to appear in Internation Journal of Human Computer Studies
Special Issue on innovative applications of the WWW

Supporting Social Navigation on the World Wide Web

Andreas Dieberger
Georgia Institute of Technology
School for Literature, Communication, and Culture (LCC) and
Graphics, Visualization and Usability Center (GVU)
Atlanta, GA 30332-0165
USA
email: andreas.dieberger@acm.org

Abstract

This paper discusses a navigation behavior on Internet information services, in particular the World Wide Web, which is characterized by pointing out of information using various communication tools. We call this behavior *social navigation* as it is based on communication and interaction with other users, be that through email, or any other means of communication. Social navigation phenomena are quite common although most current tools (like Web browsers or email clients) offer very little support for it. We describe why social navigation is useful and how it can be better supported in future systems. We further describe two prototype systems that, although originally not designed explicitly as tools for social navigation, provide features that are typical for social navigation systems. One of these systems, the Juggler system, is a combination of a textual virtual environment and a Web client. The other system is a prototype of a Web-hotlist organizer, called Vortex. We use both systems to describe fundamental principles of social navigation systems.

1. Introduction

When looking for information on the World Wide Web people make use of a variety of navigation strategies. One of the typical strategies is a more or less unoriented wandering, sometimes called *surfing* or *browsing*. As it is undirected it is not a very useful strategy when searching for specific information. Still people commonly browse a little in each Web session due to the hyperlinked nature of the Web. People also commonly browse a little after having found what they were looking for.

Very often they find unrelated information that none the less is interesting. Commonly users then store pointers to this information in their *hotlist* or *bookmark collection* and often they also copy such a pointer into a simple email message and send it to friends who they think may be interested in that information as well. This voluntary sharing of information with friends and colleagues is a typical example of what we call *social navigation.*

Besides this browsing scenario there are several other cases of social navigation observable on the Web. One of them is the more natural type of interaction between two users where one user asks another about information she assumes or knows the other user has the pointer to.

Yet another type of social navigation can be found in the Web pointer lists many people maintain on their own Web pages, which are essentially publicized hot lists for everybody else to use. Interesting information thus becomes accessible *through a person interested in it* which is another interesting facet of social navigation.

2. Navigational behavior on the World Wide Web

Navigation strategies on the Web often rely on the expertise of other users. As it is difficult to *re-find* information users collect lists of addresses of information in the browser's *hotlist* or in a *bookmarks* file. These addresses are called Uniform Resource Locators or URLs. Many users invest much time and effort to create well structured Web pages from these lists and make them available to other users by linking them to their home pages. When looking for information it is a common strategy to look at such pointer pages of people with related interests[1].

Another possibility is to consult a Web directory, most of which are maintained for free by Web users or companies. Just one example is the Yahoo (http://www.yahoo.com/). The Web has reached a state where the important issue for information providers is not to have information on the Web but to have a pointer to this information in a Web directory or many hot lists. Although many Web directories are companies that finance themselves by placing advertisements on their search pages these directories essentially still are a version of social navigation because they act as *structure providers* for other users as described in (Dieberger 1995).

The interesting and very special aspect of these types of navigation is that they are services that are essentially provided for free. In most cases people do not have personal profit from providing a pointer page -- except that they may be *known* to other people on the Web, which is mainly a social aspect of having a Web presence.

2.1. Navigation needs structure

An interesting aspect of pointer pages -- be they search engines or personal pointer pages -- is that they are a way to provide structure for an essentially unstructured information space (Dieberger 1995).

With the ongoing growth of the World Wide Web the expression *lost in hyperspace*, traditionally used to describe navigation problems in hypertext systems, has become popular again. The Web has reached a size that makes it almost impossible to quickly locate information on it. A lot of useful information is on the Web but information that is not easily accessible is as good as non-existing information.

For effective navigation users need a perceivable structure in the information space. Users cannot decide where to go when they don't know where they are and which direction they should head -- to use a spatial metaphor. Yet such spatial / navigational concepts are not available on the Web by itself. The Web does not provide meta-information for its pages and searching for information by aimless browsing is almost bound to fail.

