Browsing the WWW by interacting with a textual virtual environment -- A framework for experimenting with navigational metaphors

Andreas Dieberger
Georgia Institute of Technology
School of Literature, Communication, and Culture
Atlanta, GA 30332-0165
Tel.: (404) 894-2730
Fax.: (404) 853-0373
email: andreas.dieberger@lcc.gatech.edu

ABSTRACT
This paper describes a system that combines a textual virtual environment (MOO -- MUD Object Oriented) and a WWW browser. The MOO provides a text-only but information-rich spatial user interface in which objects and locations can be associated with pointers to WWW pages. When using a specialized MOO client, navigation in the MOO causes the corresponding Web pages to be loaded. The overall effect is the possibility to navigate the Web using spatial navigational metaphors. Textual virtual environments support the creation of diverse navigation tools and metaphors. The *Juggler* system we describe can thus serve as an experimental tool to explore diverse navigational metaphors for the WWW. The system uses references to Web pages which can be arranged in any possible way and allows users to overlay a new secondary structure on existing Web structures, even using Web pages not on one's own Web server. Textual virtual environments further support almost real time communication and interaction between several users. Because of the extensive interaction possibilities, the Juggler system can be used to discuss material on the Web, conduct guided tours through the Web or give presentations using material available on the Web.

KEYWORDS: WWW, navigation, spatial hypertext, metaphors, collaborative navigation

INTRODUCTION

Early in 1995 we considered how we could use a combination of WWW and a textual virtual environment as a teaching tool. The virtual environment would provide users with a virtual landscape where they could meet, communicate and interact even when physically separated. Course information, reading materials and students papers would be accessible via the WWW. The main problem was how to integrate these two concepts. If a user wanted to point out a Web page during an on-line discussion, he or she would have to *tell* the URL (Web address) of that document to other users. They would then retype this URL in their WWW browser to load it -- a tedious and error-prone process. The obvious solution was to automate this transfer through a specialized client program for the textual virtual environment. This client should recognize URLs in the output and cause the corresponding Web page to be loaded. By providing such a client we created an entirely unanticipated method to navigate the WWW -- Web-browsing by navigating a textual virtual environment.

The Internet and especially the World-Wide Web (WWW), appear to most users as a highly unstructured mass of information. Users have to cope with more or less badly designed pages, little developed or no navigational tools and so forth. While hypertext systems in general sometimes experience problems with how to convey the structure of the information to the user, the WWW often appears to have no structure at all.

This lack of visible structure on the Web sometimes is considered an advantage -- users do not *have* to know where information comes from. However structure is an important element in effective hypertext navigation. We think, therefore, that the hidden structure has more drawbacks than advantages. One way to show structure in a hypertext is to visualize the structure graphically. Another approach is to create an environment that shows structure in the way it permits navigation and in the metaphors it is based upon. This second approach is what we have tried to realize in our *Juggler* system.

In the next section we introduce textual virtual environments using the example of MOO systems (MUD Object Oriented). These systems are networked textual environments that can be used to create spatial environments. We point out how navigation in MOO systems is based mainly on metaphors.

We continue by describing the implementation of the MOO/WWW connection in the *Juggler* system and pointing out related work. In this section we also describe the illusion of navigating one coherent information space with two views. We then explain how we use *Juggler* to define new structures using existing WWW pages and what navigational metaphors we experimented with. We outline a few ways how users can interact and communicate using the system and a few applications. This section further describes additional features that shall simplify using the MOO for novice users.

The last section describes the major problems of our solution, gives a brief overview of possible and planned future work, and summarizes the paper.

TEXTUAL VIRTUAL ENVIRONMENTS

Textual virtual environments evolved out of text-adventure games. In these games the player character (a puppet representing the user) moves through an imagined landscape described solely by text, solves puzzles and collects treasures. While the original text adventure games supported only one player, current games often are networked and many players use the environment at the same time. They can meet in the game, communicate and interact with each other.

These game systems are called MUDs (Multi-User-Dungeons) or MOOs (MUD Object Oriented)[1] . These acronyms actually refer to the server software the games run on. In this paper we use MOO as a general term for multi-user textual virtual environments.

A MOO system without the game-related code is a distributed multi-user virtual environment described only by text. Such a system can be used for (almost) real-time communication and interaction between users. MOO environments easily cope with a larger number of users and they commonly develop into *virtual communities* [5] and *social places* [10], in which social behavior like in the *real world* is observable.

The MOO server software was developed by Pavel Curtis at Xerox Parc, enhancing earlier software created by Stephen F. White. MOOs are probably the textual virtual environment most extensively used for non-gaming purposes today. Recently there is rapidly growing interest in MOO systems as the basis for distributed conference facilities [11] and multi-user spatial user interfaces [21].

Using a Textual Virtual Environment

In a textual virtual environment locations, objects, users, and their interactions are not shown graphically, but described by text [4, 5, 10]. The user controls a *player character* by issuing commands in a more or less natural command language. Commands (for example "look at book") are executed by the MOO server and their outcome is described as if the player character had actively done something. According to the type of command this outcome may be visible only to the user who issued the command or to all other users in the same MOO room. In this example the player issuing the command may see a description of the book, whereas other players might see "X examines the book." Although this seems to be a step back to old-fashioned command-line interfaces, MOOs can provide a rich and detailed virtual world using this line-based interface.

