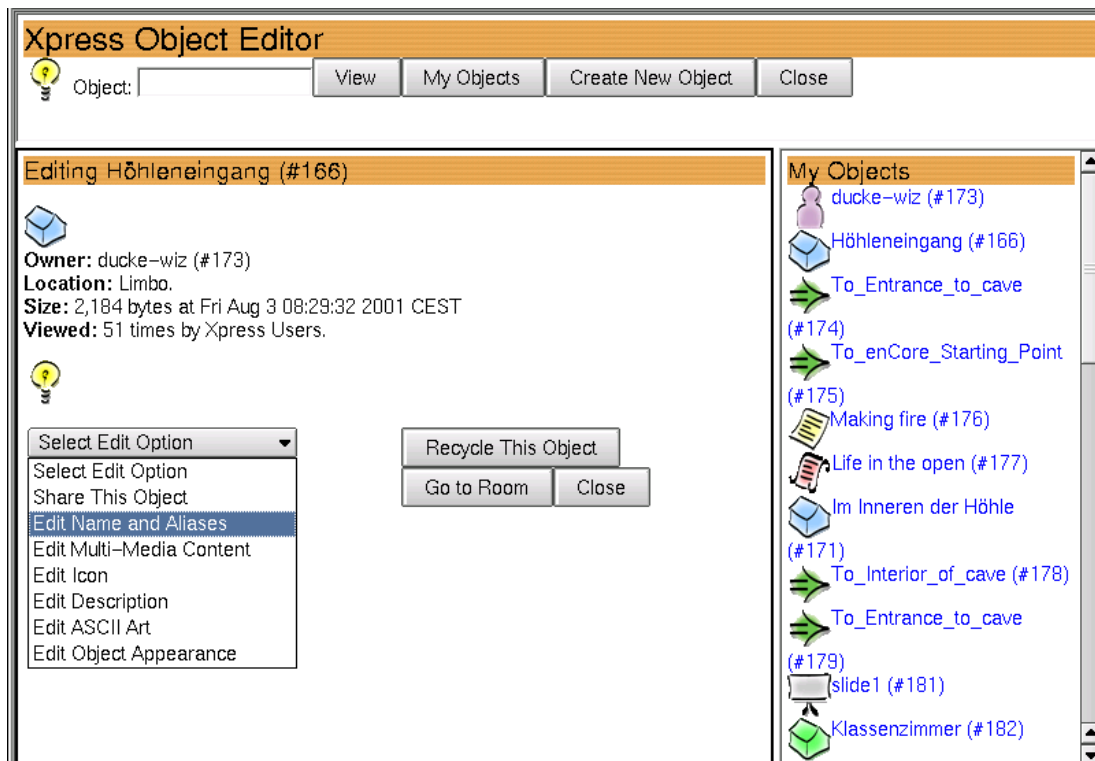
4 Gestaltung und Programmierung

Im folgenden soll nun erläutert werden, wie das MOO durch Hinzufügen neuer Objekte und Räume erweitert werden kann. Auch zu diesem Zweck bietet der Xpress-Client eine Palette an komfortabel zu nutzenden grafischen Dialogen. Voraussetzung zu deren Benutzung ist, dass man sich mit Wizard-Status am System angemeldet hat (s. hierzu die Abschnitte 2.2.2 und 2.3). Obwohl der Schwerpunkt der folgenden Erläuterungen auf der Benutzung von Xpress liegt, werden, für den Fall, dass der Xpress-Client nicht zur Verfügung steht, auch die entsprechenden enCore-Kommandos, die im Telnet-Client zur Verfügung stehen, behandelt.

4.1 Objekte und Räume

Der Xpress-Client besitzt einen komfortablen Dialog zur Erstellung und Verwaltung von Objekten. Dieser erscheint, wenn man das Icon “objects” auf der Icon-Leiste anklickt. Ein Klick auf “My Objects” fördert eine Liste der Objekte im Besitz des Benutzers zutage. Wählt man in der erscheinenden Liste ein Objekt aus, kann man seine Einstellungen ansehen und verändern. Nach einem Klick auf “Create New Object” erscheint die Liste der enCore-Basisobjekte, eingeteilt in drei Kategorien. Man kann nun den gewünschten Eintrag auswählen, um ein neues Objekt dieses Typs zu erzeugen. Will oder kann man Xpress nicht zur Objekt-Erstellung benutzen, lautet das äquivalente Kommando @create.

Hat man ein neues Objekt erzeugt, landet dieses zunächst im Inventory. Um es in einem Raum zu plazieren, begibt man sich dorthin, klickt auf das “My Stuff”-Icon in der Icon-Leiste und lässt das Objekt fallen (Icon roter, nach unten zeigender Pfeil). Das gleiche leistet das Kommando drop <Objekt>.



4.1.1 Basisobjekte und -räume von enCore

Die hier vorgestellten Basisobjekte werden benutzt, um das MOO mit interessanten Räumen und Gegenständen zu gestalten. Die MOO-Programmiersprache (s. nächster Abschnitt) erlaubt es, aus diesen Basisobjekten neue Objekte mit erweiterter Funktionalität abzuleiten. Die Zahlen in Klammern geben die eindeutige Objektzahl an (s. Abschnitt 4).



Thing (#5) Das Basisobjekt jedes MOO hat keinerlei besondere Eigenschaften. Benutzer können es betrachten und, sofern entsprechende Berechtigung vorhanden, an eine andere Stelle bzw. ins Inventar eines anderen Benutzers transportieren. Einem solchen Objekt lässt sich zur Beschreibung ein Text zuweisen (“Edit Description” in den Objekt-Einstellungen). Bei Benutzung von Xpress können außerdem weitere Medien (z.B. ein Bild) zugewiesen werden (“Edit Multi-Media Content”).



Container (#8) enthält andere Objekte. Um Zugriff auf die in einem Container enthaltenen Objekte zu bekommen, muss der Benutzer diesen mit `open <Container-Objekt>`

zunächst öffnen. Um Gegenstände in einem Container unterzubringen, bzw. aus diesem zu entnehmen, existieren die Verben “get from” und “put in”.



Note (#9) Ein einfaches Objekt, um Texte zu speichern. Es versteht die (selbsterklärenden) Verben “read”, “write”, “erase”, “encrypt”, “decrypt” und weitere. Dieses Objekt eignet sich, um kurze, schwach strukturierte Texte vorzubereiten und anzuzeigen. Für ambitioniertere Vorhaben, wie etwa Online-Vorträge, existieren erweiterte enCore-Objekte (s. nächsten Abschnitt). Um den Text zu editieren, der angezeigt wird, wenn ein Benutzer auf einen Notizzettel klickt, wählt man die Option “Edit note text” im Objekteditor.



