Collaborative Information Retrieval: Gopher from MOO

Larry Masinter [1], Erik Ostrom [2]

Abstract

There are two visions of how use of the global network will evolve in the future. First, individuals will use the network as a resource, providing access to material from libraries and other suppliers of information and entertainment. Second, in addition to communicating with these data sources, people will communicate with each other, using a variety of interactive text, audio, and video conferencing methods.

This paper is about a system that combines the two uses: adding an information retrieval tool (Gopher) to a 'text based virtual reality' environment (MOO). The combination allows informal collaboration using information retrieval to happen across the network.

I. Introduction

There are two visions of how use of the global network will evolve in the future. First, individuals will use the network as an information and entertainment resource, providing access to material from libraries and other suppliers of information and entertainment. Second, in addition to communicating with these data sources, people will communicate with each other, using a variety of interactive text, audio, and video conferencing methods.

Although these visions have usually been pursued separately, it is likely that applications will evolve that combine elements of both. The introduction of an information retrieval tool (Gopher) into a social network environment (MOO) allows exploration of some of the forms such a combination might take.

I.A. Electronic Libraries

One of the most commonly discussed metaphors for electronic information retrieval is that of the 'electronic library'. Most of the current conceptions of electronic libraries describe information repositories of online material and retrieval tools to help individual users find their way through this material.[20] In a comment on creating a national network of information superhighways, Vice President Gore claimed that one such application would be to allow a school child to "come home after class and ... plug into a digital library that has color – moving graphics that respond interactively to that child's curiosity."[7][19][17]

A typical description of the electronic library is that users will have access to the contents of a local library from home or work; larger sources, right up to the Library of Congress, could be consulted as needed. The primary technology elements proposed for funding in the development of electronic libraries are in the areas of input (scanning, character recognition), retrieval, and presentation. The entire technology emphasis is on collecting material and making it available to individuals.[12][14]

However, a library is more than just a pile of books. Libraries are also social spaces. Treating the 'electronic library of the future' as an information repository ignores many of the roles played by current institutions, where library users interact with their friends, colleagues, and professionals to finding material that is relevant for them.

In the distributed world of Internet information resources, where users at home or work interact with online library resources, where are the librarians? How can those who know lead those who do not? Only the most aggressive information seekers spend time scouring the Internet for interesting information. People most frequently rely on the most useful form of information retrieval: ask someone who knows.

Of course, some interactions of this type can be mediated by electronic mail and newsgroups. However, these asynchronous methods are not interactive: there's little opportunity to watch others interact with the system, to say, politely, that you don't understand, or to engage in the other kinds of behavior that go along with interactive learning.

The current mechanisms for distributed information in the Internet–World-Wide Web[3], WAIS[16], Gopher[1]–do not directly provide any mechanisms for collaboration. The system described in this paper, MOO-Gopher, shows how an information retrieval system designed for soli-

[1] Larry Masinter is at the Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA 94304, masinter@parc.xerox.com.

[2] Erik Ostrom is at Gustavus Adolphus College. He may be reached at eostrom@nic.gac.edu.

tary users can become a part of a social and collaborative environment.

I.B. Computer-based conferencing

Over the last few decades, a number of systems have been devised for allowing people to 'talk' to each other over computer systems. The earliest mechanisms were person-to-person communication systems, but even in the early '70s multi-person conferencing systems were being developed.[5] Out of this work evolved a variety of systems, including forums in CompuServe, bulletin-board systems, and, more recently, IRC and MUDs. These services allow multiple users to communicate with each other real-time.

MOO, the conferencing system for which we developed MOO-Gopher, is a MUD; that is, it's

> a software program that accepts 'connections' from multiple users across some kind of network (e.g., telephone lines or the Internet) and provides to each user access to a shared database of 'rooms', 'exits', and other objects. Each user browses and manipulates this database from 'inside' one of those rooms, seeing only those objects that are in the same room and moving from room to room mostly via the exits that connect them. A MUD, therefore, is a kind of virtual reality, an electronically-represented 'place' that users can visit.[9]

MUDs were originally developed as games; they grew out of attempts to build multi-player versions of text-based environments such as Dungeon, Adventure and Zork. (This part of their heritage is evident in the name–"MUD" was originally derived from "Multi-User Dungeon", although other "D" words have been suggested as replacements.) They are considered by some to be frivolous use of computer systems–some universities have taken to prohibiting use of MUDs as a recreational activity that wastes system resources.[18] However, the idea behind MUD–multi-user interaction in a simulated setting–is a powerful one with many applications.

