

HOUSE OF WORDS

Designing Text and Community in MOO Environments

by John W. Maxwell

B.A. University of British Columbia, 1988

PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE
OF MASTER OF PUBLISHING
in the Faculty of Arts

© John Maxwell

Simon Fraser University

December, 1996

Permission is granted to make or distribute verbatim
copies of this document, provided this Copyright
notice is preserved on all copies.

APPROVAL

Name: John W. Maxwell
Degree: Master of Publishing
Title: *House of Words: Designing Text and
Community in MOO Environments*

Supervisory Committee:
Chair: Rowland Lorimer

Rowland Lorimer
Senior Supervisor
Director, Master of Publishing

Jane Cowan
Instructor, Master of Publishing
Program Director, Centre for Distance Education

David Porter
Industry Supervisor
Open Learning Agency

ABSTRACT

This paper describes the process of designing and implementing a Multi-user, Object-Oriented (MOO) environment in a distance education setting.

Between May and September, 1996, I worked with the *New Directions in Distance Learning* program at BC's Open Learning Agency. During this time, I created a prototype online environment for the purpose of conducting "mock trials" online in the context of a high-school Law course. This paper describes the project and some of the decisions made in the course of design and development. It also attempts to define MOO as a form of literary technology within the larger context of publishing and documentation. MOO technology is presented as exemplary of the interrelation between documents and communities in all sorts of publishing frameworks.

ACKNOWLEDGEMENTS

Many people contributed significantly to this project. I would especially like to thank Sandra Hawkins, NDDL's Law 12 instructor, who is responsible for the mock trials in the first place and who served as the content anchor for the project. Thanks to the excellent people at the Open Learning Agency: David Porter, Enid McCauley, Rob Scales, Prescott Klassen, Bobbie Rollins, and Kevin MacDonald; and also to my eternally patient 'alpha test' group: Ron Rapin, Ruth Yates, Susan Crichton, and Greg Hockley. Thanks to Colin McLellan for keeping the hardware running in the face of Murphy's law! And to Tiffany Lee at the Law Courts Education Society of BC for making all sorts of courtroom info available to me.

On the 'Net, I would like to acknowledge the truly excellent resources which are the *MOO-Cows* and *MOO-Calves* mailing lists; thanks especially to Slayer, for the LambdaCore upgrade docs and general good advice, and to Quinn for the *pose* feature object. Thanks to Steve Caron and Nick Ingolia for *MacGoesMOO*. And thanks of course to Pavel Curtis for *LambdaMOO!*

Thanks to Prescott, Ron W., Wain, Christina, Dave, Elizabeth, Martin, and Tim for general support and encouragement. A big thanks to my wife, Kelly, for everything, not least of which the last-minute editing I foisted on you.

TABLE OF CONTENTS

Introduction	1
Defining MOO	2
<i>MOOs in Education and Research</i>	5
<i>MOO as Literate VR</i>	11
<i>MOO as a Document Paradigm</i>	12
<i>Positioning MOO in the Communications Universe</i>	14
About the Project	16
<i>NDDL and Educational Technology</i>	18
<i>The Plan</i>	20
<i>The Virtual Courthouse</i>	21
<i>My Role in the Project</i>	23
<i>The Details of Implementation</i>	26
The Technical Details—How MOO Works.....	32
<i>Building in the MOO</i>	33
<i>MOO Administration</i>	35
Designing Virtual Environments	37
<i>MOO as Textual Multimedia</i>	39
<i>Forms of Content in MOO Environments</i>	40
<i>Player Interaction in the MOO</i>	43
<i>MOO Aesthetics</i>	45
<i>Interface Design Tactics</i>	49
Publishing Environment vs. Published Environment	52
<i>Text, Document, and Community</i>	56
<i>Some Final Reflections on the Process</i>	57
Notes	60
Bibliography	66

INTRODUCTION

This document is the result of four months' exploration of MOO, a technology for immersive, textual environments on the Internet. It presents a case study of the implementation of MOO technology in a specific setting—the *New Directions in Distance Learning* program at British Columbia's Open Learning Agency. This project formed the core of my Master of Publishing practicum, May through September, 1996, and continues into the 1996–97 school year.

In this document I present the details of working with this technology to create an environment for a high-school Law course. I also attempt to place MOO technology in a general publishing context by looking into the relationship between texts, documents, and communities. Because MOO is so unlike what we might think of as a traditional publishing framework, it can shed a novel light on the process of publishing and consuming texts.

I will begin by introducing the technology and some of its applications, and then proceed to a description of this particular project and some of the decisions made during design and construction. Finally, I will reflect on the nature of the technology and the insights that working with it provides into the publishing process.

DEFINING MOO

MOO, simply put, is a form of text-based *virtual reality* (VR) that users can access and interact with over the Internet.

The term “MOO” is a rarely expanded acronym for *MUD, Object-Oriented*—which serves to place this technology within a larger class of Internet-based textual VR spaces called MUDs: *Multi-User Dungeons*. The difference between MOOs and MUDs is a technical one; much that can be said of the sociological and phenomenal characteristics of MOOs is equally true of MUDs. In this section, I follow common practice and use the terms almost interchangeably. The distinction between MUD and MOO becomes important when discussing the details of implementation, and so I will switch to the more specific term in subsequent sections.

Pavel Curtis, the inventor of MOO software, offers the following:

A MUD (Multi-User Dungeon or, sometimes, Multi-User Dimension) is a network-accessible, multi-participant, user-extensible virtual reality whose user interface is entirely textual. Participants (usually called players) have the appearance of being situated in an artificially-constructed place that also contains those other players who are connected at the same time.¹

MOO is *virtual reality* technology. One may remember that VR enjoyed a brief time at the top of the media hype sheets in 1992–93 before it faded back into relative obscurity. It is important to distinguish between the expensive ‘gloves and goggles’-style VR, characterized by detailed, interactive, 3D graphics, and the sort of VR that MUDs and MOOs represent. The latter offers none of the glitzy graphics and ‘must-have’ gadget appeal, but often goes further in terms of providing workable virtual environments—much in the way that printed

novels remain a valid form despite Hollywood's multi-million dollar movie extravaganzas².

The history of MUDs and MOOs is wrapped up with the role-playing game craze of the late 1970s and early 1980s, best exemplified by TSR's *Dungeons & Dragons* (D&D). In a way, the D&D generation were VR pioneers just ahead of the technology. As anyone who owned a personal computer in the mid 1980s will remember, there were dozens of attempts to re-create the dungeon-game experience with computers. *Zork* and *Adventure* are memorable examples, in which the computer would provide simple textual descriptions of rooms and hallways, and, as the hero, you would type such commands as "pick up the sword" and "stab the goblin."

The generation that played *Dungeons & Dragons* every Friday evening, creating vast virtual worlds on paper and in their collective imaginations, went on to university to find the first real instances of networked and shared computing resources: VAXen and campus mainframes, the tools of an emerging computer illuminati. As early as 1979,³ a clever programmer had created a computer dungeon game that allowed multiple players to log in, presumably to gang up on the monsters, but probably as often to gang up on one another.

As "Mudding" gradually gained momentum as a student pastime, it remained largely focused on dungeon games. As computing resources evolved, so did the dungeons, dug deeper and deeper into subterranean spaces and involving more complex experiences: virtual economies, ecologies, and societies were developing in the virtual catacombs.

In 1989, an important development came from a MUD project at Carnegie-Mellon University. Attempting to speed up MUD software (bogged down in many cases by too many catacombs and dragons), a programmer

named Jim Aspnes created a program called *TinyMUD*—“tiny” because it was entirely contained in system RAM (as opposed to on a disk). It was faster and more responsive and aimed at a different sort of playing style. Instead of venturing into endless hallways and lairs, TinyMudders were more interested in socializing and creating “TinyScenery” as a backdrop to their chatting. The fact that the environment was kept in RAM effectively prevented it from growing too elaborate.

TinyMUD was symbolic of a change in a portion of the MUD community away from the *Dungeons & Dragons* aesthetic and toward the creation and colonization of social VR environments. Not long afterward, a research project at Xerox’ Palo Alto Research Center (PARC) was begun: Pavel Curtis’ *LambdaMOO*, an *Object-Oriented* MUD.⁴

Sharing the small-and-fast ideal with Aspnes’ *TinyMUD*, *LambdaMOO* featured an elegant object-based programming language. Players could easily create new additions to the environment and imbue them with custom functionality by writing small scripts or programs. The object-oriented model also allowed the *LambdaMOO* environment to become very elaborate while remaining relatively economical in its use of system resources.⁵

At some point in the evolution of MUDs and Mudding, it became clear to many that what was going on was not just game-playing, but the creation and population of immersive, virtual environments—*cyberspace*, to use science-fiction writer William Gibson’s term. By the time the term “virtual reality” reached the mainstream media—to describe the 3D goggles and headsets on teenage boys’ Christmas lists—true VR worlds had been in place and inhabited for almost a decade. The kinds of things going on in places like Curtis’ *LambdaMOO* looked more like the social and political struggles of a newly

founded colony than the toys that grace the "Fetish" pages of *Wired* magazine; what was happening within these worlds was more important than the technology that made them possible.

Today, the original *LambdaMOO* continues to grow.⁶ It is the virtual home of tens of thousands of people; at any given time, somewhere between 150 and 250 people will be connected to *LambdaMOO*—a limit set by technical constraints. The environment is practically unmappable by now, but the society that inhabits it is highly evolved.⁷

More significantly for our purposes, *LambdaMOO* has spawned dozens of other MOO environments on the Internet. Pavel Curtis has made the *LambdaMOO* software freely available on the Internet. In response, a wide variety of MOO-based virtual environments have appeared online. In particular, a number of educational MOOs are online today, providing real-time virtual environments in which educational interactions can take place.

MOOs in Education and Research

Probably the most visible educational MOO project is *Diversity University*,⁸ a "virtual campus," administratively connected to a number of universities in the United States. *Diversity University's* environment is patterned after a real college campus, with a variety of virtual rooms and workspaces into which educators can bring groups of students from remote locales. The general approach is to start with the familiar—an environment patterned closely after real life—and to let individual users create more elaborate structures as specific learning projects require them. For example, within *Diversity University*, one can find an "Advanced weather computer," a conversational "Dante's Inferno," and the "Cyber-Zapatista HQ."

A similar environment is found at *Virtual Online University (VOU)*.⁹

VOU's Web site describes online virtual environments this way:

...the isolation of the distance education student is removed as students and teachers become a part of a virtual community. They do not receive packets in the mail from an impersonal instructor nor do they sit and watch a television screen while a teacher lectures. Rather, they are active, participating members.¹⁰

A MOO conversation between myself and Michael Bertsch of *VOU*, here represented by his MOO persona, *Mogat*, gives some of the flavour of the live interaction:

You ask, "So tell me, how does one attend VOU?"
Mogat says, "One registers, pays money, then attends classes over the Internet."
You ask, "And where does accreditation come from?"
Mogat says, "Our accreditation is currently under scrutiny through North Central, in the Midwest of the US. That's where our non-profit corporation is headquartered."
You ask, "OK, I get it. How many people are we talking about, roughly?"
Mogat says, "We'll need between 50 and 100 faculty to run the station correctly. Plus support staff and admins, etc."
You say, "Holy Batman! This is not a small project! What sorts of disciplines?"
Mogat says, "So far we are weighted heavily toward the Humanities, but we have math and science folks working on the proper interfaces to teach their subjects."
You say, "Mmm-hm. I can see how text would be the primary medium at this point."
Mogat says, "Right. And that actually is an advantage. I teach writing, and this tickles me no end."
You ask, "OK, is someone here working on object-oriented Ivy?"
Mogat says, "Not to my knowledge. What do you mean by Ivy?"
You say, "The stuff that grows on the walls of universities."
Mogat says, "That may very well be your first task. Make it self-pruning."¹¹

While *Diversity University* and *VOU* offer campus-like contexts for online education, there are also dozens of smaller educational MOO projects on the

Internet. Many focus on a particular course or series of courses, often operating out of English and Creative Writing departments. A casual browse through an Internet subject catalogue will produce a dozen or more pointers to MOO projects.¹²

The Media Lab at the Massachusetts Institute of Technology (MIT) is a source of high-profile research on MUDs and MOOs in education. MIT's research on this technology began in the early 1990s with a project called *MicroMUSE*: "an educational Multi-User Simulation Environment (MUSE) and Virtual Community with preference toward educational content of a scientific and cultural nature."¹³

The bulk of MUD and MOO research at MIT has been done by Amy Bruckman, a Ph.D. candidate at the Media Lab. In four years of work, Bruckman has produced two important MOO projects. The first, *MediaMOO*, came online in 1993 as an interactive environment for media researchers; it is also a crucible for media research in itself. Bruckman's more recent *MOOSE Crossing* is a MOO in which children can explore writing and programming as self-expression.