Room (#3) Ein neuer Raum wird mit dem Kommando `@dig <Raumname>` erzeugt (z.B. `@dig "Entry Hall"`). Anschließend können mit demselben Kommando Verbindungen zu anderen Räumen hergestellt werden (z.B. `dig north,n to "The North Pole"`). EnCore bietet erweiterte Räume, die sich besonders zur Verwendung als Vorlesungs- und Unterrichtsräume eignen (s. nächsten Abschnitt).

4.1.2 Didaktische Objekte und Räume in enCore

Die Präsentation umfangreicherer Inhalte erfordert eventuell etwas mehr Organisation. Zu diesem Zweck bietet enCore eine Reihe vorgefertigter, didaktischer Objekte (Klasse *Educational Objects*), die hier kurz besprochen werden sollen. In enCore existieren spezielle Räume, wie etwa der *Moderated Room*, innerhalb derer die Kommunikationsströme zwischen Benutzern stärker reguliert werden können. Die hier vorgestellten Objekte sind sehr flexibel einsetzbar. Es lohnt sich, sich mit ihrer Benutzung vertraut zu machen.

▷ Weitere Informationen zu den hier vorgestellten Objekten und Räumen erhält man im Xpress-Hilfesystem (“help” in der Iconleiste) oder durch Eingabe von `help #<Objektnummer>`. Falls kein Eintrag in der Hilfefunktion vorhanden ist, fördert `examine #<Objektnummer>` weitere Informationen zutage. Dieses Kommando produziert auch eine Liste aller vom Objekt implementierten Verben, die i.d.R. weitere Rückschlüsse auf dessen Funktionalität erlaubt.



Recorder and Player (#115) Dieses Objekt kennt die Verben “start” und “stop”, mit denen “Gespräche” im MOO mitgeschnitten werden können. Mit `play` lassen sich Aufnahmen wieder abspielen. Die Wiedergabegeschwindigkeit lässt sich mittels `setdelay` ein-

stellen (z.B. `setdelay MyRecorder to 1`, um eine Geschwindigkeit von einer Sekunde einzustellen).



Bot (#100) Ein *Bot* kann in einem Raum platziert und so programmiert werden, dass er auf bestimmte Muster oder Wörter vordefinierte Antworten gibt. Sehr einfach ist es, einen *Bot* so einzustellen, dass er auf ein von einem Benutzer geäußertes Wort reagiert. Wenn man möchte, dass der *Bot* “MyBot” auf bestimmte Schlüsselwörter reagiert, kann man mit `addword MyBot` die Eingabe für neue Schlüsselwörter starten. Man gibt nun ein Schlüsselwort und eine entsprechende Zeile als Reaktion darauf ein. Danach muss der *Bot* noch mit `activate MyBot` aktiviert und mit “hi” angesprochen werden. Flexibler ist der Einsatz von regulären Ausdrücken mittels `@addpat <Botname>`, zu deren Verwendung `help regular` Hilfestellung gibt.



Slide Projector (#101) Mit einem solchen Objekt kann ein Vortragender Textseiten anzeigen, die dann für alle im Raum anwesenden Personen sichtbar sind. Damit lässt sich eine einfache Online-Vorlesung aus maximal 10 “Dias” (d.h. Textseiten mit max. 22 Zeilen) gestalten. Ein “Dia” erzeugt man mittels `@notedit <Projektorname>.s[1-10]`. Nachdem man ein neues Objekt dieses Typs erzeugt hat, lässt sich mit `show 1 on <Projektorname>` eine Einführung zur Benutzung anzeigen, die über weitere Möglichkeiten, z.B. das Einbinden von externen Textdateien, informiert.



Web Projector (#113) Vielseitiger als der *Slide Projector* ist der *Web Projector*, mit dem sich Web-Seiten, die allen in einem Raum anwesenden Besuchern gezeigt werden sollen, organisieren und dann der Reihe nach anzeigen lassen. Die Benutzung erfolgt analog zum *Slide Projector*; die “Dias” bestehen hier aber anstelle von anzuzeigendem Text aus URLs. Auch die Anzeige funktioniert etwas differenzierter. Mit `display` lassen sich URLs öffentlich präsentieren, mit `view` bekommt man deren Inhalt zunächst nur selbst zu sehen. Die Liste der zur Anzeige vorbereiteten URLs lässt sich mit `peek <Projektorname>` aufrufen.



Recitable Note (#108) Das Objekt *Recitable Note* ist eine Art Skript, das eine einfache Möglichkeit bietet, kurze Texte vorzubereiten und anschließend zu rezitieren. Um eine Zeile Text hinzuzufügen, genügt das Kommando `write <Textzeile> on <Skriptname>`.

Nach Eingabe von `recite <Skriptname>` wird das Skript Zeile für Zeile “verlesen”, wobei zwischen den Zeilen eine Pause eingelegt wird, deren Dauer sich mit `@setdelay <Skriptname> to <Sekunden>` setzen lässt. Das Verlesen des Skripts lässt sich mit den Kommandos `pause <Skriptname>` und `stop <Skriptname>` jederzeit unterbrechen und mit `recite <Skriptname>` wieder fortsetzen bzw. von vorne beginnen.



Lecture (#109) Die tastaturbasierte Online-Kommunikation im MOO hat ihre Grenzen, wenn es darum geht, längere Gedankengänge zu äußern, die dann nicht nur in flüssigem Stil, sondern auch möglichst (Tipp-)fehlerfrei geäußert werden sollen. Um hier Abhilfe zu schaffen, existiert das Objekt *Lecture*, das vorformulierten Text Seiten- oder Zeilenweise, mit definierbaren Pausen, anzeigen kann. Der Text wird mittels `@notedit <Vorlesungsname>.text` vorbereitet und mit `deliver <Vorlesungsname>` online vorgetragen.



Note Board (#136) Dieses Objekt repräsentiert eine Pinwand, an die neue Notizen gehängt werden können. Welche Benutzer Nachrichten anhängen, ändern oder entfernen können, lässt sich einstellen (Modus “Private”, “Restricted” oder “Public”, einstellbar über das Verb “configure”). Notizen können an die Pinwand angefügt (`post <Notiz> on <Pinwand>`) und entfernt (`remove <Notiz Nr.> from <Pinwand>`) werden. Um interaktiv eine neue Notiz zu erstellen und gleich an die Pinwand zu heften, existiert das Kommando `post new on <Pinwand>`.