A session with a MUD might begin like this (taken from JaysHouse, the MOO where MOO-Gopher was implemented):

```
% telnet jayshouse.ccs.neu.edu 1709
Welcome to JaysHouse!

Type:
 'connect <character-name> <password>'
           to connect to your character
 'connect Guest'
           to connect to a guest character
 '@who'    to see who's logged in right now

After you've logged in type:
 'help'    for documentation.

>connect guest
```

You connect as a guest on JaysHouse. When you first connect, you see a description of the "room" you are in.

```
Underground
This is a dark, cramped space.  It appears
 to be very crowded in here; you keep
 bumping into what feels like drainage
 pipes, alligators, and other people
 (apparently sleeping).  One useful thing
 that you've discovered in your bumbling
 about is a manhole cover above you.
Don't forget to take a look at the
 newspaper.  Type 'news' to see it.

>up
```

Simple "direction" commands move you through the MUD world.

```
Hwy. 169
A swath of blacktop scything through the
 forest to the west and farmland to the
 east.  Trucks rumble past, accelerated to
 dangerous speeds by wild-eyed drivers on
 dangerous amphetamines; cars scream past as
 well, most piloted by alcohol-mad college
 students driven over the edge by one too
 many calculus midterms. This is a dangerous
 place to be.  A small driveway dips into
 the woods to the west here, and you notice
 (strangely enough) a manhole cover in the
 middle of the highway.
```

MOO–for "MUD, Object-Oriented"–is a type of MUD that is particularly extensible, providing an object-oriented, dynamic language and a ready-made collection of useful objects. MOO includes its own scheduling mechanism, where programs can schedule asyncronous activities to occur at a later time. The MOO language also includes constructs for creating network connections, which can be used to provide access to external information sources; it features a novel 'permissions' system, which allows multiple users to develop elements of the simulation completely independently.[8] Conventional MUD objects like rooms, exits, and notes, are other utilities such as a 'help' system, tools for collaboration on programming projects, and a mail system are all implemented within the MOO language.

MOO, like all MUDs, is also a social environment. MUDs allow individuals to converse with each other. All the participants in the same 'room' hear what others in the room say. It is possible to 'whisper' to another user privately, or speak aloud. Coordination of communication happens using familiar metaphors; to talk to specific people, you go to the room where those people are.

In addition to conversational interactions in MOO, it is common for MOO users to construct things for use by others: a cuisinart in the kitchen, an operational game (Scrabble, MOOnopoly), a refrigerator for leaving notes for your friends. It is common for MOO programmers to extend the environment in personal, interesting, and unexpected ways.

II. Gopher in MOO

The implementation of Gopher in MOO is an example of merging computer-mediated conferencing and online information retrieval. It is an embedding of an information retrieval system (Gopher) in the setting of a computer-based communication system (MOO), that is extensible, flexible, and allows users and groups to tailor their own environment.

The initial motivation behind MOO-Gopher was to provide access, from MOO, to real world data outside the MOO environment. This would allow 'virtual reality' objects to interact with data that exists outside of the mud world, but under the programmable control of the inhabitants of the MOO world.