Player characters are located in *rooms* with other objects and other player characters. The concept of the MOO room is a metaphor for *location* or *mode* where room *exits* act as links between these locations. Because of the similarity to the hypertext node and link model, MOOs are sometimes considered a special case of hypertext.

Navigational command like "go north" or simply "north" move the player character through the north exit of a room to the next room. Most rooms provide rectangular exits like up, down, north, west, east, or south but exits like southwest occur as well.

MOO users communicate using the "say" command. A line of text *said* is displayed to other players located in the same MOO room as if the player character had spoken. Other forms of communication exist as well, like the "emote" command, which provides emotions or gestures. Players can also interact by giving objects to each other, or by directly manipulating objects.

The following extract of a MOO session shows how users can interact with objects and other users. The user input is shown in bold letters. Text in italics describes additional activity.

```
The Entrance to the Residential Areas
This is the first room of the residential area.
To the [west] you see the obelisk at the center.
Behind the [19] door is Juggler's [living] room.
To the [south] is Donegal's area. The
residential areas continue to the [east]. There
is a bulletin board on the wall. It is a new
bulletin board with no postings.
(URL information)
The exits [19], [south], and [west] seem to be
used above average.
> w
The Center
This is the center of the TechMOO world. It
consists of a large plaza with a tall obelisk.
It's tip gleams golden in the sunlight and
probably can be seen from far off. Near the
obelisk a winding staircase leads [down] to the
[lounge], and to the [east] is the entrance to
```

[1] There exist several other game systems, for example MUSH, MUSE, etc. These other systems are not mentioned for the sake of simplicity. Several of these systems were developed earlier than the MOO system we use.

```
the residential area. Major streets lead towards
the [19], [west], and [south]. You see a
teleporter to the Virtual Skiles here [skiles].
(URL information)
The exit [east] seems to be used quite often.
> skiles
The room around you starts to shimmer and is
slowly replaced by another room. You teleport
out to the Virtual Skiles...

The Entrance of the Virtual Skiles
You are in an outdoor hallway of the Virtual
Skiles building, overlooking a nice courtyard
with a few trees. To the south you see a
stairway that leads nowhere right now and to the
north you see the Georgia Tech Library building.
[West] of here are glass doors leading to the
main hallway of the Virtual Skiles.
(URL information)
The exit [west] seems to be used quite often.

Merlin teleports in from the Center.
Merlin says, "Hi there - I was looking for you."
Merlin smiles.
> emote shakes Merlin's hand
You shake Merlin's hand
Merlin gives you a book.
> look at book
Its an thick leather-bound volume. The title
says "Collected bugs of TechMOO", compiled by
Merlin.
> read book
(Abstract of book is shown)
(URL information)
> give book to Merlin
You give a book to Merlin.
> say nice collection...
You say, "nice collection..."
```

The log of this session shows *(URL information)* in several places. When using the *Juggler* client this information is invisible -- instead the corresponding Web pages are displayed by the Web browser.

Navigation in MOOs -- What is *spatial*

Navigation in a MOO involves moving the player character in the imagined environment. Most MOO systems describe an environment modeled after a real space consisting of rooms, buildings, and streets. There are exceptions though, like parts of the MediaMOO system [2] [3] or the Hypertext Hotel MOO [3].

MOO locations commonly are arranged in a more or less regular grid along North-South and East-West. In interviews with frequent users of a MOO like system [6] we found that they talked about the system in a very spatial language -- as if these environments were real places they had visited.

Other studies found that people form very accurate mental representations of environments they read only descriptions of. Tversky observed in [22]: "From only studying the descriptions, students were able to produce maps that were nearly error-free, indicating that language alone was sufficient to accurately convey coarse spatial relations." (see also [20]) These observations justify to talk of MOO environments as spatial environments -- at least in a general sense. Note that this observation is valid for the 'typical' MOO, but that it is possible to design MOO systems that exhibit no spatiality or that use a non-Euclidean concept of space. In the remainder of the paper we assume a MOO using a (mainly Euclidean) space.

MOO rooms do not have a *location* and *extension*, and exits do not really have a *direction*. The position and extension of MOO rooms is inferred from the description, something that is not possible in normal hypertext unless the hypertext contains an environmental description like a MOO does. While most MOO areas are described close to real world spaces, one can define exits that *teleport* a user to the *other side* of a virtual city in one step, while another set of rooms may require the user to walk north five times to leave a room. Note that also the concept *other side of the city* is based on spatial relationships that are inferred from the description of the environment.

Also it is possible to define exits so that walking north twice brings the user back to where she came from. Because of the consistent space concept of most MOOs such disruptions are easily noticed. Most MOO areas use spaces that are intuitively understandable.

Navigation in MOOs is metaphorical: Besides *walking* in the MOO using *directional exits* like "west" or "north" there are other navigational possibilities. Essentially they all have the same function: they move a player character from one room to another and describe what happened both to the MOO player and to all other players present. Navigation in the MOO has to be understood metaphorically.

MOO exits are sometimes named after a location or an implicit direction (for example "shop", "out" or "climb rope"). We call these exits *non-directional* because of the lack of an explicit direction. This distinction is useful because these exits are perceived as something rare and special by most users. Both directional and non-directional exits typically are associated with short distances.