Classroom (#99) Die sehr gleichberechtigte Kommunikationsstruktur des MOO ist nicht für alle Zwecke angemessen. In vielen Situationen wäre eine Hierarchie angebrachter, die es einer oder mehreren Personen erlaubt, die Diskussion zu steuern, indem nicht jede Äußerung gleichberechtigt alle Benutzer erreicht. Zu diesem Zweck existieren in enCore-MOO die Räume vom Typ *Classroom* und *Moderated Room*, der weiter unten beschrieben wird. Ein Klassenzimmer enthält zunächst nur einen Lehrerschreibtisch, ein schwarzes Brett, eine Tafel, eine Uhr und einen großen Tisch für die “Schüler”. Um den Raum mit weiteren Tischen, Regalen, Stühlen auszustatten, dient das interaktive Kommando `@addfurniture`, während `@rmfurniture` das Gegenteil bewirkt. Die meisten der so erzeugten Objekte kennen die Verben “look”, “sit” und “put”. Benutzer, die an einem Tisch sitzen (z.B. mittels `sit Big Table`), können nur von anderen, am gleichen Tisch sitzenden Benutzern “gehört” werden. Objekte, die mittels `put` auf einem bestimmten Tisch ab-

gelegt worden, können nur von den dort sitzenden Benutzern gesehen werden. Um sich an alle Anwesenden wenden zu können, benutzt man die Kommandos `stand` und `speakup`.

Die Vorbereitung und Durchführung einer Unterrichtseinheit erfordert eine wenig organisatorische Vorarbeit. Zunächst wird eine Unterrichtseinheit mit dem Kommando `@mkclass <Bezeichnung>` beschrieben. Soll ein Benutzer, der keinen Wizard-Status hat, mit der Durchführung des Unterrichts beauftragt werden, muss dieser zuvor mittels `@authorize <Benutzer>` als Lehrer autorisiert werden. Mit `@setup <Bezeichnung>` wird die Unterrichtseinheit eröffnet, woraufhin neue Schüler mit dem Kommando `@register <Benutzername>` angemeldet werden können.

Zahlreiche weitere Kommandos existieren zur Benutzung des Rauminventars und zur Steuerung der Kommunikationsströme zwischen den im Raum Anwesenden. Die Tafel lässt sich mit den Kommandos `writeb` und `eraseb` beschreiben bzw. löschen, die Kommandos `door open|closed` dienen dazu, unregistrierten Benutzern den Zugang zum Klassenzimmer zu erlauben oder zu versperren. Mit `@session on` erzeugt man einen Eintrag in der Ausgabe des Kommandos `@who` (bzw. im Who-Browser des Xpress-Clients), der alle Benutzer darüber informiert, dass der Unterricht begonnen hat

▷ Eine Einführung in die Benutzung erhält man, wenn man innerhalb eines Raumes vom Typ *Classroom* die Kommandos `@tutorial` oder `help here` eingibt.



Moderated Room (#98) Der Raumtyp *Moderated Room* kann zwei verschiedene Zustände annehmen: als “offener” Raum bestehen keine Unterschiede zum gewöhnlichen LambdaMOO-Objekt *Room*. Als “moderierter” Raum stehen zahlreiche weitere Kommandos zur Verfügung. Der Raum funktioniert dann im Sinne eines Konferenzraumes, wobei nur die Personen auf der Rednerbühne (die Redner) im ganzen Raum gehört werden, aber von der Kommunikation der übrigen Anwesenden (quasi des “Publikums”) abgeschottet sind. Während sich im Publikum stehende Personen untereinander mit den gewohnten Kommandos (`talk`, `emote` etc.) unterhalten können, müssen Anfragen an die Rednerbühne mit dem besonderen Kommando `ask` gestellt werden. Eine solche Anfrage gelangt dann zunächst an den Moderator, der sie gegebenenfalls an andere auf der Rednerbühne befindliche Personen weiterleiten kann. Insgesamt ergibt sich damit eine Kommunikationsumgebung, die einer moderierten Pressekonferenz ähnelt.

Bevor diese Möglichkeiten zur Verfügung stehen, muss der Status des Raumes mit `Format panel` auf “moderiert” gesetzt werden. Ein Benutzer kann dann von einem anderen Be-

nutzer mit Wizard-Status als Moderator autorisiert bzw. entlassen werden. Dies geschieht mit dem Kommando `Authorized add|remove <Benutzer>`. Der so autorisierte Benutzer kann sich anschließend mit dem Kommando `Moderator` selbst zum aktiven Moderator des Raumes machen und mit dem Kommando `Questions` die Bearbeitung eingehender Fragen in einer Warteschlange beginnen. Nachdem alle als Redner designierten Benutzer die Rednerbühne betreten haben, kann diese mit `Lock on` vor weiterem Zutritt verschlossen werden. Ein Programm für die Konferenz kann mit dem Kommando `Program setup` erstellt und mit `Program erase` wieder gelöscht werden.

Dem Publikum stehen ebenfalls einige nützliche Kommandos zur Verfügung. Mittels `Headset on|off` lässt sich steuern, ob nur von den Redern gemachte Äußerungen oder die gesamte Kommunikation im Raum angezeigt werden soll und mittels `Program` kann man sich das aktuelle Konferenzprogramm anzeigen lassen.



Web Page (#111) Das Objekt “Web Page” ist ein Raum, der mit beliebigem HTML-Code gestaltet werden kann. Dazu schreibt man den HTML-Code in die Beschreibung des Raumes. Im Raum befindliche Personen und Objekte sowie Verbindungen zu anderen Räumen werden vom Xpress-Client am Ende der Seite angezeigt.

▷ In enCore existieren weitere, hier nicht detailliert besprochene Objekte: **Video tape (#105)**, **Video camera (#102)**, **VCR (#103)**, **TV (#104)**, **Recording System (#96)**, **Room Directory (#135)**, **Generic Public Writable Note (#134)**. Diese Objekte sind nicht über Xpress erreichbar, können aber mittels `@create` erzeugt werden. Weitere Informationen zu diesen Objekten und Räumen erhält man im Xpress-Hilfesystem (“help” in der Iconleiste) oder durch Eingabe von `help #<Objektnummer>`. Sollte sich kein Eintrag in der Hilfefunktion finden, lassen sich weitere Informationen in der Regel mit `examine <Objektnummer>` finden.