A sample 'session' in MOO-Gopher illustrates the nature of the interaction. In this transcript, you've just made your way to the library at JaysHouse.

```
Library
A small library. It is lit by several small
 lamps.  A big oak bookcase stands against a
 wall.
You see the slate dispenser, a Roget's
 Thesaurus, a treatise on why MOOs are not a
 waste, a note about MOO-client protocol, a
 dictionary, a Sample Help Text, a note
 called 'help movement', and the Gopher
 Slate here.
```

You arrive at the library. There are a lot of objects here.

```
Larry arrives.
Larry says, "Hello"
```

Larry comes into the room and greets you. You talk back.

```
>"hello

You say "hello"
```

You type in the command `"hello` *to communicate.*

```
Jay arrives.
Larry says, "welcome"

>look at gopher slate
```

The `look at` *command lets you examine objects.*

```
Gopher Slate
1. The U of I Weather Machine (menu)
2. Electronic Frontier Foundation archives(menu)
3. parcftp MOO anonymous FTP information(menu)
4. Gopher sites for biology (menu)
5. Movie reviews (menu)
6. Mankato State University (menu)
7. MSEN inc (menu)
```

The gopher slate looks like it has a familiar Gopher menu on it.

```
>"I came to try out gopher

You say, "I came to try out gopher
Larry says, "What did you want to look for?"

>"A favorite gopher for biologists is at
 ftp.bio.indiana.edu

You say, "A favorite gopher for biologists is
 at ftp.bio.indiana.edu"
Larry jumps to ftp.bio.indiana.edu socket 70
 on Gopher Slate.

>look at slate

Gopher Slate: [ftp.bio.indiana.edu] <jump>
1. About-IUBio-Archive  [ 7Dec92, 24kb]
2. About-New-Features (menu)
3. Flybase (menu)
4. Genbank-Sequences (menu)
5. IUBio-Software+Data (menu)
6. Molecular-Biology (menu)
7. Network-News (menu)
8. Other-Bio-Gophers-Etc (menu)
9. Other-Gophers-Etc (menu)
10. Species (menu)
11. This-Gopher (menu)
12. Tunnel to Gopher+ door of IUBio (menu)
```

You talk to Larry about some favorite places, and he points the slate at it.

```
Larry says, "everytime someone moves the
 gopher pointer, the slate's description
 changes"
Larry goes back up a level on Gopher Slate.
Larry picks '2. Electronic Frontier
 Foundation archives (menu)' on Gopher Slate.
Larry says, "so now, look at the slate again"

>look at slate

Gopher Slate: [gopher.eff.org] Electronic
 Frontier Foundation archives
1. About the Electronic Frontier Foundation's
 Gopher Service
2. About the Electronic Frontier Foundation
3. Electronic Frontier Foundation Files &
 Information (menu)
```

```
4. Computers & Academic Freedom mailing list
 archives & info (menu)
5. EFF-Austin Gopher Service (menu)
6. Other Gopher and Information Servers
 (menu)
Larry says, "if you want to choose an item,
 just say 'pick <number> on slate'"

>pick 2 on slate
```

Now choose a menu item.

```
You pick '2. About the Electronic Frontier
 Foundation' on Gopher Slate.
*******************************************
General Information about the Electronic
 Frontier Foundation
*******************************************

The Electronic Frontier Foundation (EFF) was
 founded in July, 1990, to assure freedom of
 expression in digital media, with a
 particular emphasis on applying the
 principles embodied in the Constitution and
 the Bill of Rights to computer-based
 communication.
-1 of 24-- 'next on Gopher Slate' for more -

>"I did it ahah

You say, "I did it ahah"
Larry claps
```

The "virtual" aspect of MOO allows the often daunting mass of available information to be embedded in more familiar metaphors: rooms, pieces of paper, a juke box which plays song lyrics. Instead of simply providing an additional set of commands for locating and retrieving information, or a gateway to a traditional client, MOO makes it possible to fit information services into the comfortable semi-realistic interface that the rest of the MOO provides.

The multi-user aspect of MOO has several significant implications. First, it makes it easy for people to work together. When using an information resource privately, the user has to be lucky enough to have a helpful person around, or may end up mailing out a plea for help to some mailing list or newsgroup. When using MOO, the user is fairly likely to find direct interaction with someone helpful. In addition, MOO provides a way for people who are simply after the same kind of information to work together to find it. This is of course simple enough for people physically located in the same room, but collaboration across a greater distance can be very difficult.