2.2. Creating structure using pointer pages

Probably all Web browsers (or Web clients) provide a feature that allows users to store the address of pages (URL -- Uniform Resource Locator) they find interesting and to which they might want to return again. Lists of such pointers normally are accessible from a pull-down menu in the client software or from a *bookmarks* list (sometimes also called *hotlist*).

[1] T. Erickson, personal communication.

2.2.1. Personal Home pages with pointer pages

Although these hotlists provide users with an *address book* of Web addresses already many people invest a lot of work to convert their hotlists into nicely formatted and structured Web pages and make them freely available on the Web. Such pages are often called "my pointers", "interesting stuff", "cool pages" and so forth. People do not really have a reason to provide such a service for free but they do it anyway -- probably for reasons similar to why they have a home-page in the first place.

These pointer pages provide addresses of a relatively small number of pages that are classified and reviewed by a person and that (ideally) are checked for broken links, that is for pages that have been moved. In case a user knows the maintainer of the pointer page she can deduce from her knowledge of that person what pointers she is likely to find on the page. For example when a user looks for information on a certain singer -- let's say "Enya", it is a good start to look at the pointer page of a friend who is a fan of Enya's music.

We cannot stress enough how different such a navigational behavior is from all other types of hypertext navigation. There is no real search involved and no commercial service structures and classifies the information space for direct profit. Instead this type of navigation is a purely social process that works only because a user has background information about this friend who happens to like Enya music.

In traditional information domains, like a library, a similar navigational behavior is to ask a librarian. Also this type of social navigation exists on the Net but when looking at pointer pages there is one major difference: the user providing the information is not directly involved. Looking up a friend's pointer page is almost like asking my friend's secretary which library shelf my friend tends to take her books from. One could even say this approach makes use of another Web user as an *intelligent agent* as Tom Erickson described it.[2]

2.2.2. Institutional home pages

Similar to personal pointer pages are institutional pointer pages. There are two different cases: we have to distinguish between a pointer page maintained by a company and all others. The reason for this distinction is that a university department may provide pointers to other departments doing related research, whereas a company is unlikely to provide pointers to competing companies.

Although providing product and company information by itself is not very interesting in the context of social navigation many companies provide also pointer pages with useful *related information*. Just two examples are the Sun and Apple Web sites that try to stress that companies' involvement in the Internet and therefore provide extensive Web style guides (at http://www.sun.com/styleguide/ and http://www.cybertech.apple.com/HI/web_design/intro.html).

2.3. Related work

Although the term *social navigation* in the way we use it is relatively new, there is a large amount of related work. First we should point out Thomas Erickson's work, especially (Erickson 1996) where he explicitly talks about how people collaborate in Web navigation. Also in (Erickson 1993) he studies the social aspects of a information environment. Like in our Juggler system (see below) Erickson bases his observations on a textual virtual environment.

The term *social navigation* was used in at least one other publication, namely in (Dourish and Chalmers 1994) but with a slightly different meaning. Dourish and Chalmers see social

[2] T. Erickson, personal communication.

navigation as *navigation towards a cluster of people* or *navigation because other people have looked at something*. They separate navigation into social and semantic navigation, where an underlying semantic relationship in information is mapped on a spatial metaphor.

With the continuing growth of the World Wide Web also interest in any form of collaborative navigation and information filtering is growing. An example is work concerning collaborative information filtering in news messages by letting users vote on a message's usefulness. Examples are the GroupLens system (Resnick, Neophytos et al. 1994) or the MessageWorld (Rose, Borenstein et al. 1995).

An example of work concerning a collaborative Web agent is (Starr, Ackerman et al. 1996). The authors base their work on the assumption that the extra work necessary for voting is more than many people want to invest. A similar assumption is the basis of the PHOAKS system (http://192.20.225.67/phoaks/index.html) which uses frequency of mention to avoid a voting scheme (Hill and Terveen 1996). Also of related interest is (Maltz and Ehrlich 1995) which describes an information filtering system based on sending pointers to information.