A third type of exit is called *special exit*. These exits give no indication of a direction and it is often impossible to infer a direction and distance. According to how they are described they are often perceived as a *magic feature*. These special exits (for example the exit "skiles" in the log example) work like a hypertext link in the spatial environment. Most users are quite flexible in coping with such unusual spatial structures [6, 21]. We believe the main reason for this flexibility is that the environment provides an overall consistent spatial framework for navigation so that these *special exits* are seen as additional helpful features. Another interpretation is that places reachable through special exits are perceived as *conceptual places* and not as part of the same spatial environment. This

[2] MediaMOO is at purple-crayon.media.mit.edu port 8888.
[3] The Hypertext Hotel MOO is at duke.cs.brown.edu port 8888.

interpretation is close to seeing MOO rooms as *modes* and not as *locations*.

Technically an exit "north" does not differ from a teleporter like "skiles". The transition is understood differently according to the description of the navigation process, that is according to how this action is *enacted* [1, 14]. In the log example above you see that the transition through the "skiles" teleporter is described differently both for the teleported as for a bystander. Another example for enactment is in the elevator log further below. Normal *walking* is not described in our system except in a few special places. The direction and distance inferred from room descriptions and the enactment of the transition make it possible to create a large set of possible navigational features. A fitting enactment for a navigation tool therefore makes a big difference for the MOO user in how he or she perceives the working of the tool.

Spatial Navigation: The three types of exits to be found in MOOs allow us to create several possible topologies. One possibility is to create MOO structures that adhere to our knowledge of everyday (Euclidean) space as much as possible. These systems certainly use the spatial metaphor most convincingly. When talking about a *spatial environment* we essentially mean a *mainly Euclidean environment*. If you adhere to everyday space you have to accept its inherent disadvantages; it is impossible to travel from a location A to a location B without visiting locations *between* A and B, for example.

A looser space conception therefore makes navigational features possible that break with some of our expectations of spatial relationships. To use the example above, such a system can provide direct 'teleporting'. Still such a system should be considered as based on a spatial metaphor as long the overall system adheres to 'normal' spatial expectations. Essentially a spatial environment that allows teleportation between a small set of locations *in addition to spatial navigation* actually uses two parallel spaces for navigation -- a conventional (Euclidean) one and a topological one to create additional paths through the environment. The spatial framework the user is acquainted with provides a structure that allows him or her to navigate as usual with the additional advantage of the extended navigation features.

A third kind of system may use mainly structures that are not in accordance with Euclidean space. Such a system probably should not be called *spatial* as its main focus is on non-spatiality in the everyday sense of space. Note, however, that in a mathematical (topological) sense of space such a system certainly is spatial as well.

The systems we are mainly interested in are systems of the second kind: they adhere to everyday space enough to provide users with a comfortable reference system for navigation, but provide additional navigational features to make navigation more efficient. The challenge in designing such a system is to use appropriate metaphors for non-spatial navigation so that it is easily understood and can be used effectively (see [7]).

In [13] it was advocated that "we will need to re-think out conception of space in hypermedia, and by extension, the dominant metaphor of *navigation*, that we use to describe transactions within it." The possibilities for creating navigational metaphors and mixed topologies make MOO systems promising tools for experimenting with different metaphors and space concepts. By combining the MOO with the WWW these possibilities extend to experimenting with navigational metaphors and space concepts for the WWW.

IMPLEMENTATION

A MOO client program basically provides a telnet-type connection to a server running the MOO software. For convenience MOO clients separate the user input and the MOO output into two fields. Our client (see figure 1) -- the prototype is implemented in HyperCard 2.2. -- essentially provides such a telnet connection. In addition it scans the MOO output for URLs [4]. Whenever a URL is found the client uses AppleScript to send it to a WWW browser to load the corresponding Web page.

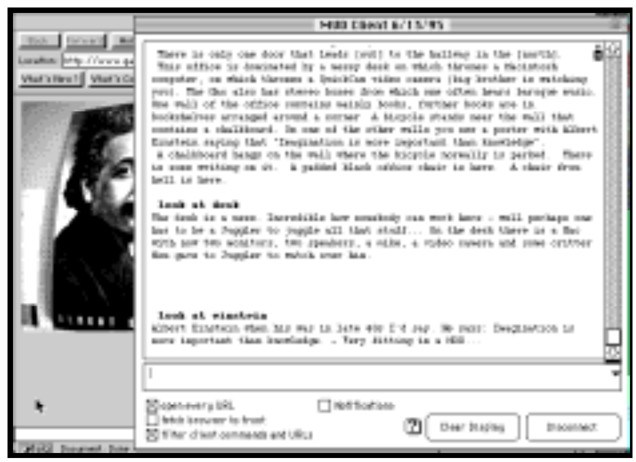

Figure 1: Juggler with the WWW client running in the background

In addition to this basic functionality the client provides local edit windows, lists of MOO commands and so forth. *Juggler* is not the only system that enhances a basically textual MOO. Instead the evolution of MOOs definitely moves towards systems with so-called *fancy clients*. Most of these graphical MOO systems are more complicated than our system. So far *Juggler* does not require modifications in the server software and it even can work with any type of textual virtual environment. The reason for this is that *Juggler* scans the MOO output for URLs and does not rely on a specific protocol to recognize URLs. This generality does not extend to some of the more recent additions, like

---

[4] MOO systems support a virtual communication channel between MOO and client, so-called out-of-band communication. Use of this second channel brings a few advantages while restricting other possibilities of the MOO/WWW combination. Presently *Juggler* uses out-of-band communication only to support local edit windows.

local edit windows, that require the MOO to understand a data transfer protocol (MCP -- MOO Client Protocol).