Second, the fact that MOO provides easy interaction between people, and that MOO information tools, programmed in MOO code, can be easily manipulated from within an interactive session, means bugs can be fixed and new features and tools added very quickly. It is usually difficult for the implementor of a software tool to observe, first hand, how the tool is being used. In MOO, this kind of interaction is frequent: by observing the errors and misunderstandings of novice users, the implementor can see the ways in which a user interface is confusing or awkward. It's often hard for an ordinary user of a piece of software to provide adequate information for the programmer to locate a bug. But in MOO sessions, the users can talk about problems, the developers can work with them, possibly adding debugging information to the code as it's being used, to figure out what exactly is going on. And often the problems can be located and fixed on the fly, in a matter of a few minutes.

II.A. Why Gopher?

Of the popular systems for network information retrieval, Gopher had three advantages for MOO implementation: there is a large supply of information available from Gopher servers across the internet; most information in Gopher servers is provided as plain text; and the protocol is very simple.

Retrieving a piece of information with Gopher requires a stateless transaction: the client opens a telnet connection to a given socket at the server, send a selector string, and collects the response: one connection, one (single-line) command, one (arbitrary-length) response. All Gopher interactions are based on this simple mechanism, whether doing a search, fetching text or an image, or getting back a directory in a hierarchy.

II.B. MOO implementation of Gopher

The Gopher interface in MOO can be divided into two parts. One part is a single object which provides raw programmatic access to arbitrary Gopher data; the other is a set of objects which provide user interfaces to Gopher data, or in some cases specific subsets of that data.

II.B.1. Getting at the data

The centerpiece of MOO-Gopher is an object which acts as the shared programmatic interface between all other MOO objects and the data accessible through Gopher, by any process or user.

In order to improve efficiency, the Gopher implementation does caching–it keeps track of recent requests, and the time they occurred, and just returns the 'old' data if it isn't too old. The cache is global.

This provides only a programmatic interface to Gopher–that is, it provides data in a form that can be passed around by MOO programs, but can't be easily used (or even seen) by ordinary users. In order to be useful, MOO-Gopher needed user interface objects–objects which provide the user with a view on some part of Gopher space. These have evolved over time, from our initial experiments with "Gopher rooms" to our current "Gopher slates" and other objects.

II.B.2. Gopher rooms

The first visible interface built was a 'Gopher room'. The idea was that the users of the MOO-Gopher system would be in a room. Anyone in the room could issue commands to manipulate the state of the Gopher interface. The choice of a 'room' was to allow multiple participants to share a common experience, and to make the manipulation of the Gopher interface as much like an conventional Gopher interface as possible. Many different Gopher rooms could operate simultaneously, so that different conversations could be held.

The Gopher Room worked very much like a traditional Gopher client in its command structure. As with most MOO things, state is permanent: if you leave the MOO and come back another day, the room will be in the same state you left it, unless someone else came in and changed it.

The Gopher room keeps track of a 'stack' of visited nodes. When you 'pick' a menu choice (or 'jump' to an unrelated Gopher node), the Gopher room's state changes to point at the new menu, but the old menu is kept on the stack. You can then 'pop' the stack to go back to where you picked from. Picking a text node was different–it would show the text to you (and no one else), but not change the state of the room.

In the initial design, everyone in the room saw all changes to the room, but there were some complaints–people didn't want to see the various state changes if they weren't interested, but did want to talk to those who did the retrievals. Also, anyone in the room could change the state at any time. This became a problem if two people were trying to work together–they could accidentally stumble over each other's commands. A simple interlock was added: you couldn't 'pick' a line unless you had seen the current menu.

II.B.3. Gopher notes

The idea behind Gopher notes was to make something that looked and acted like a regular MOO note–a piece of paper that could be carried around and that stored information that could be read–except that it got its data directly from Gopher. This was an attempt to allow 'permanent' objects on the MOO–as opposed to the rooms, which contained constantly changing information–that contained virtually (or meta-virtually) stored text. Notes had the advantage that they could be picked up, carried around, shown to people in other rooms, and essentially handled as semi-realistic objects–different from the "abstract" feel of the room.

II.B.4. Gopher lists

Many Gopher clients let the user make a list of "bookmarks" or things to remember, so you can easily get back to a place that you thought was interesting. The first attempt at providing some kind of memory in Gopher took the form of a list, like a note, that you could copy Gopher nodes onto from a room. A player could then carry that around, show it to people, or take it to a Gopher room and use it to guide a jump to a site referred to by the list.