3. Interaction on the Internet

The Internet supported interaction and communication between users long before the Web was designed. The original ARPAnet was a communication tool to reliably send messages. Initially communication on the Web was exclusively text-based -- only much later was it possible to send other types of information as well, like graphics, sound or video. Aside from the Web most Internet interaction is mainly text-based and many Internet users still have problems sending and receiving for example graphics files as email attachment -- and if they can, they often cannot view these files because they are in an unknown format.

The great achievement of the Web is to alleviate many of these problems by introducing standardized formats for formatted text and later also multi-media documents. But information on the Web is available only when the address of this information, the URL (Uniform Resource Locator) is known. The user either has to navigate to the location containing the information as described above or this URL must be communicated to her in some way. Traditional Internet communication tools can be easily used to exchange URLs.

3.1. Overview of Communication on the Net

In this section we look at several commonly used means of communication on the Internet and describe how each of these supports the sharing of pointers to Web information.

3.1.1. Electronic mail

The probably most common method to send a URL to somebody else is to type it into an email message and to send it. In systems with graphical user interfaces typing can often be replaced by a copy/paste operation. This makes the transfer of a URL into an email message easier and less error prone.

Still this copy process is a bothersome procedure, often involving two or more software packages to get the information from one user to another. Although much better and more reliable than typing, copy/pasting is still an error-prone procedure and it takes time.

To ease the transfer of URLs people frequently use special formatting in their email to make URLs easy to copy or to make them visually prominent. Examples are intended URLs on separate lines or lines of dashes to separate the URL from the remaining email content. These *(social) protocols* of URL mailing evolved because it is so bothersome to copy/paste URLs from emails. Still this does not solve all problems because some URLs are very long and may get line-

wrapped in the message. This necessitates additional steps to splice the line-wrapped URL into on piece again.

3.1.2. Newsgroups

Transmitting URLs in newsgroup messages is very similar to sending them in email. News messages often have a close connection to the Web and modern news clients often recognize URLs and show them as active links in the news message. An example for such a system is the Nuntius news client for the Macintosh (ftp://ftp.ruc.dk/pub/nuntius/) or the news reader integrated in the Netscape Navigator (http://home.netscape.com/). URLs in a news message are underlined and colored blue (like in most Web clients). Clicking them causes the URL to be loaded in the Web client. On the Macintosh this works using AppleScript, a technology that allows one application to remote-control another one.

Recognizing URLs generally is done by searching for lines containing the text "http://" and then scanning forward till a character that is not valid in a URL is found. This method has the disadvantage that URLs that point at other services (for example gopher:, news:, mailto:...) are seldom supported. Also this method cannot recognize abbreviated URLs that do not use the http:// in the beginning (for example "www.gatech.edu").

Even with these shortcomings can we assume that similar recognition features will soon be available also for email clients.

3.1.3. Chat rooms and textual virtual environments

Contrary to asynchronous communication tools like email and news, chat rooms and textual virtual environments (like MUDs -- Multi User Dungeons or MOOs -- MUD Object Oriented) provide synchronous communication. This means that two users who want to communicate have to be present at the same time to *talk* to each other.

Two or more users connect to a server that acts as a communication relay. Messages sent by a user are immediately relayed to all other users in that discussion. The delay between sending and receiving may be just a few seconds. The need to read what was *said* and to type a reply makes this type of communication a quasi-synchronous medium with response times of under a minute (largely depending on network lag and typing speed).

Also these systems can be used to communicate Web pointers to other people. People can even discuss material on the Web by just *talking* about it. However there is no real interaction, like pointing out a certain line in the document. If I need to know the address of a certain web page I ask another user in the chat or MOO room and may get a URL *said* to me. I then can either type this URL into my web browser or copy/paste it.

It is obvious that also this type of navigation would be much easier if also these systems could recognize URLs. This is essentially what the Juggler system does (see section 4). URL recognition by itself supports only the receiving end. True social navigation occurs when there is also support for the sending side as we will describe in section 4.

3.1.4. Other media

Nowadays, as the Web is so "in", clever marketing uses the novelty character and the hype of the Web for marketing purposes. Many new products now feature their own home pages and URLs appear on advertisements, in newspapers, on television and even on the radio.