The MIT projects differ from *Diversity University* and *VOU* in that they embody a strong theoretical perspective; namely, that of the *constructionist* school, which emerged from the Media Lab's work with the LOGO programming language in the 1970s.¹⁴ Amy Bruckman writes:

MOOSE Crossing is a text-based virtual world (or "MUD") designed to support the development of a "constructionist learning culture." Children from a variety of geographic and cultural backgrounds will connect across the Internet to collaboratively build a virtual world.¹⁵

Tari Lin Fanderclai, a teacher who has worked within Bruckman's *MediaMOO* environment, echoes Bruckman's *constructionist* perspective in the MOO:

...there is a great deal of incidental learning, for many students log into the MUD on their own time, building spaces for their groups to work in, learning to program objects, discussing topics from our class and other courses they're taking—and, of course, generally playing around. Quite possibly they learn more from projects and activities they invent for themselves than from any I assign; certainly they learn things I could not teach them in our four-walled classroom.¹⁶

Much of the research on MUDs and MOOs is sociological and ethnographic in its approach. Amy Bruckman's 1992 paper, *Identity Workshop*, explores the idea of MUD as

a workshop for exploring issues of social hierarchy. Is a hierarchical structure necessary for coordinating human group behavior? How do people obtain status within communities? The world of MUDs does not mirror reality; however, it brings the issues to the forefront and helps one to begin to think about them.¹⁷

In a later work, *The MediaMOO Community*, Bruckman and MIT researcher Mitch Resnick position MUDs as social spaces within larger community spheres:

In *The Great Good Place*, Ray Oldenburg eloquently argues for the importance of "third places," places which are neither work nor home... On the Internet, MUDs become third places which draw people with common interests from all around the world.¹⁸

The common thread in many MUD and MOO investigations seems to be a focus on the immediate and live interaction aspects of the technology. John Allison of the Ontario Institute for Studies in Education (OISE) writes,

With computer games and MOOs... the emphasis has been increasingly on interactivity. Students move away from passive verbs;

Homer was a poet in Ancient Greece.

to more active participation;

I am walking across a Greek landscape

to;

*"Hello Homer! (to the digital Homer)"*¹⁹

An interesting approach to the phenomenology of online experience has emerged in the work of several writers who argue that these computer-generated environments are in fact pure mental creations. Michael Heim, author of *The Metaphysics of Virtual Reality*, perhaps expresses this idea most succinctly:

Cyberspace is Platonism as a working product... The spatial objects of cyberspace proceed from the constructs of Platonic imagination not in the same sense that perfect solids or ideal numbers are Platonic constructs, but in the sense that inFORMation in cyberspace inherits the beauty of Platonic FORMS. The computer recycles ancient Platonism by injecting the ideal content of cognition with empirical specifics.²⁰

The mediating layer in such an environment is written language. MOOs offer a world of ideal, functionally defined objects manifested in narrative text. One experiences such a world by *reading* it, in very much the same way the world of a novel or short story is read and created within the reader's mind. But this literary reality is live and interactive instead of fixed on the page. The text grows and is shaped by the actions of each participant.

The goings-on and interactions in a MOO environment are expressed in several coexisting layers of text. The result is a novel and sometimes idiosyncratic way of thinking about description and communication. Eva-Lise

Carlstrom, invoking some of the gritty detail of the MOO environment, illustrates this:

Actions, as in real life, may also have communicative intent. As mentioned above, emoting “:sends Plato to hell” will not send Plato to hell. To accomplish that, the player would have to type “@move Plato to #19232” (the object number for Hell), at which point the character Plato will in fact be teleported to Hell. But this would be considered rude.²¹

A world in which the very real difference between “sending Plato to Hell” and merely appearing to send Plato to hell is a matter of *etiquette* is an intriguing place with a very flexible model of reality. Northeastern University’s Rémy Evard talks about the practical advantages inherent in the “liquid architecture of cyberspace”:²²

In a MUD, people and things exist in a place. One interacts with an object as one would in real life. This creates an interesting context for solving problems and indicating real-life situations. For example, in the systems MUD, I have an attic which is located above the cabin where we normally work. When I’m very busy in reality, I move to that attic, and leave everyone else in the cabin. In this way, I am out of the flow of conversation, but still available on the MUD if someone wishes to discuss a specific issue with me.²³

As the preceding illustrations suggest, a good deal of the research and investigation into text-based virtual reality has focused on the phenomenal aspects of MUDs and MOOs, viewing these technologies against the benchmark of live, spoken communication. Educational MOOs like *Diversity University* seem to treat their environment as an online space that happens to be textual but is not necessarily so. In these environments, technologies like Internet telephony—voice communication online—cannot be far behind.

There is a wealth of material written on issues of presentation of self,²⁴ construction of gender,²⁵ and discourse analysis²⁶ in MUD contexts; again, these analyses look at the use of text as a distinct feature of the technology but largely remain focused on a conversational model. These investigations, and the body of work they represent, only go so far toward understanding the complex web of social dynamics made possible by these technologies. To me, most do not adequately address the issue of MUDs and MOOs as literary environments.

To be fair, there are a number of educational endeavours using MOOs as environments for teaching creative writing, as I noted earlier. Also significant is Amy Bruckman's *MOOSE Crossing* project, which has a substantial focus on collaborative authoring environments, building a bridge between writing and programming as self-expression.²⁷

I would like to follow a slightly different path, investigating MUD and MOO technologies as literature, as documents, and as publishing environments. I seek to define and position these technologies in terms of the written tradition leading up to them. Instead of viewing MUD and MOO experiences as *written speech*, I see them as *live* or *dynamic writing*.

MOO as Literate VR

The title of this section may sound a little pretentious, but it is deliberately chosen to guard against the common conception of MUDs and MOOs as a kind of 'poor-man's VR'—that it is merely a stop-gap technology until the 3D

gear becomes affordable. I take issue with this idea, as do many MUDders and MOOers. Xerox' Pavel Curtis explains:

It is substantially easier for players to give themselves vivid, detailed, and interesting descriptions... in a text-based system than in a graphics-based one. In McLuhan's terminology, this is because MUDs are a 'cold' medium, while more graphically-based media are 'hot'; that is, the sensorial parsimony of plain text tends to entice users into engaging their imaginations to fill in missing details while, comparatively speaking, the richness of stimuli in fancy virtual realities has an opposite tendency, pushing users' imaginations into a more passive role. I also find it difficult to believe that a graphics-based system will be able to compete with text for average users on the metric of believable detail per unit of effort expended; this is certainly the case now and I see little reason to believe it will change in the near future.²⁸

In my initial definition of MOO as "text-based virtual reality," the adjective *text-based* is most definitely not subordinate to *virtual reality*.

I would like to offer another definition of MOO—as *live literature*. This definition places the emphasis on the *textual* qualities of the environment; the distinguishing characteristic being that it is interactive, real-time, and immersive. To approach the technology from this perspective sheds light on a whole range of possibilities for inquiry.

MOO as a Document Paradigm

In order to make the case suggested in the title of this section, I need to first clarify what I mean by *document*. The term has a general day-to-day meaning but not an entirely specific one. What do we mean when we talk about documents? Letters are documents. Technical manuals are documents—or, at least, *documentation*. Certainly anything with a lawyer's signature at the bottom is a document. In a wider sense, books and magazines are documents, too, and as

we become increasingly 'wired,' we would probably agree that e-mail messages and Web pages are documents as well.

I would like to offer the following working definition of document: *the artifactual incarnation of a text*. I am distinguishing between the terms *text* and *document*; the relationship between them is this: the text is something of an ideal form, and the document its manifestation, or artifactual reality. A text may be incarnate in several distinct document forms at different times—an idea that becomes more apparent as text becomes increasingly digital and increasingly fluid.

Traditionally—that is, before the advent of electronic media—documents commonly appeared in relatively few forms, generally distinguished by which end of the printing press you found them at. With the advent of photocopiers, fax machines, desktop publishing technology, and now the Internet as a publishing medium, the variety of potential document types is vast. In their variety, perhaps we can see the nature of documents more clearly.

With the fragmentation (or flowering) of digital document forms comes a similar change in the publishing process itself. What had traditionally to do with the management of paper going through printing presses is now a more complex set of activities. To continue with my model, I would define *publishing* as the production of documents, the process of making texts into documents, of making them into shared artifacts.

Publication—the production of *documents*—sets texts into motion. It gives them material reality (and whatever equivalent we find in cyberspace); it defines their location and circulation; and it governs their relationship to audiences and communities. The relevance of a text is dependent on the nature of

its incarnation as a specific kind of document. Publishing is the act of mediating the creation and delivery of documents.

MOO technology, as I will attempt to demonstrate here, is a publishing framework because it is all about the mediated creation and delivery of documents. The textual content of a MOO is manifested and contextualized by the position and architecture of the environment, and by the interaction of a community of users in and around it. The unique characteristics of MOO technology offer us a rich environment in which to explore the nature of documents in the context of digital media.

Positioning MOO in the Communications Universe

I defined MOO first as “text-based virtual reality,” and then as “live literature”. These two definitions emphasize different characteristics of MOO technology and the environments it creates. In order to take this discussion any further, we need to look more closely at MOOs and MOOing.

MOO is an Internet-accessible, real-time database that one can connect to and interact with using simple *telnet* (terminal-emulation) software. Once connected, the user (or *player*) reads descriptions of the virtual environment and happenings in the environment and can write (type) responses and make contributions. It is a shared environment; many people can connect at once and interact with others who are connected.

There are three central characteristics of MOO that I believe define it and set it apart from other media.

1. It is *live*, or *real-time*. MOO is a dynamic system, ever-changing, and thus qualitatively different from static publishing systems that put ink on paper or even Web pages in a browser window. MOO

blends the stable qualities of the written word with the dynamism of spoken communication.

2. It is a *persistent, immersive environment*, that is, one is projected into the virtual world—and that virtual world has a degree of permanence and predictability. In this sense it is distinct from IRC²⁹ or similar “chat” services that have a flat, one-dimensional aspect, like talking on the telephone instead of actually ‘being there’.
3. It is *text*. This characteristic must be considered in conjunction with the previous two; as a textual environment, it is again qualitatively different from television, radio, theatre, ritual, or other media that populate the live and immersive corners of the communications universe.

MOO is a system of documents. It is a complex world built of text; the textual reality exists on many levels simultaneously. In our investigation of MOO as a “document paradigm”, we confront a number of issues regarding the nature of documents, about the continuum between fixity and fluidity, and about the processes of making texts into social artifacts. As I will attempt to illustrate in the next section, MOO offers us a new perception of what the publishing process really does.

ABOUT THE PROJECT

Between May and September, 1996, I was involved—as my practicum project—in the production of a MOO environment for the *New Directions in Distance Learning* (NDDL) program at the Open Learning Agency (OLA).³⁰ NDDL delivers high-school level courses to schools and communities that would not otherwise be able to offer them; for instance, small towns in northern British Columbia. NDDL is also a research-oriented project aimed at developing an effective model for “mediated distance education.”