II.B.5. Gopher slates

After a while, it became clear that the 'room' metaphor, while convenient in terms of command line syntax, was awkward to use. It was inconvenient to have to be in a special kind of room just to access Gopher. When you were in a Gopher room, it didn't feel like a room; it felt like a Gopher client with people in it. The Gopher room idea took advantage of the collaborative nature of MOO, but not its ability to present information to people in a familiar setting. Notes were a familiar kind of object, and you could use them in familiar sorts of ways (reading a note, as opposed to "picking" an item from a list); however, they were a lot less flexible, so they lost most of the advantage of actually having the full Gopher service behind them. You couldn't navigate your way around Gopher with a note; you could only put something on it and then have people read it.

This led to the development of a 'Gopher slate'. We wanted something that was portable, like the notes, but also allowed the user to move through Gopher space, like the rooms. The initial idea was a sort of chalkboard that you could carry around and show to people. What we ended up with was a laptop computer, since it's rare that you point at something on a chalkboard and the whole chalkboard erases itself and replaces the text with some different piece of text. The laptop was a straightforward translation of Gopher into MOO terms: You use a computer which has a Gopher client (not in so many words), and manipulate it in various ways. This wasn't as intuitive as the Gopher notes, again, but it carried out the function that the Gopher rooms had handled, while also being portable.

One problem with moving from rooms to slates was a loss of convenience. With many Gopher clients, you can get around by simple one-keystroke or point-and-click commands. Given the MOO environment, we couldn't easily do that. (The shortest commands possible are two keystrokes–one letter and the enter key.) In the Gopher rooms, though, we could still use fairly short commands, one or two words like "pop" or "pick 7". Once we went to the "slates" metaphor, because of the way the usual MOO parser works, we had to use commands like "pick 7 on slate", always specifying the object we wanted to work with. If there was more than one slate in the room, we would have to specify further: "pick 7 on jonny's slate". After a while we worked out a system where you could "work with Jonny's slate" and then the slate commands would be abbreviated to the form they had in rooms–"pick 7", and so on. The long forms were still available if you wanted to do something quickly with a slate without formally starting to work with it.

II.B.6. Gopher notebooks

The next general Gopher navigation tool we plan is a "notebook". The notebook will be a looseleaf binder filled with "pages" of various sizes, arranged so that each Gopher directory page is followed by a series of smaller pages, each of them an entry from the directory. Search items will take the form of "indexes", in which you'll be able to look something up and get a reference. Remembering items will be represented as writing them down in a "table of contents" at the beginning of the notebook. It should be possible to "copy" a page from a notebook, automatically creating a Gopher note, and to "insert" a note in another notebook, adding it to the notebook's table of contents.

We hope that the notebooks will bridge the gap between the flexibility of the room and slate, and the familiarity of the Gopher note. It remains to be seen whether we can make the "page size" representation clear enough to make this familiarity work out.

II.B.7. Other objects

Besides these general ways of navigating around Gopher space, programmers have been writing specific objects that use individual Gopher sites but provide a "realistic" interface to them, using a physical analogy for information space. For example, in JaysHouse, there is a "retargetable weather map" in the Living Room; imagine something like the one a television meteorologist uses. A user can turn the map to a given state, press buttons to get forecasts for specific cities, or set the type of information it provides (temperature, pressure, and so on). In this way, you can find out the weather without going through a series of menus. There's also a juke box which prints out the lyrics to the songs selected. Users can choose to listen casually or intently to the juke box, or just ignore it completely. (Users who listen casually tend to miss a lot, and of course users who ignore it miss everything.) Both of these objects illustrate the use of data available via Gopher to enrich the text-based world of the MOO.

II.B.8. Text forwarding: linking MOO, Gopher, and electronic mail

When you use MOO-Gopher to read a text, you're forced to use the MOO's limited internal text-reading facilities. MOO is set up for simple line-by-line communications; as such, it's not a great environment for reading a lengthy piece of text, possibly moving back and forth within the text.