These URLs are *dead pointers* because they have to undergo several transfer processes before people can actually use them. The main reason for this is that URLs are long and hard to

remember. People have to write them down to store them. In case somebody sees such a URL and first communicates it to a friend via telephone the situation is even worse.

We sometimes have the impression that advertising using URLs works mainly because they are so ugly and new that people are bound to notice them. URLs stick out of an ad like a sore thumb. It heightens the visual intensity of the ad and it seems almost everything that manages to do that is a legitimate means in marketing today.

Web pointers in marketing will become useful communication tools when it is not any more necessary to scribble them down to remember them for more than a minute, that is when -- for example -- the name of a company alone is sufficient. Note that starting with Netscape Navigator version 2.0 this almost works: instead of typing a complete URL like http://www.mycompany.com/ Netscape accepts a URL of the form www.mycompany.com and in the case of a URL of the form www.***.com the middle name (***) alone is even sufficient.

In our example the URL "mycompany" therefore would yield the desired result. Communicating a URL then suddenly becomes easy. Note that this scheme presently works only with the ".com" domain. This does make sense because the ".com" domain is most likely to be used this way. Future browsers should be able try all existing domains in a predetermined sequence. A URL of the form "mycompany" could then try the URLs http://www.mycompany.com/, http://www.mycompany.edu/, http://www.mycompany.org/, and so forth.

3.2. Interaction must be much smoother

In section 3.1. we described several current methods to communicate Web pointers to other users. However people do not want to interact with URLs -- they want to point out information in an information system. Exchanging addresses for information is a cumbersome intermediate step that should be eliminated if possible.

Note that this argument does not even consider other problems with URL, like the fact that a Uniform Resource Locator (URL) describes one exact location of information. Should the information move the URL becomes invalid. URLs also don't allow the browser to automatically access the most recent or the closest (cheapest) version of the information. For more information on these issues see for example (Andrews and Dieberger 1996).

3.2.1. More natural support for pointing and showing

Assuming we want to stick to the basic URL method for some more time we have to ask how we can ease the interaction with other people to point out information? The answer is quite simple: the URL proper has to disappear and instead users should get a handle to the information that allows them to interact with it in a much more natural way. When asking for directions in a city people will probably point at a building instead of writing down the exact address. Similarly in the Web it should be possible to point out information naturally and not to have to cope with the address of the information. Users need to get a handle to the information, that essentially is a metaphor wrapped around the basic URL. This method does not replace the use of the URL internally, but it effectively hides the URL from the user.

An example for such a handle would be a news message which contains document icons the user can select and interact with. Each of the icons represents a web page linked to the news message. The URL itself is hidden but through the icon the user can access the page and associated meta-information, like the title, keywords, abstract, modification date, size of the page and so forth. Note that a plain page symbol alone would not be a real improvement as people need more information on the document before they can decide if they really want to access the page.

The page symbol by itself also has the disadvantage that users can activate the page but this does not directly help them to further distribute that pointer to other people. They can only activate the

page and then again copy/paste the URL from the newly loaded page into another message. Instead page handles ideally are objects that can be copy/pasted themselves, while preserving all associated meta-information. They also allow users to include them in other message they author, like emails, news and so forth. Such functionality requires a standardized representation of a page that includes meta-information and can be transformed into different representations according to the task at hand. For example a page handle that is copied into a normal text-editor would represent the meta-information in human readable form. Should this text be copied into a news message the news client should be able to parse that format and convert it into a page handle again.

Such page handles allow users to manipulate references to Web information as one coherent piece without having to worry about forgetting to copy meta-information. As these handles are manipulable objects in the user interface they can be dragged around, copied, like any other object.

As the development on the World Wide Web happens so fast it is no surprise that even while this paper was still in the works systems that are close to this vision actually were developed. For example it is possible to drag and drop URLs as objects between applications on the Macintosh platform now and similar functionality exists probably also on other platforms. Also Apple's Project X system (see http://mcf.research.apple.com/) is quite close to our ideas and the Meta Content Format (MCF) is a format for describing meta-information as we advocate it. For more information on MCF see http://mcf.research.apple.com/hs/mcf.html.