Juggler also allows users to click on room exits in the MOO output and generates an appropriate walk command. This feature expects room exits to be marked in a certain way and certain commands to be understood by the MOO. These additions can be easily implemented in most systems without modifications of the server software.

Through URLs *Juggler* can refer to whatever information is accessible for the WWW browser. This includes not only Web pages but also gopher- or FTP-documents, telnet connections to other MOOs, sound or video files and even VRML models. Using VRML three-dimensional models of rooms can be called up by the MOO -- these models can contain objects with links like ordinary HTML pages. We did not experiment with VRML till now because the focus of our work is elsewhere.

By modifying the MOO or supplementing it with CGI-scripts it is possible to create a connection from the WWW back to the MOO. This would allow for navigating the MOO by clicking on maps in WWW pages. This back connection has not (yet) been realized in our system.

One Information Space with Two Views

In our MOO (TechMOO) we associate URLs with object, locations and (sometimes) activities. Looking at an object or entering a room produces its description and the URL along with it. The client detects the URL, filters it from the output and causes the WWW client to load the corresponding Web page. Movement in the MOO-space therefore immediately results in navigation in the Web-space. This creates a impression of moving through *one* information space on which the user has two different views[5]. This illusion is particularly strong when MOO objects and associated Web pages refer to each other. This double information also proved to be especially useful when the connection to the Web browser is temporarily disabled or slow. This again supports the illusion of seeing two windows on the same information space.

The two views (textual and WWW) originate in our implementation using two separate applications. For simplicity we did not try to create the system as one integrated application, which would require implementing a WWW client. We consider the two separated views more an advantage than a problem. Because of the one-directional connection from MOO to WWW the user can switch to the WWW client and navigate in the traditional WWW style. While doing so he or she stays stationary in the MOO space. Should the user get disoriented in the WWW space he or she switches back to the MOO client and causes the initial page to be loaded again by looking at the room description. This provides a simple jump-back-to-a-landmark mechanism for the WWW.

[5] This association of WWW pages to MOO objects is similar to a *mnemonic space* as described in [23].

Related Work

To our knowledge there is no other system that tries to organize URLs in a MOO to support Web navigation like *Juggler* does. However there are systems that connect a textual virtual environment with the WWW. Most of these systems use a Web client to navigate an (often graphically enhanced) MOO-like system or they use specialized client/server combinations.

At the JHM system a MOO object has been developed that serves HTML files directly from the MOO[6] (this server object requires a minor server patch). JHM uses an internal hypertext-format, called jtext, which can be read both inside the MOO and -- through this HTML-server object, that converts jtext to HTML -- on the WWW. The MOO thus acts both as a MOO and as a WWW server. Further related work can be seen in MOOs providing Internet services like gopher [16] or news.

As MOOs and the WWW become more popular many systems appear on the Net that are accessible through the WWW (for example The Sprawl[7]) and even provide additional communication channels like Jupiter [5] or the Sesame system[8] by using specialized server and client software. Recently there is also a lot of activity to combine chat rooms with VRML based graphics or to create other new communication systems on the Net[9].

Other related work can be seen in systems that try to organize collections of URLs to support Web navigation. One example is the VIKI system [15] which can be used to organize Web references. In a certain sense all systems to visualize WWW structures (see for example [18]) and work on novel navigational metaphors can be considered related to this work.

ENVIRONMENTS THAT HELP IN NAVIGATION

What are the advantages of the MOO/WWW combination over using a Web client alone? One way is to view Juggler as a spatialized hotlist that also allows user interaction. The system can be also used as a tool to design paths through the Web. As MOOs support interacting with other users, like guiding them around the environment, this makes it possible to conduct guided tours through the WWW.

Defining New Structures on the Web

The spatial arrangement of URLs in the MOO environment defines a new structure on the pages referred to by these URLs. These structures necessarily are only small subsets of the Web. Representing even larger parts of the Web is

[6] JHM (or JaysHouseMOO) is at jhm.ccs.neu.edu port 1709.

[7] The Sprawl uses a combination of MOO and WWW called WOO. Accessing room MOO room descriptions from a WWW client without provisions to constantly update these pages eliminates the communication and interaction facilities of the MOO. For more information on The Sprawl see http://sensemedia.net/sprawl.

[8] See http://www.ubique.com/products/sesame_ds.html.

[9] An example is the WorldsChat system, see http://www.worlds.net/products/wchat/index.html.

impracticable as the Web changes too rapidly. However when referencing relatively stable Web pages in the MOO environment, these pages represent landmarks nodes in the Web, similar to pointers in a browser hotlist or a bookmark file. The spatial structure of the MOO environment then provides a new logical structure for these pages and the spatial navigation (hopefully) makes it easy to re-find them in the MOO.

Figure 2 visualizes this concept. Imagine a set of WWW pages with whatever linking structure. In the MOO we can create a hallway -- linearly connected rooms -- that define a path through these pages. By walking though this hallway we navigate the WWW pages in the sequence defined by the hallway -- ignoring the original linking structure on the Web.