In the late twentieth century, education is in revolution. We are in a phase of chaotic restructuring and redefinition of what we want education to be. Some argue that this revolution is long past due; MIT educational technologist Seymour Papert, for example, suggests that a nineteenth-century school-teacher, magically transported to the late twentieth century, would not have any trouble adjusting to the environment or the kinds of things going on there. Papert asks us to compare this with a similar situation involving a nineteenth-century physician in a present-day operating room.

There seem to be a few general trends emerging in educational reform. One is increased choice, both in terms of where students go to school and what they learn when they get there. Another is what is sometimes called the “mix and match” approach, in which students do not necessarily take all of their courses from a single school but may pick up a few courses through distance education or other alternative channels.

One of the most interesting directions of change has to do with the traditional student-teacher model. Traditionally, a teacher, armed with an official curriculum, would face a classroom in the autumn of each year and begin a

process of delivery wherein the teacher served as the main interface between students and the material they were to learn. The teacher would be responsible for course development, expertise in the subject matter, interpretation of the curriculum, live performance, counselling, assessment, crowd control, and administration.

We are beginning to see this model give way to a more specialized, modular approach. The Open Learning Agency's NDDL program is an excellent example. In NDDL, the relationship between student and course material is actively mediated by a number of different people. At the core level, the roles of expertise and delivery are divided between two people: the *mentor*, who is a central content expert for the course, and the *facilitator*, who is the student's local contact, providing interpretation and support (both educational and technical). This is called the *triad model*; the student, mentor, and facilitator get to know one another and work together to handle the course material, which has been developed by yet another team of specialists: course authors, instructional designers, packager/publishers, and administrators. Much of the success of NDDL is attributed to the effective team-playing of local and remote participants; indeed, NDDL serves as not only a successful example of a distance learning system, but perhaps of mediated course delivery in general.

The strength of this approach, it seems, is due to the multiple layers of interpretation that lie between the student and the content of the course. In the context of the framework I set up earlier—in which I defined publishing as the production of documents and their introduction into specific communities—we can look at this kind of mediated delivery as an example of a highly interactive publishing model. Content is authored as texts (and other media) and then is published, or set into motion, by an integrated team of facilitator/interpreters.

As the end user, the student has access not merely to a teacher and a textbook but to multiple layers of the delivery and interpretation process.

NDDL and Educational Technology

NDDL uses a variety of technologies to enrich the triad model and to provide within it as many opportunities for communication and interpretation as possible. The core of NDDL's delivery strategy is an Internet-connected FirstClass³¹ bulletin board and conferencing system. FirstClass is an integrated system offering e-mail, conferences, file archives, and live chat. Much of the administration and delivery of NDDL courses happens within the NDDL FirstClass system; it is the glue that holds together administrators, instructional designers, teachers, facilitators, and the students themselves.

NDDL also employs a variety of other communications tools, including audio and video conferencing. Part of NDDL's mandate is to evaluate new tools for mediated delivery. The *NDDL 1994-95: Phase 2 Review* tells us:

The NDDL project was designed to:

- increase access to educational programs in small or remote communities
- provide alternative educational delivery opportunities
- build upon existing courseware and delivery systems
- utilize information technology effectively
- link with complementary initiatives and projects...³²

In 1995-96, NDDL began investigating a variety of Internet-based course delivery technologies, including the development of WorldWide Web-based courseware. In 1996, the Internet has nearly approached the level of a common carrier, and the opportunities for creating integrated publishing and delivery technologies using the Internet as a base are now a reality. For NDDL, it is a

very attractive medium, because it offers a very workable mix of defined standards and flexibility of application.

The *WorldWide Web* (WWW, Web) is without a doubt the key technology of the Internet revolution. Businesses and institutions are currently in a mad rush to establish their presence on the Web, and it is hard to find a school anywhere that hasn't acknowledged the Web's strengths both as a library and as a tool for student expression. The only thing bigger than the Web in 1996 is the hype about the Web.

There is, however, more to the Internet than the Web. The Internet was a remarkable tool long before the Web emerged in 1990. In fact, in the rush to populate the Internet with Web pages, many intriguing Internet possibilities have been pushed aside. One such possibility is that of live interaction—which has been almost entirely absent from the half-decade of Web development.

I first saw the Web in late 1992, before browsers like NCSA Mosaic and Netscape Navigator had been created. The Web was, to me, a set of numbered hypertext links on a terminal screen. At around the same time, I first logged in to *LambdaMOO*, which I approached from that same terminal screen. While the early, terminal-based Web was certainly intriguing, it paled in comparison to what I was seeing in *LambdaMOO's* "living room."

Since then, of course, things have changed. Marc Andreessen (at the University of Illinois' NCSA) made the Web into a graphics-friendly medium, and it exploded at an unbelievable rate. Meanwhile, MUD and MOO development proceeded at a steady pace but remained a relatively obscure technology compared to the Web.

The Plan

Early in 1996 I approached David Porter at the Open Learning Agency about a practicum project. I had wanted to do something with MOO technology, and so I suggested that the OLA develop a MOO as an adjunct to their other Internet-based delivery systems. My initial idea was to follow in the footsteps of a number of US universities and use the environment for English and Creative Writing classes.

David Porter, along with his colleagues Enid McCauley and Rob Scales, immediately had a different idea. They saw a MOO environment as fitting into the NDDL Law 12 offering. The Law 12 course is taught by Sandra Hawkins, an energetic and creative teacher well-known in BC legal-education circles as the co-author of a series of "mock trials" used throughout the province in teaching the courtroom process.³³ Before long, we had developed the idea of a MOO environment as a "virtual courthouse" in which to conduct mock trial exercises for NDDL's distance learners.

In a traditional Law course, a mock trial is generally run by assigning various courtroom roles (crown counsel, defence counsel, witnesses, jury) to students, and having them develop their roles from a set of prepared notes on the particular case to be tried. The class then ideally takes over a real courtroom in order to conduct the trial. If the students can don the robes and regalia of the court, so much the better.

Obviously, if students are distributed all over the province and connected via computers and modems, it is not feasible to assemble for such an exercise. The idea behind the MOO project is to enable this sort of event and process to take place online. It would allow Sandra Hawkins' NDDL students to participate in the mock trials she has prepared, in a manner similar to students who

all live in the same town. Previously, NDDL students had little experience with mock trials. Ms Hawkins had experimented with conducting the proceedings via e-mail and conferences, and to some extent, in audioconferences, but not with the degree of engagement we hoped to achieve with this technology.

In a sense, MOO is a natural choice: a technology that has grown out of role-playing games being used for a role-playing exercise. However, there are a host of challenges that present themselves along the way to making this concept a reality. Let us look at the project a little more closely.

The Virtual Courthouse

There are roughly twenty students in the NDDL Law 12 course. The course begins in September and runs through June. Sandra Hawkins suggests that the students will be ready to participate in mock trials by November, and so their formal involvement with the MOO would begin at that time. However, they should probably be introduced to the MOO beforehand, perhaps in comparison (or as an adjunct) to the FirstClass conferencing system. It would be a definite advantage for the students to be comfortable in the MOO environment before participating in the formal mock trial exercise.

In any case, the students' participation will ideally go something like this:

1. The mock trial exercise will be announced on the NDDL FirstClass system (which is the primary mode of communication for the class). A set of generic documentation about the trial will be made available to the students at this time, along with a set of specific documentation on working in the MOO environment.
2. At a pre-set time, an audioconference will be held with the students, their local facilitators, Sandra Hawkins, and myself. We will go

through the plan for the exercise with everyone and attempt to identify and solve any problems with connections or interface. Ideally, students will be logged in to the MOO during this audioconference, in order to try out the MOO environment while they are in telephone contact.

3. The students will be given a period of time (a week or more) to explore the MOO environment on their own and familiarize themselves with the modes of interaction in it. During this time, more detailed trial preparations will take place, such as deciding upon roles, selecting the jury, and pre-examining of witnesses. These activities may or may not take place within the MOO.
4. Exhibits and evidence for the trial will be made available to appropriate parties in the FirstClass environment and, where possible, in the MOO as well (for example, text documents relating to the trial).
5. Students will receive a formal *subpoena* by e-mail, asking them to appear in court, prepared for the trial, at a specific time and date.
6. At the appointed time, the students will assemble in the MOO, and the mock trial will take place. Sandra Hawkins will play the role of the Judge. I will facilitate the proceedings in the guise of court clerk.
7. At a pre-set time afterward, the group will assemble in the MOO again for an informal debriefing session.

The aims and objectives of the project, from the NDDL perspective, are as follows:

- to facilitate live class interaction (in general) and to conduct mock trials within the Law 12 curriculum (specifically)
- to develop a framework that actively engages students and aids in the overall integration of curriculum and technologies

- to evaluate MOO as a communications tool alongside proven technologies like FirstClass and audioconferencing
- to explore the possibilities raised by new media and novel approaches to mediated distance learning

My interest, as I indicated earlier, is to explore the nature and possibilities of MOO technology as a document framework; to get away from viewing MOO as 'cheap VR' and instead focus on its textual richness. The subject matter (of Law 12) is highly appropriate to this angle, as our legal system is very much a system of documentation punctuated by live, oral courtroom exchanges. The depth and variety of the MOO document framework is in some ways mirrored by the depth and variety of the legal document framework.

My Role in the Project

My role in the project is as its architect and facilitator. I am charged with the design, construction, and management of the MOO environment. The project's 'content expert' is Sandra Hawkins, the course instructor and author of the mock trial material used in this project.³⁴ Ms Hawkins provides valuable information about the design of the environment and its details, as well as acting as a liaison with the students as the course instructor. Ms Hawkins is, in a sense, my most important client—unless she is able to work with the environment, there is little chance that her students will.

My job is to manage the production and distribution of *documents* in this context. Managing refers both to the design and construction of the virtual environment and to the proliferation of texts within that environment. In that I am playing the central editorial and interpretive role in the process, I am the *publisher* of this project; I am responsible for the interpretation and presenta-

tion of the supplied content—the prepared texts and notes for the mock trial—as well as the meta-information regarding the courtroom and the ambient context for the exercise.

The interpretation and editing process requires that I situate the texts within two overlapping contexts: first, that of the Canadian legal system, so that the materials are consistent with a real courtroom experience, and second, that of the MOO itself. The prepared trial texts need to be integrated with the environment in such a way that they are accessible and understandable and that the interface is as transparent as possible.

Beyond that, however, is yet another layer. The MOO environment must encourage participant engagement and contribution. The richness of the MOO is not only in the texts that we place in it but significantly in the dynamic interaction of connected users. This *live* layer needs to work with the prepared materials in such a way that my “virtual courthouse” has an existence greater than simply a backdrop for the mock trial exercise.

There are, at root, three groups of people involved in this endeavour: myself as technical facilitator and designer; the students as end-users; and the NDDL staff and faculty as content experts, authors, and organizers of the larger context of the project. My job is not so much producing content as serving the content producers.

My challenge, then, is to open and facilitate a particular kind of channel:

1. To build a situated environment—the virtual courtroom—that does justice to the kind of situated learning that is being attempted: that is, a simulated trial in a simulated court.

2. To create a mediated environment that is easy for users (both students and instructors) to enter and use effectively with a minimum of ‘fussing’ with the details of connections, interface, etc.
3. To develop a stable, usable platform that can be evaluated on a even footing alongside other distance learning tools.
4. To make it possible for other developers to build directly onto the environment. That is, to encourage others (students, staff) to participate in the process of building an environment that suits the needs of legal education.
5. To create a usable overall framework for further development of MOO-based situated learning.