3.2.2. Seeing other users

Seeing other users does not necessarily mean a video representation but *awareness of the presence of other users*. This awareness can be a special type of meta-information for a Web page. It gives an indication of how much demand there is for a certain type of information at a certain time and it can also give information about who is accessing that infomation. Be aware though, that representing users accessing information brings in a host of privacy issues. On the positive side such a representation can support better interaction in the virtual information space among members of a group working one project. For an excellent discussion of some of these possibilities see (Bernstein 1993).

One example for a less futuristic realization of such user awareness is a history enriched environment that changes from the interactions with it. This method provides indirect hints of other people and therefore avoids pricacy issues.

But both of these methods miss one important point: Sometimes users simply want so know how the entire user population moves through the information environment, or if there are locations that suddenly attract large crowds of users. Such amassments of users have been called *flash-crowds* in literature, see (Andrews and Dieberger 1996). A system showing such information may be used to avoid accessing servers that are temporarily overwhelmed with user requests and using a mirror site instead.

An example of such a system was developed by Alex Azhao, Mike Pinkerton and Jim Pitkow at Georgia Tech as a student project. It showed a graphical representation of the pages on a Web server and moving points between these pages to represent users. Such a system gives users a good sense of what is going on where. It would not be easy to create such a system for large sections of the Web however, because the visual representation of a large parts of the Web by itself is still a difficult problem. For another example of how users accessing information could be represented see (Bernstein 1993).

This overview shows that there is a number of methods to exchange pointers to information on the Web today but that most of these tools are relatively bothersome to use. Social navigation therefore can occur but most systems do not exactly support it. Most communication systems on

the Web do not provide an easy and natural way to share information and they do not create any awareness that the Web is a large *shared* information space. They rather show the Web as a huge information repository in which we are the only user.

4. Example: "Juggler"

The Juggler system principally takes the idea of recognizing URLs in the output of a communication tool and hiding them from the user and extends it to an integrated system that supports social navigation. Imagine two users communicating with another in a chat room or a textual virtual environment. One user talks about an interesting Web page she has found. Now she wants to show that page to the other user. She will copy the URL into a *say* command, the other user will then copy the URL from the output and paste it into her favorite Web browser and hit return.

It is obvious that this process can be greatly simplified by automating this process. This was the basic idea for the original design of the Juggler system: a MOO client that recognizes URLs in the output and automatically tells a Web client to load a URL the moment it finds one. Juggler was designed to support interaction between a teacher and students in a remote teaching support system at the School for Literature, Communication and Culture. The system was not meant to be a distance learning system but rather a way to communicate with the teacher in their office hours without forcing students to come to campus each time.

Through the author's special focus on spatial metaphors for navigation the Juggler system used spatial metaphors for navigation and also served as tool to study spatial metaphors see (Dieberger 1996). Using the layout of a real environment our virtual meeting space further enhanced the meaning of social interactions within this space like described in (Erickson 1993). In a follow-up project the Juggler system, consisting of a modified MOO system (MUD Object Oriented, a textual virtual environment developed at Xerox Parc) and the Juggler client were used in a virtual graduate student conference on Romanticism. The users in this conference had little Internet experience and so we created a simplified version of the overall system that had fewer features, but better support for social navigation. The examples in this section are taken from this second version of the Juggler system unless otherwise noted. For detailed information on the Juggler 2.0 system, please refer to the manual which is available on the Web (Dieberger 1996).

4.1. Textual Virtual Environments

Textual virtual environments developed out of networked, textual adventure games. Users log into a central server maintaining the system state (game status) and generating appropriate feedback for user actions. Typical examples of such systems are MUDs (Multi-User Dungeons) or MOOs (MUD Object Oriented) like the system we used for the Juggler system. MOO systems seldom have game character. Instead they are systems to support communication and interaction between a group of users. In the following we will use only the acronym MOO for simplicity but we would like to stress that most functions we describe can be implemented in any other type of textual virtual environments as well.