Several users of the early MOO-Gopher objects said that they would like to have the things they found emailed to them. This would allow them to keep the texts around in a more permanent form; to more easily cut and paste, for example to quote in another text; and, of course, to use their regular tools just to read the text. We implemented this through a "mailme" command; users could ask the MOO to mail them notes, or selections from slates or rooms.

This text-forwarding facility comes in handy frequently; for example, when one of us was looking for the registration form for the Gopher '93 conference, and happened to be logged on to the MOO, he used MOO-Gopher to find the form, and had the MOO mail it to him, so he could print it out and send it in.

Text-forwarding is also useful for items that aren't meant to be read as flat ASCII. MOO currently can't deal with binary Gopher items; however, there are a number of formats that are composed of printable ASCII characters but aren't especially readable in that primitive form. TeX and Postscript are two examples. When a user finds a document in one of these formats, the simplest way to read it is to have it mailed, and then read it with a TeX or Postscript viewer locally.

When the Gopher object is pointed towards a directory, rather than a piece of text, the mailed form is somewhat different. A directory is mailed as a sequence of Gopher nodes, in the standard "Gopher bookmark" format–these can then be edited into a file, and some clients will understand these as a list of interesting Gopher items. This allows us to bridge a gap between MOO-Gopher and local Gopher clients.

A less clumsy way to bridge that gap is to use specialized clients. We worked out a protocol in which the MOO sends a string to the client, and the client invokes a local Gopher client to bring up the indicated item.

III. Users and communities

After the initial Gopher room was working, we advertised its existence on the comp.infosystems.gopher newsgroup. There was a fair amount of response from that announcement; in particular, one user logged on, looked around, and wrote a review that he posted on netnews[11]; this brought a wave of biologists to the MOO. A group of biologists is currently building an electronic meeting center, designed for virtual conferences but also for informal discussions.

Although Gopher was the initial attraction for the biologists, it isn't central to their plans for the conference center; the nature of MOO in general lends itself to the "virtual conference" idea, by being interactive and providing a somewhat familiar environment–rooms, people, tables and so on, as opposed to the more abstract forms of, say, Usenet, email, and IRC. However, Gopher provides them with some useful tools.

At a conference, it's possible for a speaker to show slides, or overhead projections. In the context of a MOO conference, this would take the form of information available via Gopher. The speaker would prepare ahead of time by making a "slide" that points to a given item in Gopher space; then, during the presentation, the act of "showing the slide" would cause a message to be printed to the audience's client programs, indicating that this item should be retrieved and displayed. Fortunately, many biological resources are already available via Gopher; so there's some degree of Gopher sophistication in the community.

In less formal settings, people use MOO-Gopher, not for showing prepared "slides", but for collaborative exploration of Gopher space, usually with the intent of solving a problem or finding the answer to a question. Many of our users have Gopher clients locally; so why do they use MOO-Gopher instead? Mostly because MOO offers a collaborative medium for information retrieval, as opposed to the isolated feel of other Gopher clients.

For example, recently a team of six MOOers at JaysHouse entered the Internet Scavenger Hunt. The Internet Scavenger Hunt is a monthly contest to find information that is publically available over the network. The team worked together, on MOO, to collaborate on finding the answers to the questions posed. They used Gopher slates extensively (although not exclusively) as a way to help each other.

People involved in that experience found that:

- MOO makes it easier to brainstorm about how to find a desired piece of information. The scavengers could pick up on each other's ideas and take them in different directions, as opposed to just searching independently.
- MOO, by being a semi-realistic environment, made it easier to communicate results to one another. It feels more comfortable to "show a note" to someone than to supply a Gopher address.

Of course, these advantages would hold for a group working together in physical space as well. The difference is that the MOO scavenger hunt team could work together while situated in offices and computer labs in such disparate locations as Boston, Palo Alto, and Oxford.

IV. Related Work

The Astro VR project was conceived as a way to allow astronomers to hold virtual conferences using MOO. Its creators believed that MUDs could be used to provide an intuitive interface in a closed "cyberspace".[10] In Astro-VR, a special 'client' program is used to decode commands sent by the moderator of a conference. Many systems (CompuServe, BBS systems) are discovering novel ways to combine interactive communication with information retrieval.[6]

V. Conclusions

The Gopher interface in MOO is evolving. The environment of MOO provides a rich, active environment for exploring a variety of user interfaces to information retrieval tools. Watching users attempt to explore current tools provides insight into how the tools might evolve.