4.1.3 Objektverwaltung

Jedes Objekt bzw. jeder Raum hat einen Besitzer. Dabei handelt es sich um den Benutzer, der das Objekt bzw. den Raum erschaffen hat. Der Besitzer eines Objektes/Raumes kann dieses mit anderen Benutzer teilen. Die Objekt Nummer eines Benutzers erfährt man am einfachsten, in dem man ihn sich im Who Browser (Icon “who” in der Icon-Leiste) herausucht und dann mittels der “view”-Funktion die detaillierte Beschreibung anzeigen lässt. Das Kommando `@examine` gibt ebenfalls den Besitzer eines Objektes aus.

Sehr störend ist die Eigenschaft des Objekt-Editors, direkt eingegebene Umlaute zu verschlucken, so dass sie im MOO verloren gehen und umgekehrt HTML-kodierte Umlaute im Editor wieder in direkte Umlaute umzuwandeln, so dass sie, wenn z.B. ein Text ein zweites Mal editiert werden soll, alle wieder in HTML-Kodierung überführt werden müssen. Es hat sich daher als zweckmäßig erwiesen, längere Texte mit einem externen Texteditor zu bearbeiten und in separaten Dateien abzuspeichern. Der Texteditor bietet auch den Vorteil, dass Sonderzeichen sich im ganzen Dokument mit einem "Suchen-und-Ersetzen" durch die entsprechenden HTML-Codes ersetzen lassen.

4.2 Die MOO-Programmiersprache

Trotz der Vielzahl der vorgefertigten Objekte, die v.a. enCore bereits eine große Vielseitigkeit verleihen, wird ein ambitionierter MOO-Administrator früher oder später an einem Punkt anlangen, an dem von ihm selbst oder anderen Benutzern Wünsche hinsichtlich der Gestaltung des MOO geäußert werden, denen mit den vorgefertigten Objekten nicht oder nur sehr umständlich entsprochen werden kann. Um dem System eine höhere Flexibilität zu geben, verfügt LambdaMOO⁵ in der Grundausstattung über eine objektorientierte Programmiersprache. Mit ihrer Hilfe können neue Objekttypen mit erweiterten Eigenschaften und Manipulations- bzw. Interaktionsmöglichkeiten implementiert werden. Der Xpress Web-Client bietet zu diesem Zweck auch einen grafischen Editor, auf den in einem separaten Abschnitt kurz eingegangen wird.

▷ Eine vollständige Beschreibung der MOO-Programmiersprache würde den Rahmen dieser Arbeit sprengen. Eine offizielle, vollständige Referenz findet sich auf dem FTP-Server des LambdaMOO-Projekts [14]. Weitere Informationen finden sich auch in der Hilfefunktion nach Eingabe von `@help programming` bzw. im Help-Browser von Xpress.

MOO-Programme werden in der Datenbank gespeichert und vom Server zur Laufzeit interpretiert. Die MOO-Programmiersprache ist auf leichte Erlernbarkeit optimiert und verfügt über einen recht überschaubaren Sprachumfang. Die zur Verfügung stehenden Datentypen sind Integer, Real, String, Object (die im MOO sichtbaren Objekte), Error (während der Programmausführung vom Server erzeugte Fehlercodes) und List (rekursiver Datentyp, der alle anderen Datentypen inkl. List enthalten kann).

Der wichtigste Datentyp der MOO-Programmiersprache sind Objekte (Typ "Object"). Objekte

⁵In dieser Programmiersprache erstellte Objekte lassen sich natürlich auch in enCore-MOO verwenden, das ja auf LambdaMOO basiert. EnCore-MOO beinhaltet in dieser Hinsicht keine Änderungen gegenüber LambdaMOO. Im folgenden wird der Einfachheit halber daher nur von LambdaMOO die Rede sein.

repräsentieren alle im MOO für die Benutzer sichtbaren Personen, Gegenstände (im weitesten Sinne) und Räume. Jedes Objekt besitzt eine eindeutige, ganzzahlige Nummer. Ein Objekt wird mit der Funktion `create()` erzeugt und mittels `destroy()` wieder aus der Datenbank entfernt. Jedes Objekt besteht aus den folgenden grundlegenden Attributen:

Attribute Ein Flag (`false/true`), das anzeigt, ob es sich bei dem Objekt um eine Person, d.h. einen Benutzer des MOO handelt; ein Zeiger auf das Elternobjekt und eine Liste der Kinderobjekte. Da es sich bei der LambdaMOO-Programmiersprache um ein objektorientiertes Modell handelt, können Objekte aus bereits vorhandenen abgeleitet werden, um deren allgemeine Eigenschaften zu übernehmen (Vererbung) und evtl. zu erweitern oder zu verändern.

Eigenschaften sind Platzhalter mit zugeordneten Namen, in denen beliebige Werte in einem der unterstützten Datentypen (`Integer`, `String`...) abgelegt werden können. Einige Eigenschaften sind für jedes Objekt vorgegeben, dazu gehören z.B. der Name des Objekts, dessen Eigentümer und der Ort, an dem es sich befindet ("`name`", "`owner`", "`location`"). Weitere Eigenschaften sind frei definierbar. Ein Stein könnte z.B. die Eigenschaften "`Größe`", "`Gewicht`" und "`chemische Zusammensetzung haben`". Jede definierte Eigenschaft besitzt einen Eigentümer (normalerweise der Benutzer, der das Objekt instantiiert hat) und einen Satz von Zugriffsrechten, über den sich regeln lässt, in welcher Weise das Objekt für andere Benutzer benutz- oder manipulierbar ist.

Verben sind benannte MOO Programme, die mit einem Objekt assoziiert sind. Verben sind die Kommandos, die ein Benutzer auf ein Objekt ausführen kann (z.B. "`read`", "`take`", "`drop`", etc.). Verben können aber auch von anderen MOO-Programmen ausgeführt werden. Verben entsprechen damit den Funktionen (Prozeduren, Methoden) gängiger objektorientierter Programmiersprachen. Genau wie Eigenschaften, verfügen auch Verben über einen Eigentümer und Benutzerrechte. Damit der Parser⁶ des Servers die Benutzung der Verben für den Benutzer "natürlich" machen kann, verfügt jedes Objekt über drei syntaktische Argumente, die für das direkte und das indirekte Objekt sowie die Präposition stehen (bsp. `put box behind screen`).