A MOO is a virtual environment in which everything, including the user and her actions are described by text. When a user says something this is described as

```
Andreas says: "Hi there"
```

If the user uses an *emote* command to smile this is described as

```
Andreas smiles.
```

MOO systems provide users with a wealth of verbal and non-verbal communication commands which make communication and interaction in the system very flexible (Curtis 1992). The system

response time is quite short so we can see MOO communication as *almost real-time communication*.

MOO environments have more than one *location*, that is, users can be in different *rooms* and communication works accordingly -- only people in the same room can hear when a user talks. MOO environments, albeit only textual systems, provide users with a surprisingly rich and -- after getting used to the environment -- quite realistic environment for interaction and communication. For more general information on these systems see (Curtis 1992; Curtis and Nichols 1993; Erickson 1993; Bruckman and Resnick 1995). For more information on technical aspects of the Juggler system and on how Juggler supports spatial navigation see (Dieberger 1995; Dieberger 1996).

4.2. Talking to people and pointing out pages by saying them

The basic idea of the Juggler system -- as mentioned above -- is that the MOO client recognizes URLs in the MOO's output and causes a Web client, in our case Netscape Navigator, to load that page. Netscape is remote controlled using AppleScript and Juggler is implemented on a Macintosh computer using HyperCard.

URLs are recognized by searching for the typical prefixes in URLs, like http://, gopher:// and so forth. From such a found prefix the output is scanned forwards till a character that is not valid in a URL is encountered. The URL then is hidden from the output and sent to Netscape to be loaded.

The effect of this setup is that if a user talks about a web page she found using the command:

```
say Have a look at that page http://www.gatech.edu/
```

Another user will see (depending on the preference settings) either

```
Merlin says: "Have a look at that page http://www.gatech.edu/"
```

or

```
Merlin says: "Have a look at that page "
```

The second case may look a little strange by itself but as the corresponding Web pages pop up in Netscape out of nowhere users soon learn what happened. Easy as this looks already it is only an intermediate step to really supporting social navigation for two reasons:

- The other user still has to copy/paste or type the URL into a MOO command
- The other user has to describe by hand that she will now send a URL.

We soon recognized this shortcoming and changed the design of the Juggler client by providing a facility to easily point out Web pages (see section 4.4.).

It must be noted here that this simple setup of scanning for URLs does not work for shortened URLs without the http:// prefix. However there is a sort-of standardized MOO-Client Protocol (MCP) that has been developed for the Jupiter and the AstroVR systems (Curtis and Nichols 1993; VanBuren, Curtis et al. 1995). MCP uses *out-of-band commands* to establish a virtual communication channel between the MOO server and the MOO client. The loading of Web pages can then be caused by sending a MCP-command to the client thereby eliminating the need to search for URLs in the output. Juggler supports also this particular MCP command.

4.3. Looking at people and objects

One of the advantages of scanning for URLs is the simplicity with which URLs can be associated with all types of activities and objects in the Juggler system. It is just necessary to embed a URL in the description of an object and when looking at that object (which displays its description) the URL is output and loaded by the client.

An application for this behavior is to insert a home page URL into a user's description. When looking at the user the home page pops up. When an object has a URL associated with it users can give such an object to other users. Looking at the object then retrieves the associated Web page. The object's overall description serves as reminder and access metaphor for the Web page. Typical examples are objects that represent books, letters or signs.

Similarly it is possible to associate URLs with locations in the MOO so that each room automatically displays a certain Web page on entering. These aspects pertain more to the spatial navigation aspect of the Juggler system. For more details see (Dieberger 1996).

However this allows people to easily access a user's home page without having to deliberately ask for the address of that page. Thus such an activity is similar to accessing people-related information through a person's home page on the Web: It does not directly involve that person. Contrary to accessing a home page on the Web this approach does not even require the exact URL -- instead users directly look at the *person* to access the information. As I mentioned before the main advantage of querying pointer pages of another Web user is that it is unintrusive -- being able to obtain the address of a person's home page in the MOO carries this thought one step further.