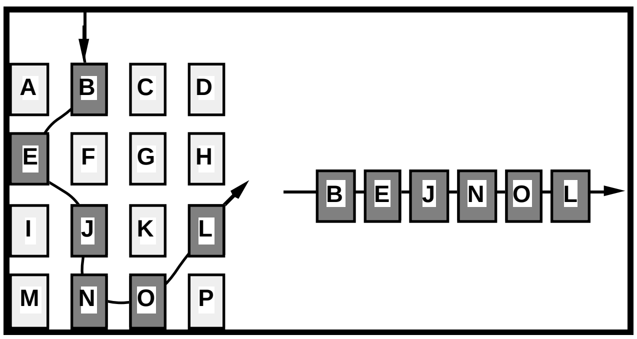

Figure 2: Defining new paths through existing structures on the Web

Structures need not consist of closely related documents as figure 2 suggests. Instead they can use pages distributed all over the Web, thereby creating new collections of hyper documents (see figure 3).

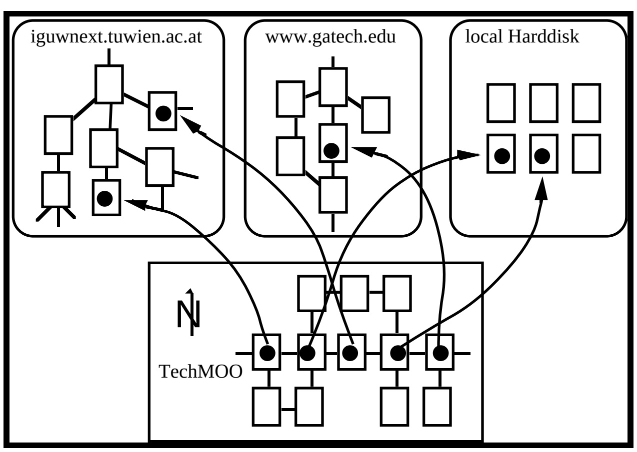

Figure 3: Creating a new path through existing structures on the Web.

Since URLs can access also local files these structures may include local pages as well. A walk through such a MOO structure then presents a sequence of pages (containing both proprietary material and material on the Web) to the user.

By creating such new structures the MOO 'builder' designs a reduced version of the Web. A hallway containing references to pages with a certain type of information functions like a Web page containing pointers to these Web pages. In this pure example the spatialization itself does not provide any real advantage. It does have its advantages, however, when such a new structure is combined with new navigational metaphors.

One may argue that such 'reuse' of web material out of context is not legitimate, but on a small scale this process happens all over the WWW already. With stable and reliable servers it is irrelevant if a Web document consists of text, pictures, and icons from the same server. Many WWW authors therefore reference icons or pictures in clip-art collections on the WWW.

Examples of Navigational Metaphors

A MOO does not *have* to be used in a spatial way. Instead MOO objects themselves can yield various URLs depending on how users interact with them. A MOO-bookshelf may have a Web page associated with its description (showing a list of documents on the shelf), it can produce URLs leading to document abstracts by looking at the documents and yet other URLs when reading the documents. Such a bookshelf therefore can act as a hotlist object by itself, without the typical MOO navigation involved.

While the bookshelf may return one URL for each command, objects can deliver a series of URLs over time. Being in the same MOO room with such an object results in an automated presentation of a series of Web documents. The objects also can deliver additional information on the Web pages. Fitting metaphors for such objects are a virtual person giving a lecture, or a slide projector. We have such a presentation object in TechMOO. It allows people to easily create presentations that give 'lectures' in the style of a multimedia slide show with the text being presented in the MOO and the graphics or other media presented in the WWW client.

```
> start presentation
Merlin starts the presentation.
The presentation says: "Welcome to Juggler's
Ego-trip through his own Web pages."
(delay)
The presentation says: "You may have know that
Juggler has a home page on the Web, but did you
ever bother to visit it? Well if you didn't its
time you did - here it is..."
(URL info)
(delay)
The presentation says: "This page shows you a
list of Juggler's publications."
(URL info)
(delay)
(...)
```

Such objects do not exploit the spatiality of the MOO. A corresponding spatial metaphor is a tour bus, that works like the presentation, but actually guides the user through the MOO. As entering a room triggers displaying of the room description and therefore the associated URL the bus again presents a series of URLs. Contrary to the presentation object users can walk the bus route on their own after having done the tour. Such a tour bus can be realized in two ways. It either can jump from room to room, ignoring the structure of the MOO exits. As this ignores the MOO structure it gives the user no chance to learn the layout of the (spatial) environment. A better tour bus therefore will navigate the structure defined by room exits (like a real bus has to use a road) and show the path to locations containing interesting information.

As the MOO lends itself to create environments that are similar to everyday spaces, obvious candidates for large-scale structures are building and city structures [6, 9]. MOOs traditionally often use village and building structures. This choice of structures is not iron cast, however, as the Virtual Internet in MediaMOO and the whole Hypertext Hotel MOO show. One structure we think is promising for spatialized hotlist in this scheme is the metaphor of a shopping mall. The mall consists of several levels, that are connected by an elevator. Here is an example of such an elevator -- note how the enactment is used to make the metaphor more believable.

```
> push button
You call the elevator.
The URL Mall Elevator appears and opens its
doors.
> enter elevator
You step into the elevator cabin.
The URL Mall Elevator
You stand in the ultramodern elevator leading to
and from the URL Mall. Through the glass walls
you can look outside. There are 4 buttons for
the levels this elevator can reach:
   3...3rd Level - <Under Construction>
   2...2nd Level - Hypertext, WWWW
   1...1st Level - User Interfaces and
Navigation
   0...The Entrance of the WWW Library (Exit)
Use the elevator with {press <num>}. You can
leave the elevator using [out].
> press 1
Merlin presses 1.
The elevator beeps and closes its doors. Then it
starts to move upwards.
Outside URL Mall Elevator, you see:

URL Mall - 1st Level
You are in the 1st Level of the URL Mall
```

Each level contains *departments* which contain URLs related to one topic. The interior of departments can be arranged either using sub-departments, or using hallways. These hallways ideally would be built such that more detailed information is provided the farther one navigates into the department [8]. Inside the departments non-spatialized objects like presentations and bookshelves are placed.