Das Kommando `@verb <Objektname>:<Verbname>` fügt dem Objekt `<Objektname>` ein neues Verb mit der Bezeichnung `<Verbname>` hinzu. Ein Verb kann auch mehrere Bezeichnungen er-

⁶Der Parser ist derjenige Teil des Servers, der die vom Benutzer im MOO eingegebenen Kommandos in ihre syntaktischen Bestandteile zerlegt und interpretiert, d.h. die entsprechenden Programme mit den entsprechenden Argumenten ausführt. Seine Funktionsweise ist ebenfalls im Programmierhandbuch detailliert beschrieben.

halten. Die Syntax lautet dann `@verb <Objektname>:"<Verbname1> <Verbname2> <...>"`, wobei die unterschiedlichen Bezeichnungen durch Leerzeichen separiert und in Anführungszeichen gesetzt werden. Verben können, wie oben erwähnt, mit Objekten und Präpositionen verknüpft werden. Das erweiterte Kommando lautet `@verb <Objektname>:<Verbname> <Direktobjekt> [<Präposition> [<Indirektobjekt>]]`. Die Argumente `<Direktobjekt>` und `<Indirektobjekt>` können die Werte "none", "this" oder "any" annehmen, deren Bedeutungen im MOO-Programmierhandbuch erläutert werden. Als `<Präposition>` gelten "none", "any" oder eine der Präpositionen, die nach Eingabe von `help prepositions` aufgelistet werden. Alle drei Argumente werden per Voreinstellung auf "none" gesetzt.

▷ Weitere Informationen zur Programmierung von Verben und Eigenschaften finden sich nach Eingabe von `help @verb`, bzw. im Xpress-Hilfebrowsers, sowie im MOO-Programmierhandbuch [14].

Objekte, Verben und Eigenschaften verfügen weiterhin über einen Satz von Berechtigungen, die u.a. kontrollieren, ob und durch wen Verben und Eigenschaften eines Objektes verändert werden dürfen. Die vollständige Syntax des Kommandos lautet demnach:

```
@verb <Objektname>:<Verbname> <Direktobjekt> <Präposition> <Indirektobjekt>  
<Berechtigungen> <Eigentümer>
```

▷ Weitere Informationen zu Berechtigungen und Eigentümern finden sich nach Eingabe von `help @chmod`, bzw. im Xpress-Hilfebrowsers, sowie im MOO-Programmierhandbuch [14].

4.3 Programmieren in Xpress

Xpress bietet einen eingebauten Editor (s. 2.3), der die Programmierung vereinfacht. Mit diesem lassen sich Attribute, Eigenschaften und Verben eines Objektes leicht einsehen und verändern. Dazu tippt man den Namen oder die Objektzahl des zu editierenden Objekts in das Eingabefeld "Object" ein. Um direkt den Programmcode für ein Verb eines bestimmten Objekts zu editieren, gibt man das Verb nach dem Objektname (bzw. der -nummer) ein und trennt beides durch einen Doppelpunkt (`<Objektname>:<Verbname>`). Um Eigenschaften zu bearbeiten, benutzt man anstelle des Doppelpunktes einen einfachen Punkt (`<Objektname>.<Eigenschaft>`). Ein Klick auf den Button "View" zeigt das Objekt mit allen seinen Verben und Eigenschaften an. Um ein neues Verb zu programmieren, klickt man auf "New Verb", woraufhin sich ein weiteres Fenster öffnet, in dem man Code eingeben, und diesen anschließend mit Klick auf "Compile" kompilieren kann. Sofern der Code keine Fehler enthielt, ist das neue Verb dann verfügbar. Analog ist das Vorgehen zum Hinzufügen einer neuen Eigenschaft. Ein Verb oder eine Eigenschaft

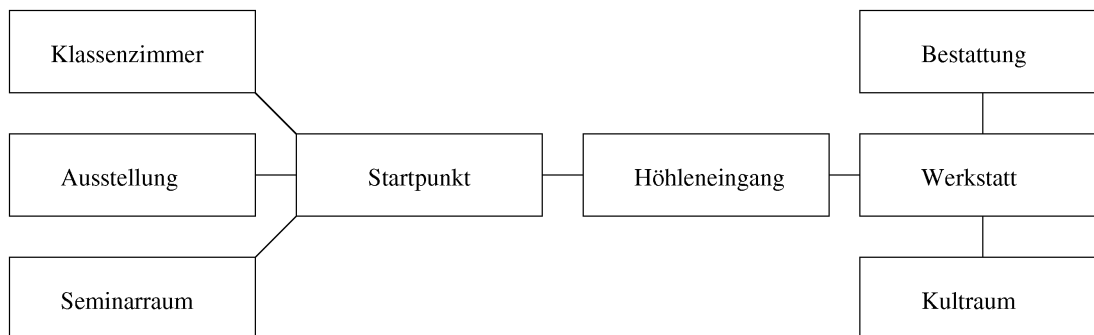
kann gelöscht werden, indem man es in der Liste auswählt und auf “Delete” klickt.

▷ Eine ausführlichere Einführung zum Thema “Programmieren in Xpress” erhält man nach Eingabe von `help Xpress_Program_Editor` bzw. unter dem entsprechenden Eintrag im Xpress-Help-Browser. Das MOO-Programmierhandbuch von Pavel Curtis ist ebenfalls vom Xpress-Programm-Editor aus erreichbar.

5 MaMOOth – Steinzeit erfahren im MOO

Der folgende Abschnitt beschreibt die Anwendung der im vorigen Abschnitt beschriebenen Konzepte auf ein kleines Beispielprojekt, das auf enCore aufbaut. Erstellt werden soll ein MOO, das dem Besucher einige Ausschnitte aus dem Leben altsteinzeitlicher Menschen nahebringen soll. Die zu implementierende Lernumgebung ist sehr einfach strukturiert. Man betritt das MOO über den Startpunkt. Es gliedert sich funktional in zwei Teile, die, vom Startpunkt aus gesehen, dessen linke und rechte Hälfte ausmachen. Links befinden sich die Seminar- und Ausstellungsräume, rechts die Räume der Höhle, die thematisch angelegt sind. Vom Eingang der Höhle gelangt man in die “Werkstatt”, in der sich Objekte mit Bezug zu altsteinzeitlichem Handwerk und Überlebenskunst befinden. Von hier zweigen zwei weitere Räume ab, die jeweils einen Einblick in Bestattungs- und Kultsitten bieten.