The creation of such channels—through which texts become documents at work within a community of learners and teachers—is what educational publishing is really about. Literary critic Stanley Fish talks about the “interpretive communities” that serve to construct the meanings of texts for their members.³⁵ These communities form the frames and contexts for our understanding. Educational publishing in the sense that I am trying to establish here is about dialogue with and within interpretive communities—dialogue that extends to the conscious creation and publishing of documents.

The trend toward increasing specialization of educational delivery roles carries with it the need for an elaborate system of document publishing and interpretation, one with opportunities for feedback and interaction along the way. As the distance between student and course material is bridged by more specialized and discretely identified *layers* of facilitation, the importance of communication and publishing between and across these layers increases.

What sorts of documents will serve to populate and define these layers?
Hopefully, *interactive* ones will.

The Details of Implementation

I began in May with the installation of a MOO server on a computer at the Open Learning Agency. The task before me was to design and create an online environment for mock trial exercises. I felt, however, that this environment needed to have a larger scope than just bench, dock, and witness stand; it needed to be an inclusive environment that would encourage exploration and elaboration by the participants.

I started building *NooDDLe*³⁶ as an outdoor environment, described as recently cut out of the British Columbia wilderness. Within this virtual clearing I created a number of virtual buildings, a workshop, a gallery and foyer, and then the courthouse itself. The courthouse contained a number of rooms, including courtroom, library, and a number of offices and semi-private spaces. I wanted to create a coherent world with both an inside and an outside, in order to give the place a setting. The richer the environment, the more there is for the participants to think about. So, we not only had a courtroom, but a courtroom within a cedar building set in a garden in the coastal rainforest.

I spent some time adding descriptions to the environment, both indoors and out, to make it as complete as possible. I extended the environment into my virtual forest and created a few spaces just for fun, thinking that students who take the time to explore should be rewarded by finding interesting things just beneath the surface—something like real life.

One of the first things I built indoors was the *Library*, a space to keep specific documents within the MOO. I had to move the *Library* entrance a few

times, before finally merging the gallery and foyer with the courthouse itself, so that the library, courtroom, offices, and garden all led to a central common area called the *Courthouse Foyer*. This central area includes a *Guest Book* that visitors can sign for posterity.

The *Courtroom* itself was challenging to design. How does one describe a courtroom in any sort of meaningful way? To describe it visually is not very fruitful. I had to come up with a way to describe the courtroom functionally. What I eventually created was a 'smart room' that is aware of which trial roles are associated with specific players and that produces custom descriptions and outputs to those players. So, whether you are a member of the court proper or merely a spectator defines which doorway you enter through, which in turn produces a slightly different description of the room. Further, the room 'responds' to certain players; whenever Sandra Hawkins enters the *Courtroom*, it recognizes her as the Judge, and an automated *sheriff* says, "Order in Court!"

The *sheriff* is only one such automated "bot"³⁷ in the MOO. The *court recorder* is an also automated routine; it logs or records everything that occurs within the *Courtroom* and will play back portions of this log on request.

Ms Hawkins was interested in being able to use the environment as a demonstration tool on occasion, and so she asked me to create an automated 'tour guide' that would lead guests through the various areas and talk about what was there. I created an overly polite *Host* bot that normally inhabits the *Garden* (which is where players find themselves when they first log in). The *Host* will respond to a request for a tour by leading a player into the courthouse and through some of the rooms. We felt that the *Host* would be a useful service for players trying out the technology for the first time.

With the basic environment constructed, I next had to turn my attention to the presentation of the mock trial materials themselves.

The environment as I have described it so far is purely a shell. There is some built-in functionality, but just as a bare courtroom does not make a legal system, the empty MOO environment would not offer much richness to the mock trial exercise. Trials and law are all about documents, and there needs to be at least a basic set of documentation for the trial to make sense. Exhibit materials need to be presented in the court, and relevant legal documents (sections of the *Criminal Code of Canada*, for instance) must be available. Further, a trial in progress is an ongoing work of documentation; in addition to the court record, there needs to be a system for notes and note-taking by the trial participants.

Creating notes and documentation within the MOO is a relatively simple matter; there is a built-in function for creating 'writable' objects. However, a courthouse environment requires an extensive set of internal documents, and so—as I will describe in some detail later—I paid attention to the issue of creating in-MOO documents.

We must remember, though, that the participants of our mock trial—Law 12 students sitting in front of computers—do not *live* in the MOO; they have access to all sorts of documents and notes and tools outside the environment. They undoubtedly have pens and paper, word processors, and electronic note takers. And they will also have the FirstClass e-mail and conferencing system. Since a MOO client takes up relatively little memory and space on a computer monitor, it is reasonable to expect that other programs will be open and on-screen alongside the MOO. So, practically speaking, the documentation component of the trial will exist both and inside and outside the MOO environment.

What seems clear is that those documents that relate to the immediate details of the MOO experience need to be immediately available within the environment. Exhibits, court records, and anything that needs to be referred to directly and/or shared within the trial should be present as documents in MOOspace. These documents may also exist outside the environment. For instance, it makes sense to place a set of the exhibit materials on the FirstClass system, which would make it possible for participants to view them in a separate window if they choose. Any pictorial exhibits would be served by a textual description within the virtual courtroom and a graphic file within the FirstClass system, allowing easy access to either version.

There are a number of challenges inherent in conducting so text-heavy an event as a trial within a textual environment like a MOO. Here are some of the issues that have guided the development of the environment so far:

- The speed of the trial proceedings will owe much to the typing speed of the participants. In order to keep the process moving, the stock sections of the trial (i.e., those portions of the proceedings that are generic or written in advance) can be automated or delivered by macros. The court clerk's contributions, for instance, can largely be automated, as can the Judge's introductory remarks and charge to the Jury.
- The success of the exercise will probably owe much to the atmosphere of the court, inasmuch as the virtual courtroom is able to convey the stateliness and formality of a real court. If the ambient setting for the mock trial is detailed and accurate, the result should be a richer experience for participants. Since MOO is a textual environment, we have all the possibilities of written literature to help in

the creation of this atmosphere. However, the danger of overwhelming the participant with text is very real.

- It is worth noting that what we will be attempting in running a mock trial in a MOO environment is a *two-tiered* role-playing scenario. On one level, we are asking students to enter into an artificial world wherein they are the participants in a trial in a court of Law. On the other level, we are asking students to enter into the artificial world of the MOO as virtual entities in a virtual world. In fact, we are doing both at the same time, with very little time for overlap. We will, in a sense, be asking the students to find their feet in an artificial world two layers deep. While this particular aspect is not the focus of my research, I am aware of the complexity of the situation, and it will be interesting to see how students navigate this “frame within a frame”.³⁸

The stated objectives of this project—to facilitate live class interaction and to actively engage the students in the tasks at hand—will be the measures for the success of the project. It seems likely that success will rely on striking a balance between including sufficient detail to make the mock trial as ‘real’ as possible and ensuring that interaction in the environment is smooth, understandable, and responsive.

Writers on the subject of interface design commonly point to the ultimate goal of making the interface disappear completely into the functionality of the piece. In this exercise, the interface design issue is especially critical, as the user must successfully navigate several layers of textual material in order to understand what is happening. The challenge is most apparent at three stages:

1. The virtual courthouse must behave in predictable and coherent ways that provide a helpful context for the trial exercise. This challenge applies not only to the architecture of the environment but also to the documents, props, and enhancements that exist to facilitate the trial.
2. The language used must be both simple enough and rich enough to produce an engaging experience. The environment must be presented in ways that allow the user to create the experience in her own mind, as a good novel or a short story would. At the same time, if this textual presentation is too long or too complicated, it will stifle the 'live' aspects of experience. The same constraint applies to what the participants are required to type themselves.
3. The technical requirements must not present any barriers to involvement. We will be distributing a very simple MOO client application (essentially a glorified *telnet* program) to the students, so that logging in to the MOO should be as easy as double-clicking on an icon. However, once logged in, the matter of "finding one's feet" in the MOO will also take a certain amount of time and effort.

THE TECHNICAL DETAILS: How MOO Works

MOO software consists of a server and a database, both of which are freely available from Xerox PARC.³⁹ The server is a Unix program, available from Xerox as source code and in a number of pre-compiled versions for popular Unix systems. The server's role is to manage connections and processes and to load and parse the database, which is in the form of a large text file. The server has been ported to a number of non-Unix operating systems;⁴⁰ the database file itself is in a generic format, and so it works with any platform.

MOO runs entirely in the system RAM of the host computer. While this makes the environment quick and responsive, it also places a considerable load on the host system. While MOO does not place many demands on the system's processor, its RAM requirements are roughly twice the size of the database loaded. The basic "LambdaCore" database is nearly two megabytes as shipped, and as MOOs evolve, they grow. Xerox' original *LambdaMOO* is over *one hundred* megabytes; the server requires nearly twice that amount of RAM to remain responsive.

At pre-set intervals (commonly once or twice per day) the server dumps (*checkpoints*) the entire database to the hard disk for safekeeping. If the host computer were to crash at some point, the environment could easily be recovered up to the last checkpoint. Normally, however, the MOO resides in RAM continually, saving itself periodically. The checkpointing procedure is quick and, for the most part, transparent to users.

The LambdaCore database is the basic virtual environment. As delivered from Xerox, it is a complete environment, composed of exactly one "room".

When the MOO is started for the first time, it loads the core database and allows the administrator (the *Wizard*) to log in. At this point, nothing exists beyond the first room, but there is within the core a set of primitives or generic objects with which one can build an entire world of rooms, things, notes, doorways, and other players. There is also a programming language, commonly called *MOO code*, with which the virtual reality can be configured and brought to life. As well, within the core is a detailed *help system*, which can be accessed by any user simply by typing "help" within the environment.

Connections to the MOO are made via telnet protocol to a specific port on the server, much as one would connect to an online library catalogue. The interaction is then simply a matter of text being written to the user's screen and the user typing responses back. While the connection is based on telnet, most users will employ a more specialized client application⁴¹ that splits the input from the output onscreen, and that may offer some automation or macros.

*NooDDL*e runs on a Sun Sparcstation 5 (star.ola.bc.ca port 8888) in the Information Services (IS) department at OLA. The machine is administered by IS staff, but I take care of the administration of the MOO itself. The system has been running continually since May, with one or two planned shutdowns during that time.⁴² As of this writing, the database is roughly 3.5 megabytes. It checkpoints twice a day.

Building in the MOO

Creating and populating a MOO is a matter of creating and manipulating *objects*. In accordance with object-oriented programming principles, everything in the environment is structured as some kind of *object*. Objects can be said to be 'parents' or 'children' of other objects, and there is a hierarchical system of

inheritance among these: objects share properties and functions with their parent object. In the LambdaCore database, there are about one hundred objects, most of which are generic and which can be used as templates for creating new objects.

In order to allow people to connect and log in to our MOO environment, we need to create a number of *player objects*. When connected and logged in to a MOO, each individual is represented within the environment by a specific player object—an avatar or MOO persona. Every *player* is a descendant of the generic *\$player*⁴³ object (part of the core database), and therefore shares all the functionality of the parent object.

A more complex example is the construction of a 'virtual courtroom', a child of the generic *\$room* object, but with a set of added functions and properties related to the running of trials: prisoner's dock, witness stand, and controls on who can speak at what times. These details are described in MOO code according to their function, so that interaction within the virtual courtroom can take place by reading and typing appropriate statements. Finally, the completed virtual courtroom can then be seen as a *generic courtroom*, from which subsequent courtrooms can be created, without having to rebuild each time.

The creation of new *room* objects is the most straightforward way of building a MOO environment—for the most part, one can simply "dig" new rooms and give descriptions to them. It is not necessary to add specialized functions as I did with the *Courtroom* unless there is a specific purpose.