Information retrieval without collaboration lacks an important element of how people interact with information. MOO-Gopher is an effective means of bringing information retrieval into a collaborative environment. Friends and colleagues use MOO-Gopher together, brainstorming ways to find needed resources.

In the online setting, information access tends to be far less formal than what happens in a physical library. Online information is used for gathering small bits of information (a word in a dictionary or encyclopedia, an article in a database), while library use tends to be structured more around formal research.

There is much more peer collaboration in online information access than there is in a physical library; those who maintain their site's information repositories fill the role of librarians, but currently do not interact as strongly with the user community.

Acknowledgments

Thanks to Michele Evard, Pavel Curtis, Kent Pitman, Marilyn Carter and Amy Bruckman for their comments on this paper. Thanks to Jay Carlson, who runs JaysHouse, and to Northeastern University, where it runs. JaysHouse is accessible from the Internet at jayshouse.ccs.neu.edu, port 1709.

Note: references available over the Internet are followed by a Uniform Resource Locator[2] pointing to the source of the reference.

References

[1] F. Anklesaria, M. McCahill, et al., "The Internet Gopher Protocol," RFC 1436, University of Minnesota, March, 1993.

`ftp://nic.ddn.mil//rfc/rfc1436.ps`

[2] T. Berners-Lee, "Uniform Resource Locators," working draft, CERN, 30 March 1993.

`ftp://info.cern.ch //pub/ietf/url4.ps`

[3] T. Berners-Lee, R. Cailliau, J-F. Groff, B. Pollermann, CERN, "World-Wide Web: The Information Universe", in *Electronic Networking: Research, Applications and Policy*, Vol. 2 No 1, Spring 1992, Meckler Publishing, Westport, CT, USA.

[4] S. Bhargava, "Hit the stacks–Without Leaving The Dorm," *Business Week*, p. 97, March 9, 1992.

[5] R. Bretz, J. Carlisle et al. "A Teleconference on TELECONFERENCING," Working Paper, ISI/WP-3, USC Information Sciences Institute, September 1976.

[6] S. Brinker, "Corporate bulletin board systems: customer support and more in the 1990s," *Telecommunications*, 25(11):33, November 1991.

[7] B. Clinton and A. Gore, "Remarks by the President and Vice President to Silicon Graphics Employees," Executive Office of the President, The White House, Office of the Press Secretary, February 22, 1993.

`gopher://wiretap.spies.com /0/Clinton/prez/930222.silicon`

[8] P. Curtis, "LambdaMOO Programmer's Manual," October, 1992.

`ftp://parcftp.xerox.com //pub/MOO/ProgrammersManual.ps`

[9] P. Curtis, "Mudding: Social Phenomena in Text-Based Virtual Realities," in the Proceedings of the 1992 Conference on the Directions and Implications of Advanced Computing, Berkeley, May 1992. Also available as Xerox PARC technical report CSL-92-4.

`ftp://parcftp.xerox.com //pub/MOO/papers/DIAC92.ps`.

[10] P. Curtis and D. Nichols. "MUDs Grow Up: Social Virtual Reality in the Real World," unpublished report, May 5, 1993.

`ftp://parcftp.xerox.com //pub/MOO/papers/MUDsGrowUp.txt`

[11] R. Harper, "For Such Times As These," *Bio-Naut Newsletter,* No 23, August 1, 1993.

`news:1993Feb25.122623.6473@nic.funet.fi` or `gopher://actlab.rtf.utexas.edu /0/MUD/Research/Notes/moo.gopher`

[12] J. Hartmanis and H. Lin, eds. "Computing the Future: a Broader Agenda for Computer Science and Engineering," National Academic Press, Washington, D.C. 1992.

[13] J. Markoff, "The staggering scope of the Internet: a thicket of networks wound 'round the globe," *Digital Media: A Seybold Report,*, vol. 1, issue N11, April 20, 1992.