Die vom Startpunkt aus gesehen links liegenden Räume können für Unterrichtsveranstaltungen genutzt werden. Klassenzimmer und Seminarraum funktionieren wie in 4.1.2 besprochen. Der mit “Ausstellung” bezeichnete Raum dient dazu, interessante Gegenstände, Texte und sonstige Medien aufzunehmen, die einen Bezug zum Thema haben, sich aber nicht in die vorhandenen thematischen Räume einordnen lassen.



Einen wichtigen Teil der Lernumgebung soll die Einbeziehung von Abbildungen darstellen. Die Fundstücke sollen vollständig als Bilddateien vom Besucher zu betrachten sein. Textuelle Be-

schreibungen der Objekte in den Räumen sollten nicht zu ausführlich sein, um den Erkundungsdrang des Besuchers nicht zu schnell zu lähmen. Sowohl die Schwierigkeit der Text also auch der Detailgehalt sind dem jeweiligen Objekt angepasst. Die Raumbeschreibungen sind kurz und einfach gehalten. Da diese Texte das Erste sind, was der Besucher zu sehen bekommt, müssen sie gut formuliert sein, um von Anfang an sowohl interessant, verständlich als auch hilfreich zu sein. Wichtig ist, dass der Besucher nicht sofort mit einer Flut an Informationen überschüttet wird, das Interesse behält und an jedem Punkt weiß, welche weiteren Möglichkeiten ihm offenstehen. Das Schema sieht daher zunächst einen kurzen einleitenden Text vor, der eine interessante Atmosphäre vermitteln soll. Danach folgen didaktische Inhalte in Form einer Beschreibung der Bedeutung des Raumes und der in ihm enthaltenen Gegenstände. Am Ende folgen ein paar Hinweise, die dem Benutzer Hilfestellung zur weiteren Navigation geben.

Die Erstellung der Räume und Objekte erfolgt im Xpress-Objekteditor, der nach einem Klick auf “objects” in der Iconleiste erscheint. Für das Klassenzimmer und den Seminarraum muss jeweils ein Objekt vom Typ *Classroom* und *Moderated Room* erstellt werden, alle übrigen Räume sind vom Typ *Room*. Beim Erstellen eines Raumes bietet der Xpress-Objekteditor auch gleich die Möglichkeit, die Verbindungen zu den anderen Räumen herzustellen.

Möglichst viel Information soll mit direktem Bezug zum Fundstoff vermittelt werden. Dazu werden Funde aus der archäologischen Literatur herangezogen und so in den Räumen verteilt, dass sie sich in das jeweilige Raum-Thema fügen. Sie werden mit einem Bild und einer Textbeschreibung versehen und in den entsprechenden Räumen platziert. Dazu dient der einfache Objekttyp *Thing*. Ein neues Objekt wird, wie in 4.1 beschrieben, mit dem Xpress-Objekteditor (Klick auf “objects” in der Iconleiste) erzeugt.

Das Objekt *Note* enthält einen Text, der angezeigt wird, wenn der Benutzer darauf klickt. In MAMOOth wird es genutzt, um Texte zu Themen anzuzeigen, die nicht direkt mit einem bestimmten Artefakt verbunden werden können oder sollen, Detailfragen beleuchten oder etwas ausführlicher zu behandeln sind. Im Raum namens “Höhleneingang” liegt beispielsweise ein solches Objekt, das einen Text zum Thema “Feuer” enthält. Mit “Edit Note Contents” im Objekteditor können einem *Note*-Objekt entsprechende Texte zugewiesen werden.

Im nächsten Schritt sollen die Objekte und Räume mit Texten und Bildern versehen werden. Nach einem Klick auf “my stuff” erscheinen die neu angelegten Objekte und Räume in der Inventarliste. Über das “edit”-Icon (Stift) erhält man Zugriff auf die Eigenschaften. Um einem Objekt oder Raum ein Bild zuzuweisen, wählt man im nun erscheinenden Dialog aus der Liste “Edit Multimedia Content” und gibt die URL für die entsprechende Bilddatei ein. Die Auswahl

von "Edit Description" öffnet ein Eingabefeld, in das man den beschreibenden Text eingeben kann. Den Räumen zugewiesene Bilder und Texte werden angezeigt, sobald ein Benutzer einen Raum betritt. Bilder und Texte zu Objekten bekommt ein Benutzer zu Gesicht, sobald er auf das das Objekt repräsentierende Icon (welches ebenfalls im Objekteditor konfigurierbar ist), klickt.

Mit diesen Elementen ist es bereits möglich, ein einfaches MOO aufzubauen, in dem der Besucher seinem Erkundungsdrang freien Lauf lassen kann, wo andererseits aber auch ein vorbereiteter Unterricht oder ein Vorlesung online stattfinden kann. Dazu dienen die bereits in 4.1.2 besprochenen Räume vom Typ *Classroom* und *Moderated Room*, die in Verbindung mit weiteren bereits vorgestellten enCore-Objekten, wie etwa dem *Web Projector*, alle erdenkbaren Funktionen bieten, um Inhalte online zu vermitteln.

Der weiteren Ausgestaltung des MOO sind praktisch kaum Grenzen gesetzt. Beliebige weitere, im Internet verbreitete, Medien, wie Ton- oder Videosequenzen, aber auch interaktive Java- oder Java-Script Anwendungen sind mit dem Raumtyp *Web Page* integrierbar, URLs können mit dem Kommando @URL jederzeit auf den Bildschirmen der in einem Raum befindlichen Personen angezeigt werden. Der Objekteditor von Xpress nimmt unter dem Feld "Edit Multimedia Content" ebenfalls nicht nur Bilder, sondern beliebige, über eine URL erreichbare, Medien entgegen. Darüber hinaus eröffnet die Programmierung eigener Objekte und Verben weitere Möglichkeiten, das MOO dynamisch zu gestalten.

Was die unterrichtsbezogene Vermittlung von Inhalten betrifft, steht mit VASE (Virtual Assignment System Encore) sogar ein System bereit, welches das Erstellen von Multiple-Choice Tests erlaubt. Diese Funktion, die hier nicht weiter besprochen werden soll, ist erreichbar über den Menüpunkt "xpress" in der Iconleiste des Xpress-Clients.