In addition to *rooms* and *players*, we can create *things*, which are simple objects without any built-in functionality, *notes* to write on, *containers* to place other objects in, and so on. From there, we can begin to specialize objects by adding new MOO code to descendants of these basic building blocks. Just as I

created the *Courtroom* from the generic *Sroom*, it is important to understand that the generic *Sroom*, *Sexit*, and *Splayer* were all created from the generic *Sthing*, which in turn is a specialized instance of the *root class*, or bottom level of the database. At each step in the hierarchy, more functionality is added. Xerox' *LambdaMOO* contains tens (if not hundreds) of thousands of objects.

MOO Administration

The administration of such an environment grows similarly complex. In *LambdaMOO's* case it has begun to look quite a bit like civic politics. The most interesting aspect of MOO administration is that it is handled entirely internally—that is, completely within the bounds of the virtual environment. There is, in fact, no access to any aspect of the virtual environment from outside the MOO. It is an entirely self-contained world. Even the documentation for using and building the MOO is contained within the environment.

This is not necessarily a limitation. In fact, it reinforces a long-standing principle of MUD and MOO administration: that one should always try to manage things from within the system. In a classic article on multi-user environments, *The Lessons of Habitat*, Chip Morningstar and F. Randall Farmer warn that in building a virtual community, structures need to evolve naturally from within, and that,

Wherever possible, things that can be done within the framework of the experiential level should be. The result will be smoother operation and greater harmony among the user community. This admonition applies to both the technical and the sociological aspect of the system.⁴⁴

MOO administration is—at least at the technical level—a matter of governing object ownership. Each MOO user is represented in the environment by his

player object, and there is a simple class system built into LambdaCore that manages a set of permissions and rights. At the top of this system is the *Wizard*, roughly analogous to the system administrator. The Wizard's powers are unlimited within the environment.

The rest of the population are members of either the simple *player class*, the *builder class*, or the *programmer class*; these distinctions refer to a player's ability to create, modify, and destroy objects within the environment. They set the ground rules for a collaboratively authored world.

In the case of *NooDDLe*, I occupy the *Wizard* role, but only when it is necessary. The rest of the time, I exist as a *programmer-class* player. There are a handful of other *programmer-class* players in *NooDDLe*—OLA staff who are interested in working with the environment—but most players are at the *builder* level. As builders, they are able to create new objects (notes and documents, for instance) and even to “dig” new rooms if they desire, but they do not have the power to program or modify the functionality of objects.

MOO is at once a very simple and a very complex system. To the extent that it is both at once we might say that it is an elegant technology; where it fails to be both and instead is either too simple or too complex, it is not an elegant technology. The challenge of making MOO really work is to create a virtual world and a virtual community where the elegance can shine through, and the ugly aspects fall away.

DESIGNING VIRTUAL ENVIRONMENTS

MOO is, at heart, an Internet-accessible information service or database that one can use to retrieve information from, interact with, or add to at will.

Technically, it is not much different from many other types of Internet information services: one establishes a connection using a *client* program, logs in (identifies oneself), and navigates through the information by issuing specific commands in an interactive session. When finished, the user logs out and breaks the connection.

A telnet session to a library catalogue or other terminal-access database is very similar in this respect. What is different about MOO is that one is aware of other users and can interact with them as part of the data. In the library catalogue example, the fact of the user's presence is in some way 'known' by the system, but that information is kept separate from the real content of the database. In a MOO, we might say that the user's presence becomes an integral part of the data. Hence, MOO is immersive in ways regular databases are not.

Similarly, we might compare MOO with the WorldWide Web, another client-server system in which one can connect to various information spaces and interact with them. But the difference here is even greater, as the Web is a *stateless* system; that is, a connection or query to a Web site is a discrete transaction: query and response. The information base is unaffected (aside from a line written to the server's log file) by the user's request; the existence of the user has no representation within the data. Furthermore, there is no temporal element to Web interaction; information pulled from a Web server is a snapshot of that data at the time of the request.

MOOs can also be compared to a different set of network services: e-mail and conferencing systems. Many of the things people do in MOO environments can be done via e-mail correspondences or newsgroup discussions. In these cases, the user's presence is important in the system, and the process is definitely interactive, but it is not 'live'—not real-time.

Much has been written on the wonders of asynchronous communication; on the convenience of dealing with one's e-mail at any time of the night or day; on the curious compression of time and space that occurs in electronic correspondence. This is fundamentally different from what is happening in a MOO, where the experience is either solitary (when there is only one user logged in) or a live, social happening.

A fourth MOO comparison would be with real-time "chat" services, such as *Internet Relay Chat* (IRC), bulletin board "chat rooms," or FirstClass live chat sessions. These services rely on the system 'knowing' who is logged in currently and creating temporary interactive links between users. But chat sessions are certainly not immersive. All that is possible is conversation. Like talking on the telephone, the interaction is limited to dialogue and cannot refer to other items in the system.

MOO, as distinct from a chat session, is a database of information objects, a database that includes users within it but is not limited to them. It is a persistent virtual environment, more or less the same place each time one logs in. A MOO environment can contain many of the things mentioned above: indexes and catalogues, Web pages (there are several MOOs on the Internet operating as sometime Web servers), e-mail and discussion groups,⁴⁵ and, of course, live interactivity and dialogue.

MOO as Textual Multimedia

With so many layers of experience and interaction within a MOO environment—the architecture, the contents of the environment, the live discourse, and the variety of discrete document forms—one can almost call MOO a multimedia technology. True, there are no graphics, video, or stereo sound, but to call the technology “merely text” misses the point about immersive environments and the richness of the experience. MOO is a system of texts within texts. The layer of background context is defined textually; player interaction is an exchange of textual messages; objects in the environment are represented textually. Each layer occupies a different part of the user’s imagination and attention at different times. Jeffrey Young describes it this way:

...the text gains a unique blend of *transparency and opacity*, as players constantly shift stance from immersion in the imaginative space to evaluation and control over the textual objects.⁴⁶

To continue with the idea of documents being texts incarnate in an interpretive community, how many layers of “documentation” are present (and active) within a MOO world? The whole thing is textual, and while some of its components look something like traditional documents, other elements and aspects challenge traditional understandings of what documents are. Can we really be said to exist and interact *within* a document? There are schools of literary criticism that would not take issue with this concept at all;⁴⁷ in the MOO context, though, it is *literally* true (pun intended), and on a very immediate level.

Forms of Content in MOO Environments

Content in a MOO—though always textual—is handled in a number of different ways. Each different form implies a different set of interactions and functions. The following is a brief summary of these forms.

The most basic form of MOO content is *topological*, to use the terminology of LambdaCore's internal help system. *Topology* in this sense refers to the virtual environment described in spatial and architectural terms. The basic unit of MOO architecture is the *room*, which can be functionally defined as a top-level container (and child of the generic *\$room*), and within which can be found players, other objects, sub-containers, and so on. Each *room* has a description, and this description is displayed automatically as a user enters the room. Movement in the MOO—from room to room—will produce a series of descriptions that together describe the overall topology of the MOO.

Other objects in the MOO—players, things, containers, notes—also have descriptions that one can see by typing the *look* command. At the most basic level, creating a virtual environment in the MOO is a matter of describing places and the things in them, and arranging these so that the user, in moving through the environment and encountering different objects, has a coherent experience, perhaps like that of reading a novel; architecture as literature.

A particularly important class of objects in the MOO is the *note* class; that is, descendants of the generic *\$note* object. Notes are functionally defined as objects that one can write on and read. The *note* object class has been elaborated to include different sorts of notes for different purposes, including *letters* (that can be “burnt” when read), and an internal e-mail system.

It is particularly *noteworthy* that in a world created entirely of words, there is a class of objects that exist to put words on. There is a peculiar frame

shift in this—creating a distinction between words that are to be treated as 'reality' and words that are to be treated as words.

A last form of content and communication in the MOO environment is live dialogue and/or interaction with live objects (meaning connected players as well as programmed, animate objects). Conversational functions in a MOO are designed in such a way that what scrolls up the user's screen looks roughly like a dialogue in a novel.

```
Jack comes in from the Library.  
The Host says, "Greetings, Jack!"  
Jack asks, "Hey! What's happenin'?"  
Jill hands Jack a cookie.  
You say, "Jill baked some cookies."  
Jack devours the cookie.
```

This is decidedly different from other network chat services both in the syntax used and in the integration with the rest of the environment. While the tone of MOO conversations may not be wholly different from those in a chat room, they are significantly grounded in the surrounding environment, both by reference to the environment and structurally in terms of the way the text flows on the user's screen.

Ideally, a log of a MOO session reads something like a bizarre novel. The plot may or may not be completely nonsensical, but the structure of the written language holds together as a coherent, second-person, present-tense account of what is happening, with whom, and where. Plain, unedited log files of MOO sessions are a common archival form within the MOO community on the Internet. A MOO log can be read as a simple narrative, because the system is designed to generate simple narrative output to each user.

The idea of a MOO experience being like that of reading a novel is a useful one. However, unlike a novel, which is composed as a linear narrative, a

MOO environment is designed and composed as short, discrete textual passages chained together by a player's actions. Hence, the aesthetics of MOO description take a cue from emerging literary hypertext forms in demanding a certain level of second-guessing on the part of the author. For instance, in *NooDDLe*, two room descriptions in the outdoor *Garden* appear like this:

The Garden

A wide opening in the forest which reveals a lush and verdant patch of sunlight. The Garden is tall and verging on overgrown now, in high summer. A series of groomed paths lead you through the foliage. A thicket of roses insulates you from the forest to the east, and toward the western perimeter are a number of laden fruit trees: cherries, pears, and a fig. In amongst the trees are a number of cedar buildings.

You see The Host (wearing a summery straw hat) here.

Go: [n] to North Garden, [s] to South Trail, [steps] to The Courthouse, and [w] to West Garden

> w

West Garden

Here stands the majestic fig tree, small green figs hanging from its branches. Beneath the fig is an overgrown patio, made of dark red paving stones, that leads back to an almost-hidden doorway, framed in dark ivy. Now and then you can hear voices drift down from above.

Go: [e] to The Garden and [door] to JMax' Workshop

In composing these descriptions, I can refer specifically to the fig tree in the second description, because I am confident that anyone reading the *West Garden* description will have first seen the description for the *Garden*. I can be confident of this because of the arrangement of rooms.

A similar sensitivity needs to be exercised in the design of automated objects like the *Host*. The *Host* is programmed to greet visitors to the *Garden*

and to take players on a tour of the courthouse when requested. While it was a fairly easy task to make the *Host* wander from room to room, offering brief comments at each place, it was not so easy to tune this behaviour so that it appeared to be coherent, deliberate, and above all, tolerant of players who do not tag along dutifully.

I called this sensitivity “second guessing”, but in truth it is the basis of interface design. In the context of programming a computer, interface design can be a messy problem because of the large number of presentation variables. Interface design exists in the traditional publishing world as well, but there it has become a craft based on a set of time-honoured devices and strategies. In the emerging digital world, publishers are becoming cognizant of computer interface design, and programmers are becoming aware of traditional publication design.

Player Interaction in the MOO

Every object in the MOO has an *owner*; that is, every object has an inherent set of relationships to specific players in the MOO environment. The owner of a particular object—generally its creator—enjoys complete power over the object: to create or destroy it, to manipulate it, to change what it looks like or what it does. Other players have a more limited relationship to it; they may see the object, interact with it, or move it, but they cannot destroy or change it.

The system of rules governing these relationship is modeled on the system of permissions from the Unix operating system;⁴⁸ files and processes can be read, written to, or executed (used) by the owner, a group of specified others, or anyone at all, according to a specific set of codes describing these permissions.

In the MOO, object permissions act as a set of inalienable rights that are 'hard-wired' into the virtual world.

Different players have different kinds of powers, as well. I have mentioned the MOO's *Wizard*, who is roughly analogous to the system administrator. The Wizard has complete access to everything; there are no barriers to the Wizard's powers to create, destroy, and modify objects in the MOO, including creating and destroying players themselves.