Architectural metaphors lend themselves to holding related information together. On a higher level buildings, districts and cities can be constructed to form more complex information structures connected by diverse transportation metaphors (see for example [9].) Such structures are mainly hierarchical, and it is important to supplement them with special exits that provide fast movement to other places. One useful metaphor to supplement a city structure with less restricted navigation may be a subway system, which connects landmarks in a city without being influenced by the structures between these landmarks. Users should be able to *summon* the subway wherever they are located so that they do not have to find a subway station to *enter* the subway system.

These metaphors are far from being a complete list of possibilities. A large variety of every day objects and spaces can be created in in the virtual environment by simply describing them appropriately.

Using Meta Information

Real environments provide people with many navigational cues that are often missing in synthetic environments. In TechMOO we try to use room descriptions that create the impression of an information-rich environment that makes navigation easier [6]. Room exits are pointed out using square brackets and are embedded in the room description. This is not necessarily the optimal solution but motivates users to read the entire room description, which gives them a better understanding of the environment.

Read Wear in TechMOO: For navigational cues we use the *read wear* concept, first described in [12]. Read wear uses information access statistics to point out objects that are used especially often. As navigation data in a MOO is averaged over a user population, read wear points out navigational trends.

We presently use two types of read wear. Room exits count how often they have been used. Room description are automatically modified on the basis of this data to point out exits that are used especially often, for example:

```
The exit [east] seems to be used quite often.
```

These descriptions hint at navigation patterns of the user population. These patterns (perceived as often traveled paths) help users to orient themselves and also make it easier to locate rooms of general interest.

The second type of read wear occurs on bulletin boards. Postings on these boards remember their usage and their descriptions change accordingly from 'brand-new' to 'new', 'recent', etc. In this case the read wear points out both the number of accesses as the age. We use a post-it metaphor on the boards and postings drop to the floor after some time. If somebody thinks a particular posting is important he or she can re-post it to the board where it will stay again for some time. Postings that are not re-posted are eventually deleted.

Like any other object in the combined MOO/WWW system these bulletin boards can be used to distribute URLs. Postings containing URLs therefore keep track of read wear for this particular WWW page, a concept that is difficult to realize on the WWW alone. The read wear of a posting shows the accesses through this particular posting only. This has the advantage that the read wear mirrors the interest of a small group of WWW users -- namely those who access the page through this MOO system.

Interaction with other Users

One of the strengths of MOO based systems is the user interaction. As in real environments navigation in MOOs often is a collaborative task [6]. People ask for the way, give route descriptions, point out landmarks, or describe locations. MOOs support this type of interaction as users can not only talk to each other, but interact directly, for example by guiding each other. The quite detailed room descriptions further make it possible to give directions vaguely (like "go this direction till you encounter a red building"), a method of navigation the WWW itself hardly supports.

The easiest way to point out a Web page in Juggler is to simply *say* its URL. This allows users to give presentations like the presentation object does. To further support this type of interaction *Juggler* provides a 'URL palette (see figure 4). Clicking the upper button on this palette fetches the URL of the Web page currently displayed in Netscape and displays it in the MOO along with some additional text. (Essentially the client generates an emote command describing that a user points out a Web page). The palette further contains a pop-up list of URLs. The client can import any local HTML file (for example a bookmark file) and places the URLs it finds into the pop-up list. Using the URL window it is very easy to discuss material on the WWW and to point out pages one finds while browsing.

Figure 4: The URL Palette

More on Applications

The major use of the *Juggler* system is in managing collections of WWW pointers in a spatial way and in communicating with other users about them. This makes several applications possible. One might be to organize landmark URLs in the MOO. The spatial navigation in the MOO makes it easier to re-find these landmarks in the context defined by the MOO or to informally describe a path there for other MOO users. From these WWW pages users can navigate using conventional Web navigation in the Web browser.

The MOO essentially defines *paths* through the Web pages referenced in the system. These paths can be used to guide users to these Web pages, or to teach them the environment. According to Zellweger "Users are less likely to feel disoriented or lost when they are following a pre-defined path rather than browsing freely, and the cognitive overhead is reduced because the path either makes or narrows their choices." [24]. Such path structures can be easily defined in the MOO and could provide navigational infrastructure on the Web where it is badly needed.

We also see a lot of potential in the communication and interaction possibilities of the system. Guided tours, automated presentations, pointing out pages and discussing them on-line are the application for which we see most value for educational uses. People can give presentations by pointing out Web pages like overhead slides and can discuss the material on these pages immediately. However *Juggler* is not designed as a CSCW environment, even when some collaborative work is possible in this system.

As the pages referred to in the MOO do not need to be pages on the WWW, graphic or video files can be retrieved from a CD-ROM previously distributed. We imagine applications where a company provides maintenance information on CD-ROM and additional technical support using the MOO. Problems can be discussed on-line while a specialist points out information pages on the CD-ROM.