6 Aspekte der Wissensvermittlung im MOO

Das Konzept der MOO-Technik ist besonders stark auf individuelles Lernen ausgerichtet. Benutzer betreten und verlassen das MOO nach Belieben. Es steht ihnen frei, sich innerhalb des virtuellen Raumes zu bewegen und mit anderen Benutzern zu interagieren. Diese Bewegungsfreiheit widerspricht natürlich vollkommen den etablierten Lehrstrukturen an konventionellen Bildungseinrichtungen wie Schulen und Universitäten. Außerdem stellt sich unwillkürlich die Frage, inwiefern es sinnvoll ist, mit den Mitteln eines MOO Klassenzimmer, Unterrichtskurse oder Museumsräume nachzubilden, wenn die Wissensvermittlung ebenso gut in realen öffentlichen Räumen stattfinden kann.

Das Potential eines MOO erkennt man nicht, solange man es nur als eine mehr oder weniger gelungene Simulation traditioneller Unterrichtsmethoden begreift. Tatsächlich profitiert ein solches System als Kind des Internet direkt von all den Möglichkeiten, die das Internet selbst zum wichtigsten Wissensnetzwerk unserer Zeit gemacht haben. Es bildet eine weltweit erreichbare Kommunikationsplattform, die es geographisch von einander getrennten Personen ermöglicht, jederzeit miteinander zu kommunizieren, Ideen, Medien und Meinungen auszutauschen und bietet zu diesem Zweck weit ausgereifere und flexiblere Funktionen, als beispielsweise Email-, Chat- oder Videokonferenzsysteme. Dabei bieten LambdaMOO und vor allem das hier vorgestellte enCore alle technischen Voraussetzungen, um bereits mit einem Internetzugang, der nur über eine geringe Bandbreite verfügt, vollen Zugang zu diesem ausgereiften System zu erhalten. Um wieviel einprägsamer Informationen sind, wenn sie auf selbstgewählte Zusammenhänge hin untersucht werden können, ist bekannt. Die Möglichkeit, sich selbstständig mit Dingen, seien sie abstrakter oder gegenständlicher Natur, beschäftigen zu können, den "Entdeckungstrieb" ausleben zu können, schafft eine persönlichere, weit dauerhaftere Bindung zum Betrachtungsobjekt, als es die Aufnahme von "fremdaufbereiteten" Informationen, etwa in Vorlesungen, jemals könnte. In diesem Bereich liegt eine besondere Stärke des MOO-Systems im Vergleich zu anderen Online-Systemen. Zur Vermittlung bestimmter, vor allem abstrakter Sachverhalte, wie etwa Programmierkenntnisse oder anderen technischen Fähigkeiten, scheint ein MOO geradezu ideal geeignet zu sein, sofern entsprechend fähige Benutzer zur Verfügung stehen, die sich bereit erklären, solche Inhalte über das MOO zu vermitteln. Inwiefern es auch die Beschäftigung mit real existierenden Gegenständen ergänzen oder sogar ersetzen kann, wird in Zukunft vor allem eine Frage der fortschreitenden Entwicklung im Bereich der über das Internet zugänglichen Medien sein [15, 16].

Das inhaltliche Konzept eines ansprechenden MOO sollte, insofern als Zielgruppe nicht nur eine enge Gruppe von speziell ausgebildeten Personen ins Auge gefasst wird, vorsehen, dass die darin präsentierten Themen so gewählt und aufbereitet werden sollten, dass sie einem breiten Publikum zugänglich werden. Eine Vielzahl von Benutzern führt bei einem offenen System wie einem MOO zu einer qualitativen Verbesserung. Durch deren Feedback können zahlreiche in der Planungsphase unbeachtete Probleme erkannt und beseitigt werden. Bei der späteren Ausgestaltung des MOO werden die Benutzer eine wesentliche Rolle spielen. Ein "lebendiges" MOO profitiert vor allem vom Ergeiz seiner Benutzer, selbst Objekte und Räume zu erstellen, eigene Ideen und eigenes Wissen anderen zur Verfügung zu stellen. Zu diesem Zweck bietet ja bereits LambdaMOO ein ausgefeiltes Rechtesystem, das es ermöglicht, ausgewählten Benutzern einen entsprechenden Status als Builder, Programmierer oder sogar Wizard einzuräumen, um sie an der

Ausgestaltung des MOO und seines Themanangebotes teilnehmen zu lassen.

7 Schlussbetrachtung

Abschließend lässt sich in Bezug auf die eingangs formulierten Fragen folgendes resümieren:

1. Die verfügbaren MOO-Systeme, d.h. in aller Regel LambdaMOO und darauf aufbauende Erweiterungen, sind ausgereift und frei im Internet erhältlich. Ihre Installation auf gängiger PC-Hardware bereitet keine Schwierigkeiten. Auch die weitere, zum Aufbau und zur Benutzung eines MOO benötigte Software ist entweder standardmäßiger Teil gängiger Betriebssysteme oder kostenlos im Internet erhältlich.
2. Benutzer mit geringen PC-Kenntnissen dürfte die Benutzung eines Telnet-Clients zunächst erfahrungsgemäß abschrecken. Allerdings sind die im MOO implementierten Konzepte zur Kommunikation eingänglich und durchaus nicht schwieriger zu erlernen, als etwa die Benutzung eines Chat-Dienstes, wie sie ja auch bei zahlreichen Benutzern mit geringen PC-Kenntnissen beliebt sind. Der Xpress-Client von enCore-MOO bringt hier nochmals eine deutliche Vereinfachung. Kenntnisse der englischen Sprache sind aber in jedem Fall unerlässlich. Zudem erfordern bestimmte Objekte und Räume (z.B. der "Classroom") ein Verständnis für spezielle MOO-Kommandos.
3. Mit der Benutzung von enCore-MOO und dem darin enthaltenen Xpress-Client bieten sich zahlreiche Möglichkeiten zur Integrationen von Medien ins MOO. Sofern der Web-Browser des Benutzers mit entsprechenden Fähigkeiten ausgestattet ist, stehen sämtliche im Internet verbreiteten Medienformate zur Verfügung, was der anspruchsvollen Ausgestaltung des MOO sicherlich ebenso zugute kommt wie der erfolgreichen Vermittlung von Lerninhalten.
4. Das besondere Potential eines MOO liegt in seiner Fähigkeit, eine flexible Kommunikationsplattform zu schaffen, die sich mit den Fähigkeiten und Interessen ihrer Benutzer praktisch beliebig weiterentwickeln kann. Im Hinblick auf seine Flexibilität und seine Offenheit ist ein MOO anderen im Internet verbreiteten Kommunikationssystem ([17]) weit überlegen.