Most MOO users will not log in as *wizard-class* players. Most players are either of the generic *player*, *builder*, or *programmer* class, defining their level of access to the MOO's programming language. They have the power to create, modify, and destroy their own objects, but not those of other players. It is also possible to 'open up' the permissions on specific objects, so that others could modify or destroy one's things (for example, letters that can be "burnt" after the recipient reads them).

This hierarchy of property and privilege provides a basic context for the goings-on in a MOO. To explore this, consider my *virtual Courtroom*.

As a *programmer-class* player, I have created a *Courtroom*.⁴⁹ The *Courtroom* object and related objects are therefore 'mine' and are not subject to the control of anyone other than myself (and, of course, the all-powerful Wizard).

Once created, the *Courtroom* is linked in to the rest of the MOO and opened to the MOO public. Participants can be invited into the *Courtroom*, shown to various materials, instructed on the basic use of the facility, and charged with the business of conducting a mock trial.

Once a trial is under way, players will be actively adding to the virtual environment in a variety of ways. First, they will be present in the *Courtroom*,

and so the room will take into consideration their presence and roles in the way it arranges the proceedings and descriptions. Second, they will probably be manipulating and adding objects to the *Courtroom* environment: documents (notes), evidence, and so on. Finally, they will be contributing to the *court record*, which is essentially a log of the happenings in the room.

The goings-on in the *Courtroom* are a collaborative process. The trial is collaboratively authored by the participants, myself (as author of the *Courtroom* object itself), and Sandra Hawkins (as the designer of the mock trial proceedings). This collaborative authoring is not true merely on an abstract literary level; it is explicitly true in the sense that the trial is a text that scrolls up each participant's screen (and which is also logged internally by the *court recorder* object).

In the sense that this is a collaborative and dynamic process governed by a set of basic rules, the challenge is to create a kind of structured chaos. The MOO experience is not like that of a text adventure game, in which there is a pre-planned 'world' that one must explore and puzzles to solve. Rather, the MOO experience is a dynamic happening in a structured shell. It is nicely analogous to the interaction in a real-world courtroom; the rules that govern what goes on in a real court are very strict. As a result, the text that emerges out of courtroom interaction is coherent, structured, and usable as the basis for our cumulative legal system. What goes on within the strict framework of court (or MOO), however, is unpredictable.

MOO Aesthetics

Players connect to a MOO by way of a telnet session to a specific port on the host computer. Once connected, the MOO's *welcome message* appears onscreen

and the user is prompted to log in with name and password. Unlike a Unix system, MOO will only accept one login at a time for each player; that is, it is not possible to log in as the same player twice simultaneously. There can only be one of you in the MOO.

Once logged in, commands are typed as simple English sentences (or fragments thereof). The MOO attempts to parse commands according to the basic structure of *verb – direct object – preposition – indirect object*. It then attempts to match each of these syntactical elements to objects in the MOO database. If they are successfully matched, the commands are handled by appropriate *verbs*⁵⁰—MOO programs written to handle those specific commands. For instance, the following command:

```
throw frisbee to Fred
```

will be successful if there is a player named *Fred* in the room, there is an object called *frisbee*, and there is a *throw* verb defined on the *frisbee* object. If any of these matches fail, the MOO will respond with a simple error message saying that it doesn't understand what you mean.

The elegance of a MOO environment has much to do with the mapping of these command handlers to what one would expect from real-world discourse. Ideally, the commands a user types are grammatical sentences relating to the action to be performed. For instance, if there is a *park bench* in the MOO, then we should presumably be able to “sit on the bench” or “sit down on the bench”. If we have to “blorg gimble garfle bench” in order to seat ourselves on it, the virtual illusion suffers.

As commands are entered in this fashion, the MOO responds by reporting actions and changes to the virtual environment: “You sit down on the bench... Marty arrives... Marty says ‘Hi’... Marty sits down on the bench,” as

they happen. The resulting stream of text scrolls up the user's screen as the events happen.

Conversation and movement in the MOO are the simplest forms of user interaction. The command, *say Hi*—usually abbreviated to *"Hi* (a quotation mark stands for the command *say*)—will cause *John says, "Hi"* to appear on the screens of any players in the same room (if Bob typed the command, it would appear that Bob said, "Hi"). Typing *north* or even just the letter *n* will move you (your *player*, that is) to the 'north', assuming there is an available exit and room in that (virtual) direction.

Other, more complex interaction is also possible. An object can be created to respond to any typed command and respond in whatever way the object's creator wants. For instance, *note* objects respond to a command:

```
write "This is a line of text" on note
```

...by adding the text between the quotation marks to the object's set of properties. *Note* objects also respond to a *read* command, which prints out the text that had been *written*. The variety of these command/response interactions is truly only limited by imagination.

Second, there is a layer of direct interaction with the MOO database itself. A set of commands exists within the MOO that allow the user to manipulate the database directly. For instance, the command *@describe* allows one to set the description of any owned object. The command *@who* results in a list of all currently connected players. Commands like these are generally distinguished by the *@* prefix.

Finally, there is the actual programming of the environment. Programming is a matter of composing scripts written in MOO code and

attaching them to specific objects, so that they respond to triggers such as typed commands.

All of these interactive modes are integrated in the one interface. From anywhere in the environment, and at almost any time, one can type commands to engage in conversation, query or make changes to the database, or actually program objects in the environment.

As simple as this command-parsing interface is, it is also MOO's greatest limitation. There is no simple way to suspend or break out of the live interaction model. This is most irritating, at least on the day-to-day level of building in a MOO environment, because programming must be performed via the same live interface. A simple type of text editor is included as part of the LambdaCore database, which allows a player to enter a *room* supposedly conducive to writing and editing MOO code, but this is very kludgy and unsatisfactory for all but the most trivial editing chores. The alternative is to compose MOO code outside the MOO and then cut and paste the edited code into the MOO.⁵¹ In doing so, however, one is 'breaking frame' by removing one's attention from the immediate experiential level.

This gripe, however, doesn't slow dedicated MOOers down much, nor has it hindered the evolution or propagation of MOOs on the Internet. Thousands of users in dozens of Internet MOOs manage to get along fine with the internal editing system, even to the point of making extensive use of the internal MOO mail system—something I personally consider almost too cumbersome to use. The interface is not always simple and elegant. It sometimes presents an obstacle to new users and probably has hindered the overall public awareness of MOOs and MOOing.

It is not that the interface is complicated or hard to use; just that it takes a certain leap of faith and a half hour of feeling disoriented before a user becomes acclimatized to the ways of the environment. This steep—however brief—initial learning curve has proven to be the biggest obstacle in presenting *NooDDL*e to the OLA community.

It is worth noting that while MOOs remain for the most part entirely textual environments, the internal object hierarchy of MOO can serve as a structure for delivering more than plain ASCII text. MOO servers can be made to communicate with Web browsers, delivering objects in the form of Web pages.⁵² Another approach is a MOO client called *Pueblo*,⁵³ which can interpret HTML tags. By embedding HTML codes directly into object descriptions, fully formatted, colour-enhanced MOO output can be generated. In the next year or so, we can expect to see this sort of convergence of Internet technologies blossom into a wealth of new possibilities.

Interface Design Tactics

Within the text-only world of existing MOOs, there are certain measures that a MOO designer can take in order to make the environment friendlier for new users. The basic strategy is to think about what the community really needs to do, and then customize the interface to facilitate it. An example of this is a new subclass of objects I created in *NooDDL*e: the *document* object. Recall that there is an object class of *notes* in the LambdaCore that allows one to write text onto them and read that text back at will. The command to enter text on a *note* object looks like this:

```
write "This is a single line of text." on MyNote
```

While this command is fairly straightforward, it is rather limited when it comes to adding more than a single line to the note. In order to add multiple lines, the user must issue the command repeatedly.

My solution was to create a new subclass of *notes* called *documents*, with a new verb called *writeln*, which works like so:

```
writeln MyDocument
```

After issuing this command, the user is then prompted to add as many lines of text as desired, and, when finished, to type a single period at the start of a new line—telling the MOO that the text has ended.

Writeln is no more 'obvious' than the original *write* command, but it is a more convenient way to enter long passages on document objects, and it has facilitated the creation of an in-MOO library of reference materials.

This sort of specialization of objects is at the heart of MOO design. The *Guest Book* that players will find in the *Courthouse* is a further specialization of the *document* class; the difference is that the object permissions have been 'opened' to allow anyone to *writeln* the object, not just the object's owner.

Another example is the *pose* feature, a superb piece of MOO code created by a player named Quinn on Xerox' *LambdaMOO*.⁵⁴ *Pose* allows a player to type commands in this form:

```
.smile at Kelly
```

This results in a set of perfectly conjugated sentences: depending on whose screen it appears on, either "you smile at Kelly," "John smiles at Kelly," or "John smiles at you." The *pose* feature adds sophistication and convenience to the basic set of player functions, which would otherwise print generic messages on

the screens of everyone in the same *room*—without performing these individual conjugations.

The point of these examples is to demonstrate that MOO objects and the environment they create are entirely plastic, and can be altered to suit one's needs or wishes. In terms of environmental design, the creation of specialized objects is an important layer in the interpretation and presentation of content material.

PUBLISHING ENVIRONMENT vs. PUBLISHED ENVIRONMENT

The multiplicity of layers in the MOO environment—layers of presentation, facility, and meaning—results in a publishing framework we can look at in at least two ways. On one hand, we can approach MOO technology as a publishing medium on the Internet, much as the WorldWide Web is a publishing medium. It allows an audience to access a body of interpreted and edited material and to read and interact with it.

On the other hand, we can look at MOO as containing an entire virtual world, *within which* we can present documents to an audience or community. In this scenario, entities within the environment interact with specific textual artifacts, and the relationship between text and community is defined within the larger context of the MOO. In looking at the Law 12 mock trial exercise, I think that this latter perspective is more appropriate.

In either scenario, MOO technology involves a wealth of document forms. The variety of textual layers within the MOO world parallels the blossoming of digital document forms outside the scope of MOO itself.

A notable case in point is the body of information available about MOO technology. Most of the technical and research documentation on MOOs and MOOing is available on the Internet at various locations. There are Web-based hypertexts and link collections, FTP file archives containing all sorts of document forms: ASCII text files, MSWord files, Adobe Acrobat documents, TeX files, and raw PostScript files. There are half a dozen Internet mailing-lists, a similar number of Usenet newsgroups, three or four different FAQ collections,

and searchable archives of most of these. Exploring this subject area is itself a course in dealing with digital document formats.

But there is a separate body of material that is only available *inside* MOOs. For instance, in LambdaCore, there is the internal *help database* that is the last word on many of the details of MOO. Additionally, in established MOO environments, there is usually a large body of user-contributed material. For instance, one of the best sets of documentation on MOO security issues resides in a virtual library within *JaysHouseMOO*; it exists publicly nowhere else on the Internet (to the best of my knowledge). Some MOO documents can be found in both environments—inside MOOs and outside MOOs—but most fall into one category or another. It is interesting to look at where certain types of documents reside and thus at what groups of people have access to them.

Partly, this characteristic has to do with the inclusive nature of MOO worlds; they are designed to be administered and experienced entirely from within. Certain meta-topics, such as the details of server installation, are clearly external to this process. The *MOO-Cows* mailing list is a good example of this; discussions are almost always limited to the technical details of running MOO servers. Other topics, though, are clearly pertinent to the internal management of a MOO world. In the case of Xerox' *LambdaMOO*, there are hundreds of internal mailing lists—within the MOO environment—that deal with the intricacies of a judicial and political system that has evolved over many years. Not a word of this appears outside of *LambdaMOO* until it is reported third hand.

MOOs (and their inhabitants) are not, as a general rule, isolationist or xenophobic. There is, on Internet mailing lists and in many of the large MOOs themselves, a sense of an overall MOO community. Many objects and features

have been *ported* from MOO to MOO, and there is an established etiquette for doing so and giving proper credit to the authors. Many MOOers appear in a number of different MOOs, often recognizably and under the same names.