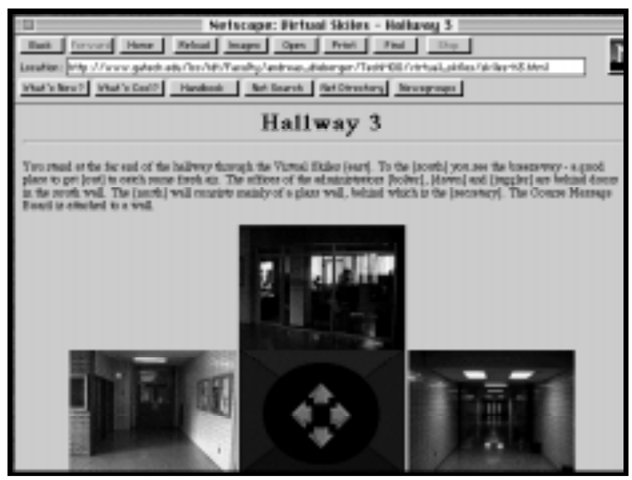

Figure 5: A hallway in the "Virtual Skiles"

Another application is to recreate an existing space in the MOO. Such an environment provides access to information about real locations. We created a part of our department in the MOO using Web pages containing photos of the rooms (see Figure 5). In such an environment, person-related information can be placed into corresponding MOO offices.

In this case spatial navigation to somebody's office retrieves information about this person (like a home page). Knowing the location of an office in the real environment allows users to locate the homepage of the office's owner without knowing its URL. The offices can contain additional information -- like brochures or publications. Similarly it is possible to create message boards containing information of general interest. Such virtual mirror spaces can also be coupled with sensors in the real environment to update information in the MOO (for example WebCams) [10].

Making MOOs easier to use

In Spring 96 we will used an improved version of the Juggler system to host a virtual conference on Romanticism at Emory University. One major problem is that most participants will be first time MOO users. MOO systems are not exactly the most easy-to-use type of software with their command line interfaces and hundreds of possible commands. A major focus of current development in the *Juggler* client is in making the system somewhat easier to use. These additions tighten the coupling of the MOO and the client -- many of these extensions will not work correctly with non-MOO systems without additional programming.

One of the simplest improvements is the URL Palette described above. Another new feature is local edit windows. An interesting application of these local editors is to copy and paste text into a text object in the MOO. Using the HTML server object this text can be served as an (unformatted) Web page through the MOO, without writing any HTML.

Another addition is a palette with the most frequently used MOO commands. Instead of remembering the exact command, users click on the command in the palette. Commands that contain templates (text in angle brackets like '<player>') trigger a dialog that asks users to complete the command with the text in angle brackets preselected (see figure 6).

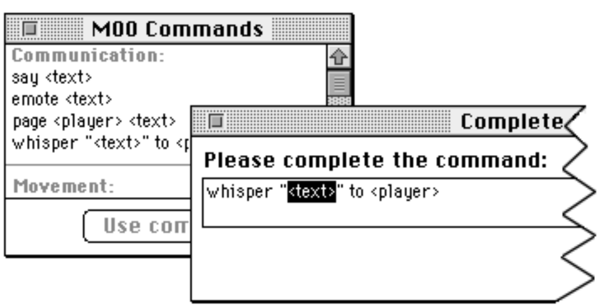

Figure 6: The MOO Commands palette.

Another extension is the possibility to navigate the MOO by clicking into the output field. When clicking on a text like [19] the client recognizes this as an exit and executes the command "go north". Similarly when the user clicks text enclosed in curly braces this text is interpreted as a command template as if it had been clicked in the commands palette. Using this feature MOO objects and locations can contain unusual commands in their descriptions (for an example see the log of the elevator above.)

[10] Such enhancements of mirror spaces have been discussed at Xerox Parc but to our knowledge they have not been realized and published.

UNSOLVED PROBLEMS AND FUTURE WORK

The main drawback of our system is that there are no tools to support creation of MOO structures. Creating, linking, and describing rooms is still done by hand. We are thinking about creating a simple room editor that supports importing of URLs and linking rooms. An improved room editor could be based on an existing hypertext system like VIKI [15] or Storyspace [2], and the structures created would be translated into MOO structures. Such a converter exists for the Hypertext Hotel MOO [17]. In that system such a conversion is easier as the system does not have a spatial metaphor and the room descriptions *are* the hypertext. A converter cannot *create* room descriptions by itself. A converter for systems like TechMOO therefore would always be semi-automatic at best and be rather an extended structure editor than a real converter.

Another problem is that changing MOO structures is quite tedious. We think a part of the MOO has to be stable to provide a sort of navigational backbone -- most other MOO rooms should be reconfigurable dynamically. For this a specialized MOO system might be necessary or one could create a system with meta rooms that simulate sets of rooms according to a defining data structure (like a markup language).

So far we use only the connection from the MOO to the Web and navigation in the spatial environment is reflected in Web navigation. A connection from the WWW client to the MOO is possible, although a bit more difficult to realize. The picture of the Virtual Skiles in figure 5 shows already a navigation palette, albeit a non-functional one. One of our future goals is to provide a limited set of MOO commands through the WWW, for example commands to navigate in the standard directions (north, south,...) and to use the bulletin boards using forms.

Using the WWW server object we want to serve the contents of bulletin boards as Web pages. Comments on the board then should be linked to the Web page associated with the room the board is located in and provide a sort of asynchronous discussion facility. We see several other applications for serving HTML directly out of the MOO, not the least interesting of which is to provide read wear for WWW pages served from the MOO.