[14] J. McCormick, "Library of Congress Wants to Charge for Electronic Records," *News-Byte newswire*, Clarinet News Service, ¡NB930226.22@clarinet.com¿, February 26, 1993.

`news:NB930226.22@clarinet.com`

[15] K. Savetz, "Astronomers to Chat in Virtual Reality–Unique software based on MOO," *LambdaMOOspaper,* Volume 1 Issue 6, June 21, 1992.

telnet: lambda.parc.xerox.com:8888, then connect guest and @read 6 on *paper.

[16] R. Stein. "Browsing through Terabytes– Wide-area information servers open a new frontier in personal and corporate information services," *Byte*, 16(5):157-164, May 1991.

[17] P. Trounstine, "Clinton's High-Tech Initiative – He Proposes Spending Billions To Keep U.S. on Cutting Edge," *San Jose Mercury News*, p. 1A, February 23, 1993.

[18] B. Verser, "Network utilization by MUD players (was Re: Gaming)," In *Computers And Academic Freedom Newsletter*, v.1, n.43, Dec 15, 1991.

gopher://gopher.eff.org /0/academic/news/cafv01n43

[19] "Technology for America's Economic Growth: A New Direction to Build Economic Strength (White Paper)," *White House Press Release*, February 23, 1993.

gopher://wiretap.spies.com /0/Clinton/prez/930222.tech.2

[20] Undated discussion on PACS-L Bitnet group (PACS-L@UHUPVM1.BITNET), topic "virtual libraries".

Author Information

Larry Masinter is at the Xerox Palo Alto Research Center, where he currently works on projects relating to document management and information access. He has worked on programming environments and development tools, and received the 1992 ACM Software System Award for his work on the Interlisp system. He received his B.A. from Rice University in 1970, and his Ph.D. in Computer Science from Stanford University in 1980. He is known as Larry on JaysHouse, and Grump on LambdaMOO.

Erik Ostrom is a student at Gustavus Adolphus College, where he is studying computer science and religion. He has worked on a variety of projects related to text-based simulated environments (primarily MOO) and hypertext information access (jtext and, more recently, Hyper-G). He is known as Joe Feedback or Erik on assorted MOOs.

Epilogue

The authors developed MOO-Gopher together (and wrote the first draft of this paper) without ever having met, communicating only via MOO and electronic mail.

```
You join Michele.
Michele says, "Hi!"
>:waves
Grump waves
JoeFeedback tried to just stay in one place.
>:made a shortened transcript, but it is
 too long still
Grump made a shortened transcript, but it is
 too long still
JoeFeedback is too!
JoeFeedback says, "Maybe not."
JoeFeedback says, "Send it to me."
Michele says, "Transcript?  Oh, right."
>:added an introduction
Grump added an introduction
>"at least, Mickey thought we should talk
 more about the second thing
You say, "at least, Mickey thought we should
 talk more about the second thing"
JoeFeedback says, "exporting?"
JoeFeedback hsm
JoeFeedback nods.
JoeFeedback says, "It's too bad we don't
 have more to say about it."
>"oh, we can make stuff up
Michele rereads 'combining bringing'
You say, "oh, we can make stuff up"
JoeFeedback says, "Actually, we have some
 stuff, but it's not clear in the beginning
 that we're going to talk about it."
JoeFeedback says, "Oh good."
>:changes combining bringing
Grump changes combining bringing
>:wonders if we should include a transcript
 of talking about the paper
Grump wonders if we should include a
 transcript of talking about the paper
>:grins
Grump grins
JoeFeedback thought that was a significant
 part of Mickey's comments--that we should
 say in the beginning why people should be
 interested.
JoeFeedback picks up bonker.
>"I tell people I'm working on a paper with
 someone that I've never met who goes by
 'Joe Feedback' and they don't believe me.
JoeFeedback bonks Grump with bonker....
 Grump says "Oif"
You say, "I tell people I'm working on a
 paper with someone that I've never met who
 goes by 'Joe Feedback' and they don't
 believe me."
Michele grins.
JoeFeedback laughs.
```