The broader MOO universe appears to operate on three vague levels: the first being the internal goings on of a particular MOO world. Second is the meta-information level of the *MOO-Cows* mailing list and others of its kind, in which generic technical and social issues can be discussed. Third is a supra-MOO sensibility or awareness that transcends these other two spheres. At this level, there is a overall community that recognizes the distinctiveness of different MOO worlds and the populations that inhabit them, but that allows for a common MOO culture between them.

I had an interesting experience one morning last spring that illustrates this point. I logged in to *JaysHouseMOO* (a popular hangout for MOO programmers), and after I had convinced the assembled group that I wasn't a refugee from *LambdaMOO's* "living room" (a 24-hour house-party environment that many find tedious, but which spills over to other MOOs when the *LambdaMOO* machine is temporarily unavailable), I had an interesting chat about the Law 12 MOO project with the very same people I had been reading over the last few days in the *MOO-Cows* mailing list on the topic of core database upgrades. But instead of talking about core upgrades here, we idly discussed the Law MOO project; some expressed interest and some skepticism (that mock trials could work in this environment). The interesting thing was the multiple levels of access that I had to these same people and the differences in tone and content at each level.

Stanley Fish says that the interpretive communities we belong to construct the meanings of the texts we read. But it also seems true, in these exam-

ples, that the documents we publish serve to establish and define those very same communities. John Seely Brown and Paul Duguid of Xerox PARC suggest that documents serve to define and negotiate boundaries between communities, both in providing needed cues to distinguish groups and also in “bringing people from different groups together to negotiate and coordinate common practices.”⁵⁵

MOOs, interestingly, are both literally the documents *and* the community, and the two define one another. The textuality of a MOO environment is defined and created by its inhabitants, while at the same time, those inhabitants exist within the MOO as text. The community that exists within a MOO, like the community of MOOers in general, is defined and held together by the set of documents that compose the environment. The overlapping constituencies of various MOO worlds, MOO mailing lists, and MOO research are each in turn defined by the circulation of specific documents. And at the same time, the document forms that make up both specific MOO environments and the broader MOO world are continually created by those communities.

John Unsworth, of *PMC-MOO*—a MOO set up originally as an extension of the electronic journal *PostModern Culture*—points out that “MOO is the world it models, as well as the model of that world.”⁵⁶ In many ways and on many levels, the content of MOO is MOO itself. But if MOO as medium and MOO as content are bound in so tight a circle of text defining community defining text—*ad nauseam*—does this speak to the issue of MOO not being about anything, really? Or does it point to a more general principle of increasing reflexiveness in digital media, in which the tight relationship between documents and communities represents an overall collapsing of the relationship between self and other?

Text, Document, and Community

As I proposed earlier in this paper, a document is a specific incarnation of a text; it is text as an artifact. A text may be incarnate in many different document forms in its lifetime: a word processor file; a manuscript in circulation among editorial staff; a hardcover book on a display rack; a tattered, well-thumbed paperback; a digital file in an Internet archive; and so on. Each document, as a textual artifact in space (perhaps cyberspace) and time, goes some distance toward defining:

- the coverage and dissemination of the text
- its lifespan and currency
- its mode of transferal
- its relationship to various interpretive communities

In taking a text and embodying it in various document forms, we are setting up a different set of artifactual, interpretive, and social dynamics around the same text. Or is it, in fact, the same text? A document provides a host of contextual clues to interpretation—provenance, age, authority, cost, and value. The textual content cannot be considered without reference to this frame.

Stanley Fish tells us that interpretive communities are the structures that allow texts to be shared and understood in useful ways. But text has to have artifactual reality in order to be shared and considered, and that reality defines its relationship to an audience.

John Seely Brown and Paul Duguid talk about documents defining and underwriting social realities. But since publishing—the process of documentation—is itself a social and interpretive activity, what we have is an active dialogue between document and community.

Consider the case of a (real world) courtroom. Our legal system is a set of documents that guides and shapes how justice is to be interpreted and dealt. At the same time, the process of the trial itself is the active creation of a text that will come around again as part of that guiding body of documents. There is a very immediate interpretive and creative process going on, the result of which is a community of active negotiation.

The same happens in a MOO. Participants enter a world of documents that create a virtual reality. At the same time, the process of MOOing is the creation of those very documents that define the MOO community.

This is not a new idea; the same processes and dynamics exist with documents and document communities everywhere, in whatever form they may appear. In the case of MOO technology, though, the workings of the process are fast enough and immediate enough to be readily apparent.

The *speed* of documentation and interpretation in this sense is significant to my discussion, because the trend toward "being digital" means things are speeding up all over. The kinds of social and economic dynamics that may previously have been noticed only by historians are increasingly visible on a day-to-day level, and perhaps faster still.

Some Final Reflections on the Process

The process of designing and developing the *NooDDLe* environment was one of constant negotiation with a host of uncontrollable circumstances, personal, technological, and social. What was imagined last May is not quite what has emerged this September. We are both ahead of and behind schedule. Parts of the project are less elaborate than planned, some are more so. I have the good

fortune of being able to further develop the environment and see it through a real *beta-test* stage this winter.

NooDDLe, as it exists right now, is complete in terms of the dynamics and design strategies I have outlined here. It still awaits its real user community, however. The current plan is to bring Sandra Hawkins' students into the environment in November for the mock trial exercise. At that time, we will find out if the students can make sense of the environment I have designed. It makes sense in my mind, but will it exist in the minds of a dozen Law 12 students who have had no part in the design process? It is not at all clear (yet) whether MOO works for the casual participant in the same way it works for the builder. Is my "document community" a community of one (or perhaps of the handful of people who have followed my development)?

The analysis of the project that I have presented here does not directly address any specific educational or pedagogical issues. I have approached this paper from the standpoint of interpreting and presenting a body of content and from my own perspective on immersive environments. In November 1996, the real test of the environment will take place: Sandra Hawkins' Law 12 class will be invited into *NooDDLe* for their mock trial exercise. The MOO will at that time be evaluated by myself and by NDDL staff on the basis of its educational efficacy, and its *fit* within the NDDL model, a step which will add significantly to what this report offers. Further analysis of this technology can be expected.

Finally, it is worth noting that advances in client technology did not emerge as quickly as we had expected. Last spring, I believed that we would very soon be seeing multimedia, styled-text MUD clients in the form of Web-based Java *applets*. Alas, the real state of Java development is somewhat behind the media and marketing hype. While there *are* Java-based MUD client

applets, we are not at a point where I feel comfortable unleashing them on an unsuspecting audience of Macintosh users.

I am, however, very intrigued by what Chaco's *Pueblo* client offers in an integrated package. With *Pueblo*, the MOO client incorporates the Web browser, instead of the other way around. Unfortunately, the *Pueblo* client is not available as a Macintosh application—which leaves the majority of my audience out. There is still a lot of work to be done in this area. ♣

NOTES

1. Pavel Curtis, *Mudding: Social Phenomena in Text-Based Virtual Realities* (Submission to the 1992 conference on Directions and Implications of Advanced Computing, 1992)
2. Curtis (ibid.) aligns MUDs and MOOs with McLuhan's idea of a "cold" medium—one that demands a relatively high degree of involvement and participation on the part of the reader—distinct from graphic VR, which would be a "hot" medium.
3. Lauren P. Burka, *A Hypertext History of Multi-User Dimensions* (1993)
4. *LambdaMOO* was not the first or the only object-oriented MUD, but it is by far the most popular implementation and the basis for almost all present MOOs. See Curtis, *Mudding*.
5. Object-oriented systems allow for re-use of code and the inheritance of properties and functionality. One "barking dog" object can be used as the basis for hundreds of virtual pets, with a minimum of (technical) redundancy.
6. *LambdaMOO* is reachable by telnet connection to:
`lambda.parc.xerox.com 8888`
7. See Curtis, *Mudding*; Julian Dibbell, "A Rape in Cyberspace," *Village Voice* (December 21, 1993).
8. *Diversity University* is reachable at: `moo.du.org 8888`
9. *Virtual Online University* maintains the *Athena* campus, reachable at:
`athena.edu 8888`
10. *Virtual Online University, Inc.: An Overview*.
11. From a 1994 conversation with Michael Bertsch, instructor and Director of Academic Affairs at *Virtual Online University*.

12. Some notable MOO-based English and Writing projects include:

University of Pennsylvania's *PennMOO*:

<http://www.english.upenn.edu/PennMOO/>

University of Texas *Creative Writing Research Lab*:

<http://www.cwrl.utexas.edu/moo/>

University of Missouri's *MizzouMOO*:

<http://www.missouri.edu/~moo/>

Phoenix College/Longview Elementary

Pueblo Global Learning Collaboratory:

<http://pc2.pc.maricopa.edu/>

13. The *MicroMUSE Charter and Bylaws* (1991)

14. Constructionism "asserts that learning is an active process, in which people construct knowledge from their experiences in the world. (This idea is based on the theories of Jean Piaget.) To this, constructionism adds the idea that people construct new knowledge with particular effectiveness when they are engaged in constructing personally-meaningful products." – Amy Bruckman, *MOOSE Crossing: Creating a Learning Culture*, Thesis Proposal for the degree of Doctor of Philosophy (Massachusetts Institute of Technology, December, 1994).

See also Seymour Papert, *Mindstorms: Children, Computers, and Powerful Ideas* (New York: Basic Books, 1980); Seymour Papert, "Situating Constructionism," in *Constructionism*, ed. I. Harel and S. Papert (Norwood, NJ: Ablex Publishing, 1991)

15. Bruckman, *MOOSE Crossing*.

16. Tari Lin Fanderclai, "MUDs in Education: New Environments, New Pedagogies" in *Computer-Mediated Communication Magazine*, vol 2, no 1 (January 1, 1995)

17. Amy Bruckman, *Identity Workshop: Emergent Social and Psychological Phenomena in Text-Based Virtual Reality* (MIT Media Lab, 1992)

18. Amy Bruckman and Mitch Resnick, *The MediaMOO Community: Constructionism and Professional Community* (MIT Media Lab, 1995)

19. John Allison, *MOOs and Education: Their Role, and Relevance*, Ontario Institute for Studies in Education (1995)

20. Michael Heim, "The Erotic Ontology of Cyberspace," in *Cyberspace: First Steps*, ed. Michael Benedikt (Cambridge: MIT Press, 1991)
21. Eva-Lise Carlstrom, *Better Living Through Language: The Communicative Implications of a Text-Only Virtual Environment* (1992)
22. Marcos Novak, "Liquid Architectures in Cyberspace," in *Cyberspace: First Steps*, ed. Michael Benedikt (Cambridge: MIT Press, 1991)
23. Rémy Evard, *Collaborative Networked Communication: MUDs as Systems Tools*, Proceedings of the Seventh Systems Administration Conference (LISA VII), pages 1-8 (November 1993)
24. The presentation of self in cyberspace. See:
- Jill Serpentelli, *Conversational Structure and Personality Correlates of Electronic Communication* (1992)
- Elizabeth M. Reid, *Cultural Formations in Text-Based Virtual Realities* (1994)
25. Gender in virtual environments. See:
- Lynn Cherny, *Gender Differences in Text-Based Virtual Reality*, Berkeley Conference on Women and Language (April 1994)
- Amy Bruckman, "Gender Swapping on the Internet," *Proceedings of INET '93* (Reston, VA: The Internet Society, 1993)
26. Ethnography and discourse analysis in cyberspace. See:
- Michael S. Rosenberg, *Virtual Reality: Reflections of Life, Dreams, and Technology* (1992)
- Lynn Cherny, *The Modal Complexity of Speech Events in a Social Mud* (1995)
- Don Langham, "The Common Place MOO: Orality and Literacy in Virtual Reality," *Computer-Mediated Communication Magazine*, vol 1, no 3 (July 1, 1994)

27. In her proposal for the *MOOSE Crossing* project, Amy Bruckman suggests that "MUDs can be compared not just to school, but also to television and books. A television viewer is a recipient of commercially produced content. A MUD player is a collaborative creator of media content."