CONCLUSIONS

MOO environments by themselves are interesting tools for communication, interaction, and exploring novel navigational metaphors. The strength of the MOO are the communication facilities and the ease with which structures based on various -- mostly spatial -- metaphors can be created in it. Using a special MOO client -- *Juggler* -- we can associate MOO locations and objects with URLs and create the illusion of one information space with two views, one textual and one WWW based. This combines the navigational and communicational advantages of the MOO with the Web interaction. The system supports creating new structures using existing pages on the Web using diverse navigational metaphors. The MOO's communication facilities permit conducting guided tours and discussing material on the Web.

ACKNOWLEDGMENTS

This work is made possible by a research grant from the Austrian "Fonds zur Förderung der wissenschaftlichen Forschung" (Dr. Erwin Schrödinger Stipendium), Grant J01021-MAT for which we are very grateful. We would like to thank Jay D. Bolter, who first suggested using this quite simple approach to combine a MOO and the WWW as well as Anne Bourne, Carole F. Meyers, and Nick Sawhney for proofreading and valuable comments. We are very grateful to the anonymous reviewers for their very insightful comments.

REFERENCES

1. Bernstein, M. Enactment in Information Farming, Hypertext'93, Seattle, 1993, pp. 242-249.

2. Bernstein, M., et. al. Architectures for Volatile Hypertext, Hypertext'91, San Antonio, 1991, pp. 243-260.

3. Bruckman, A. and Resnick, M. Virtual Professional Community: Results from the MediaMOO Project, Convergence, 1, 1, (1995), ftp://media.mit.edu/pub/MediaMOO/Papers/convergence.*.

4. Curtis, P. Mudding: Social Phenomena in Text-Based Virtual Realities, accessible at ftp://parcftp.xerox.com/pub/MOO/papers/DIAC92.*, 1992.

5. Curtis, P. and Nichols, D.A. MUDs Grow Up: Social Virtual Reality in the Real World, ftp://parcftp.xerox.com/pub/MOO/papers/MUDsGrowUp.*, 1993.

6. Dieberger, A. Navigation in Textual Virtual Environments using a City Metaphor, PhD Thesis, Vienna University of Technology, 1994.

7. Dieberger, A. On Magic Features in (Spatial) Metaphors, SigLink Newsletter, 4, 3, (1995), http://www.lcc.gatech.edu/faculty/dieberger/magic_features.html, forthcoming.

8. Dieberger, A. Providing Spatial Navigation for the World Wide Web, in Frank, A. and Kuhn, W. (Eds.). Spatial Information Theory, Proc. of COSIT'95. LNCS 988, Springer, Semmering, 1995, pp. 93-106.

9. Dieberger, A. and Tromp, J.G. The Information City Project - A Virtual Reality User Interface for Navigation in Information Spaces, VRV'93, Vienna, 1993, accessible at http://www.lcc.gatech.edu/faculty/dieberger/VRV.html.

10. Erickson, T. From Interface to Interplace: The Spatial Environment as a Medium for Interaction, COSIT'93, Elba, 1993, pp. 391-405.

11. Evard, R. Collaborative Networked Communication: MUDs as System Tools, LISA'93, Monterey, 1993, ftp://parcftp.xerox.com/pub/MOO/papers/Evard.*.

12. Hill, W.C. and Hollan, J.D. Edit Wear and Read Wear, CHI'92, ACM Press, Monterey, 1992, pp. 3-9.

13. Kaplan, N. and Moulthrop, S. Where No Mind Has Gone Before: Ontological Design for Virtual Spaces, ECHT'94, ACM Press, Edinburgh, 1994, pp. 206-216.

14. Laurel, B. Computers as Theatre, Addison-Wesley, 1991.

15. Marshall, C.C. and Shipman, F.M. Spatial Hypertext: Designing for Change, Communications of the ACM, 38, 8, (1995), pp. 88-97.

16. Masinter, L. and Ostrom, E. Collaborative Information Retrieval: Gopher from MOO, INET'93, 1993, ftp://parcftp.xerox.com/pub/MOO/papers/moogopher.*

17. Meyers, T., Blair, D., and Hader, S. A MOO-Based Collaborative Hypermedia System for WWW, Proc. of 2nd International WWW conference, 1994, http://www.cs.brown.edu/people/twm/wwwmoo.html.

18. Mukherjea, S., Foley, J.D., and Hudson, S. Visualizing Complex Hypermedia Networks through Multiple Hierarchical Views, GVU TR 95-08, 1995, ftp://ftp.gvu.gatech.edu/ pub/gvu/tech-reports/9508.*

19. Taylor, H.A. and Tversky, B. Spatial Mental Models Derived from Survey and Route Descriptions, Journal of Memory and Language, 31, 2, (1992), pp. 261-292.

20. Tromp, J.G. and Dieberger, A. MUDs as text-based spatial user interfaces and research tools, Journal of Intelligent Systems, 5, 2-4, (1995), pp. 179-202.

21. Tversky, B. Cognitive Maps, Cognititve Collages, and Spatial Mental Models, COSIT'93, Springer, Elba, 1993, pp. 14-24.

22. Yates, F.A. The Art of Memory, The University of Chicago Press, 1966.

23. Zellweger, P.T. Scripted Documents: A Hypermedia Path Mechanism, Proc. of Hypertext'89, Pittsburgh, 1989, pp. 1-14.