Bis zu diesem Punkt sind die Grundlagen geschaffen worden, ein funktionsfähiges MOO zu installieren und mit Inhalten zu füllen; von hier ist der weitere Weg keineswegs vorgezeichnet.

Die Möglichkeiten, die ein MOO zur Wissensvermittlung bietet, sind im Rahmen dieser Arbeit nur stellenweise beleuchtet, keineswegs erschöpfend behandelt worden. Darum ist es angebracht, an dieser Stelle noch einige Überlegungen zu weiteren interessanten und vielleicht bisher zu oberflächlich oder gar nicht betrachteten Aspekten anzustellen.

Ein offensichtliches, für Administratoren mit Programmierkenntnissen allerdings nicht unüberwindbares, Problem ist die Tatsache, dass der Erschließung des MOO mit dem enCore-Client enge Grenzen gesetzt sind. Die grafische Benutzerführung dieses Clients reicht aus, um im MOO zu navigieren, Objekte zu betrachten und mit anderen Benutzern zu kommunizieren. Die Benutzung zahlreicher Objekte und Räume verlangt aber die Verwendung von speziellen Verben, die von den Elementen der Xpress-Benutzeroberfläche nicht abgedeckt werden. Besonders die Benutzung der Klassen- und Seminarräume erfordert ein Verständnis von dessen Funktionsweise auf Seiten des Benutzers. Neue Benutzer müssen sich hier gezielt einarbeiten, bzw. bei technischem Unverständnis, eingearbeitet werden.

Die im MOO herrschende Sprache ist Englisch. Das zeigt sich an jeder Äußerung des Systems gegenüber dem Benutzer. Die Verben zur Manipulation von Objekten, die Hilfetexte und die Bedienungselemente des Xpress-Clients sind vollständig in englischer Sprache gehalten. Es würde einen erheblichen Aufwand auf Seiten des MOO-Administrators bedeuten, dies zu ändern. In der Praxis beschränkt sich der Benutzerkreis des hier vorgestellten enCore-MOO daher auf Personen, die ein gutes Verständnis der englischen Sprache mitbringen.

Und schließlich gibt es noch einen ganz zentralen, im Rahmen dieser Arbeit recht vernachlässigten Aspekt, nämlich die im Vergleich zu traditionellen Lehrmethoden völlig neuen Möglichkeiten des MOO. Es wäre sicherlich lohnenswert, einmal ohne den Ballast von etablierten Lehr- Lernstrukturen übernommener Konzepte das Potential eines MOOs für sich genommen zu untersuchen. Insbesondere das im MUD/MOO eigentlich fundamentale Konzept, welches vorsieht, dass Benutzer eigene Objekte und Räume erstellen dürfen, um sie mit Anderen zu teilen, dürfte noch viele interessante Aspekte beinhalten. Die Auswirkungen, die das Vorhandensein einer virtuellen Identität auf das Lehr- und Lernbewusstsein der MOO-Benutzer hat, gehören ebenso in diesen Bereich, wie die Tatsache, dass es sich bei den Benutzern eines MOO durchaus um eine bunte Mischung aus Personen mit ganz unterschiedlichen Interessen, akademischem, beruflichem und kulturellem Hintergrund handeln kann.

▷ Ein hervorragender Startpunkt zur weiteren Beschäftigung mit dem Thema ist die "Educational MUDs"-Seite der Uni Genf. Hier finden sich Querverweise zu weiteren Internet-Seiten, die alle nur denkbaren Aspekte rund um Educational MUDs/MOOs abdecken und eine kategorisierte Liste von im Internet zugänglichen MUDs/MOOs [18]. Die letzte Aktualisierung scheint allerdings 1998 vorgenommen worden

zu sein. Gedruckte Literatur zum Thema findet sich im Buchangebot auf der Homepage des enCore-Projekts [4].

Anm.: Alle hier erwähnten URLs waren zum Zeitpunkt der Fertigstellung dieser Arbeit (30.09.01) erreichbar.

Literatur

- [1] GNU General Public License: <http://lingua.utdallas.edu/encore/gpl.html>
- [2] Apache Web-Server: <http://www.apache.org>
- [3] Informationen zum freien Betriebssystem Linux: <http://www.linux.org>
- [4] enCore: <http://lingua.utdallas.edu/encore/>
- [5] Linux Networking-HOWTO: <http://www.linuxdoc.org/HOWTO/Networking-Overview-HOWTO.html>
- [6] Apache-Overview-HOWTO: <http://www.linuxdoc.org/HOWTO/Apache-Overview-HOWTO.html>
- [7] Linux Software-Building-HOWTO: <http://www.linuxdoc.org/HOWTO/Software-Building-HOWTO.html>
- [8] Der Original LambdaMOO: <ftp://ftp.lambda.moo.mud.org/pub/MOO>. Der Ur-LambdaMOO-Server läuft auf <http://lambda.parc.xerox.com:8888> und ist öffentlich zugänglich.
- [9] TECFA-MOO der Uni Genf: <http://tecfa.unige.ch/moo/tecfamoo.html>
- [10] tkMOO-light: <http://www.awns.com/tkMOO-light>
- [11] Skriptsprache Tcl/Tk für Unix, Macintosh, Windows: <http://tcl.activestate.com>
- [12] MacMoose: <http://www.cc.gatech.edu/fac/Amy.Bruckman/MacMOOSE>
- [13] Verschiedene Linux-Distributionen auf CD als ISO-Images zum Download: <http://www.suse.de>, <http://www.redhat.de>, <http://www.mandrake-linux.de>

- [14] **LambdaMOO-Programmierhandbuch (englisch):** <ftp://ftp.lambda.moo.mud.org/pub/MOO/ProgrammersManual.ps>
- [15] **Multimedia-MUD/MOO Client Pueblo:** <http://www.chaco.com/pueblo/>
- [16] **VRML-MOO:** <http://www.bug.village.virginia.edu>
- [17] **Beispiele für kommerzielle virtuelle Räume:** <http://www.donnyworld.com/>; <http://www.worlds.net/>
- [18] **Educational MUDs (Uni Genf) mit kategorisierten Querverweisen zu zahlreichen Themen und einer Liste verfügbarer MOOs:** <http://tecfa.unige.ch/edu-comp/WWW-VL/eduVR-page.html>