28. Curtis, *Mudding*.

29. *Internet Relay Chat* (IRC) is a distributed, Internet-based live chat application. When one logs in to an IRC "relay", one has the option of choosing one of several hundred "channels", each devoted (more or less) to a particular topic. Participants are identified by nicknames, and an elaborate IRC culture is apparent.

30. *New Directions in Distance Learning* is an ongoing project of the Open Learning Agency and the British Columbia Ministry of Education, Skills, and Training. See: <http://www.ola.bc.ca/schools/>

31. *FirstClass* is a bulletin-board, conferencing, and groupware environment developed and distributed by SoftArc, Inc. FirstClass employs a icon-based file-and-folder user interface very much like the Macintosh "Finder" interface. Inside folders are conferences, made up of individual messages that one can double-click on to read. See: <http://www.softarc.com/>

32. David Porter et al, *NDDL 1994-95: Phase 2 Review*. Executive Summary (1995)

33. Sandra Hawkins has written (with Judge C.C. Barnett) a number of mock trials in use by various legal education organizations. The trials are published in the form of short booklets containing notes and details of each case. Often, the trials are based on real cases from British Columbia courts.

34. *R. v Howard: a Mock Trial* is the case of a high-school industrial education teacher who struck a student on the head with a hammer in order to make him pay attention. The trial involves balancing a section of the *Criminal Code* that allows teachers to take whatever "corrective measures" are necessary with a section of the *British Columbia Public School Act* forbidding corporal punishment.

35. Stanley Fish, *Is There a Text in this Class? The Authority of Interpretive Communities* (Cambridge: Harvard University Press, 1980)

36. *NooDDLe* is the working title of the MOO, for *New Object Oriented Directions in Distance Learning Environment*. Note that I did not opt for the more obvious *MOO Court*.

37. *Bots* (from "robot") are automated programs that appear to be human. They are artificial intelligences (to use the term loosely) that inhabit online spaces like MUDs and IRC. For an excellent discussion, see Lenny Foner, *What's an Agent Anyway: A Sociological Case Study*, MIT Media Lab (1993)

38. "Frame" in this context comes from Erving Goffman, *Frame Analysis: An Essay on the Organization of Experience* (1974; reprint, Boston: Northeastern University Press, 1986)

39. The MOO server and core are available from Xerox PARC's anonymous ftp archive, at <ftp://parcftp.xerox.com/pub/MOO/>

40. Steve Caron and Nick Ingolia have produced an excellent port of the *LambdaMOO* server for the Macintosh called *MacGoesMOO*.

<http://neon.ci.lexington.ma.us/SpamCentral/scaron/mgm.html>

A Windows NT and '95 port is made available by Christopher Unkel.

<http://www-personal.engin.umich.edu/~cunkel/WinMOO/winmoo-faq.html>

For a discussion of my experiences with an early version of the Macintosh port, see the *Phase One Project Report* that accompanies this paper.

41. We chose *MUDDweller 1.2*, a freeware MUD client for the Macintosh written by Oliver Macquelin. For Windows users, we chose *Phoca 1.1f*, a freeware version of software written by Lawrence Adams. Both clients were chosen for their simplicity of interface.

42. As I write this (late September, 1996), *NooDDL*e is down, due to a hardware-level disk crash last week. We are awaiting a replacement disk.

43. Generic objects are referenced in LambdaCore with a preceding \$ sign. Hence, *\$note* refers to "the generic note".

44. Chip Morningstar and F. Randall Farmer, "The Lessons of Lucasfilm's Habitat," in *Cyberspace: First Steps*, ed. Michael Benedikt (Cambridge: MIT Press, 1991)

45. There is a fully functional e-mail system within LambdaCore—well-used on many MOOs; as well, the MOO server is capable of operating as an Internet e-mail program.

46. Jeffrey R. Young, *Textuality in Cyberspace: Muds and Written Experience* (1994)

47. Cf. Derrida's "There is nothing outside the text." See Barbara Johnson's "Translators Introduction" to Jacques Derrida, *Dissemination*. (University of Chicago Press, 1981): xiv

48. John Unsworth, *Living Inside the (Operating) System: Community in Virtual Reality (Draft)* (1995)

49. I could have created the *Courtroom* as the Wizard, but this would be an unnecessary use of power, since there is no need to overstep the privileges of my programmer-class self.

50. In MOO parlance, a program attached to an object is called a *verb*, because it is generally called by the verb part of a parsed MOO command. The server attempts to parse all typed commands as *verb, direct object, preposition, indirect object* and then to match these with objects in the MOO database. *Say, look, go, get, and @quit* are all *verbs* written on specific objects.

51. There are some notable client programs, such as the one written for Amy Bruckman's *MOOSE Crossing* project that solve this problem by offering a client-side editing window.

52. A number of MOO-Web experiments are underway in different MOOs, attempting to allow users to browse the contents of the MOO with a program like Netscape, or conversely, making the MOO server act like a Web server. While these offer a more interesting visual interface than a terminal window, they sacrifice the live interaction mode almost completely.

53. Chaco's *Pueblo* client enhances the existing MOO interface by interpreting embedded HTML tags within the MOO output. The interaction remains live, but the output is formatted, and can contain multimedia elements just like a Web page can. The *Pueblo* client is thus a very interesting hybrid Web browser/terminal emulator. See: <http://www.chaco.com/pueblo/>

54. I "ported" Quinn's *pose* feature object to *NooDDLe* with his permission, from a second-generation copy at MIT's *MediaMOO*.

55. John Seely Brown and Paul Duguid, "The Social Life Of Documents," in *f j ® s † - m ø Ñ d @ ¥*. Issue 1. (1996)

56. Unsworth, *Living Inside the Operating System*

BIBLIOGRAPHY

Allison, John. *MOOs and Education: Their Role, and Relevance*. Ontario Institute for Studies in Education. 1995.

<http://www.oise.on.ca/~jallison/abst4.html>

Benedikt, Michael, ed. *Cyberspace: First Steps*. Cambridge: MIT Press, 1991.

Brown, John Seely and Paul Duguid, "The Social Life Of Documents." in *fj@s†-møÑd@¥*. Issue 1. 1996.

<http://www.firstmonday.dk/issues/issue1/documents/index.html>

Bruckman, Amy and Mitch Resnick. *The MediaMOO Community: Constructionism and Professional Community*. MIT Media Lab. 1995

<ftp://ftp.media.mit.edu/pub/asb/papers/convergence.txt>

———. *Identity Workshop: Emergent Social and Psychological Phenomena in Text-Based Virtual Reality*. MIT Media Lab, 1992.

<ftp://ftp.media.mit.edu/pub/asb/papers/identity-workshop.ps>

———. "Gender Swapping on the Internet". *Proceedings of INET '93*. Reston, VA: The Internet Society, 1993.

<ftp://media.mit.edu/pub/asb/papers/gender-swapping.txt>

———. *MOOSE Crossing: Creating a Learning Culture*. Thesis Proposal for the degree of Doctor of Philosophy. Massachusetts Institute of Technology. December, 1994.

<ftp://ftp.media.mit.edu/pub/asb/papers/moose-crossing-proposal.txt>

Burka, Lauren P. *A Hypertext History of Multi-User Dimensions*. 1993.

<http://www.utoxia.com/talent/lpb/muddex/essay/>

Carlstrom, Eva-Lise. *Better Living Through Language: The Communicative Implications of a Text-Only Virtual Environment*. 1992.

<ftp://parcftp.xerox.com/pub/MOO/papers/communicative.txt>

- Cherny, Lynn. *Gender Differences in Text-Based Virtual Reality*. Berkeley Conference on Women and Language. April, 1994.
<ftp://parcftp.xerox.com/pub/MOO/papers/GenderMOO.ps>
- . *The Modal Complexity of Speech Events in a Social Mud*. 1995.
<http://lucien.sims.berkeley.edu/MOO/ejc.txt>
- Curtis, Pavel. *Mudding: Social Phenomena in Text-Based Virtual Realities*. Submission to the 1992 Conference on Directions and Implications of Advanced Computing .
<ftp://parcftp.xerox.com/pub/MOO/papers/DIAC92.txt>
- Derrida, Jacques. *Dissemination*. Translated by Barbara Johnson. University of Chicago Press, 1981.
- Dibbell, Julian. "A Rape in Cyberspace." *Village Voice*. December 21, 1993.
<ftp://parcftp.xerox.com/pub/MOO/papers/VillageVoice.txt>
- Evard, Rémy. "Collaborative Networked Communication: MUDs as Systems Tools." *Proceedings of the Seventh Systems Administration Conference (LISA VII)*. November, 1993.
<http://www.ccs.neu.edu/home/remy/documents/cncmast.html>
- Fanderclai, Tari Lin . "MUDs in Education: New Environments, New Pedagogies." in *Computer-Mediated Communication Magazine*. vol 2, no 1. January 1, 1995.
<http://sunsite.unc.edu/cmcmag/1995/jan/fanderclai.html>
- Fish, Stanley. *Is There a Text in this Class? The Authority of Interpretive Communities*. Cambridge: Harvard University Press, 1980.
- Foner, Lenny. *What's an Agent Anyway: A Sociological Case Study*. Agents Memo 93-01. Agents Group, MIT Media Lab, 1993.
<http://foner.www.media.mit.edu/people/foner/Julia/Julia.html>
- Goffman, Erving. *Frame Analysis: An Essay on the Organization of Experience*. Boston: Northeastern University Press, 1986 (1974).
- Hawkins, Sandra, and C.C. Barnett. *R v Howard: A Mock Trial*. Law Courts Education Society of B.C. 1995.

- Langham, Don. "The Common Place MOO: Orality and Literacy in Virtual Reality." *Computer-Mediated Communication Magazine*. vol 1, no 3. July 1, 1994.
<http://sunsite.unc.edu/cmcmag/1994/jul/moo.html>
- MicroMUSE Charter and Bylaws*. MIT. 1991
<ftp://musenet.bbn.com/micromuse/Charter.gz>
- Papert, Seymour. *Mindstorms: Children, Computers, and Powerful Ideas*. New York: Basic Books, 1980.
- . "Situating Constructionism." in *Constructionism*. eds. I. Harel and S. Papert. Norwood, NJ: Ablex Publishing, 1991
- Porter, David, et al. *NDDL 1994-95: Phase 2 Review*. Open Learning Agency, B.C. Ministry of Education. 1995.
http://www.ola.bc.ca/schools/nddl/About/Pubs/PDFs/rep94_95.pdf
- Reid, Elizabeth M. *Cultural Formations in Text-Based Virtual Realities*. 1994
<ftp://parcftp.xerox.com/pub/MOO/papers/CulturalFormations.txt>
- Rosenberg, Michael S. *Virtual Reality: Reflections of Life, Dreams, and Technology*. Northwestern University. 1992.
<ftp://parcftp.xerox.com/pub/MOO/papers/ethnography.txt>
- Serpentelli, Jill. *Conversational Structure and Personality Correlates of Electronic Communication*. 1992.
<ftp://parcftp.xerox.com/pub/MOO/papers/conv-structure.txt>
- Turkle, Sherry. *Life on the Screen*. New York: Simon & Schuster. 1995.
- Unsworth, John. *Living Inside the (Operating) System: Community in Virtual Reality (Draft)*. Institute for Advanced Technologies in the Humanities. University of Virginia. 1995
<http://jefferson.village.virginia.edu/pmc/Virtual.Community.html>
- Virtual Online University, Inc: An Overview*. 1996.
<http://www.athena.edu/overview.html>
- Young, Jeffrey R. *Textuality in Cyberspace: Muds and Written Experience*. 1994.
<http://sunsite.unc.edu/pub/academic/communications/papers/muds/Textuality-in-Cyberspace>