4.4. Point-out button for people

The Juggler client provides a feature to easily point out material on the Web. The basic assumption is that both users who want to share information are using the Juggler client and a Web client at the same time. Browsing the Web essentially is independent from the communication channel in the MOO unless URLs are transferred. When a user find an interesting page of which she wants to transmit the URL to the other user it is therefore obvious which page the user wants to point out -- it is the page in the front-most window of the Web browser. Juggler provides a button that grabs the address of the page currently displayed in the Web client, wraps it into explanatory text and sends it as *emote* command to the other MOO (see figure 1).

Clicking that button results in the following output:

```
Merlin points out a page Web page with the title Netscape: Georgia Tech
Homepage.
```

and the corresponding page pops up on the screen of all users who are present in the same MOO room. Because this method of pointing out information is so much easier than copying/pasting it or typing the URLs this feature truly supports social navigation in the system and users very frequently make use of it.

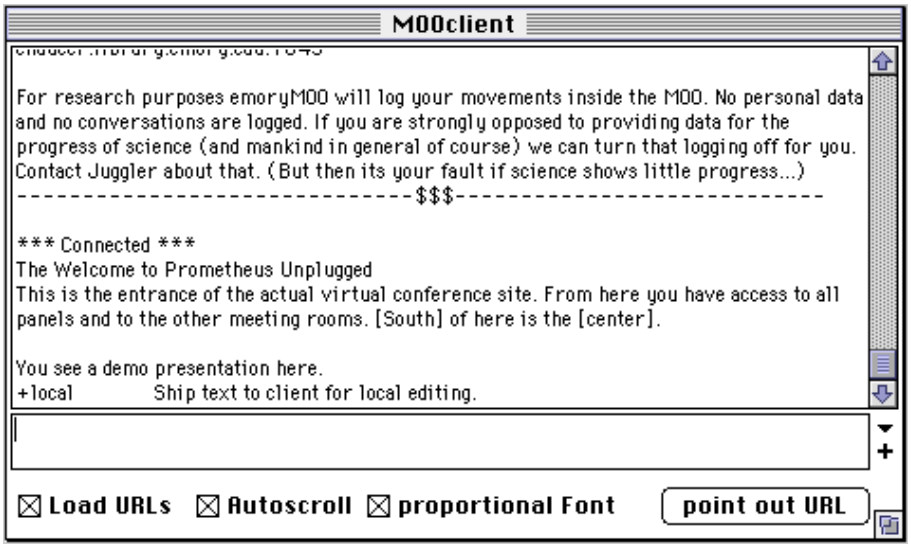


Figure 1: Juggler MOO client. Note the *point out URL* button.

There are certain drawbacks of such a simple implementation. First of all the emote command addresses all users currently in the same MOO room. It is very easy to annoy people by constantly pointing out Web pages -- especially when they are still reading the previously sent page. The obvious solution to this is to temporarily switch off the loading of URLs, but then users may miss interesting pages. Another solution (not implemented in Juggler) is to store all URLs in a conversation log using document handles like described in section 3.2.1. so that the URLs can be easily recalled and used within the context of the conversation.

The Juggler client and the MOO system we used for the virtual conference were instrumented so that all navigation activities and also all typical social navigation activities (like the use of the point-out button) were logged. Although these logs are not formally evaluated yet we can conclude already that the availability of such a simple and effective point out feature increased the social navigation activity. Both in the logs and also from direct experience in MOO sessions we observed that users of the Juggler client were much more likely to point out Web pages than users with other clients. This observation is not surprising considering the effort such an operation requires in a normal MOO client.

From conversations with our users we further know that they generally liked the point out button and that most users set their client to not display URLs in the MOO output. Although these are preliminary results we consider them informal support for our thesis that people do not want to see URLs and that they crave for a tool that allows them to share Web pages in a very natural way.

A related form of pointing out pages was realized in the earlier versions of Juggler but was eliminated for the conference version. This feature could parse a local HTML file and extract all URLs in that file. These URLs were provided in a pop-up list and could be pointed out by selecting from that pop-up list (see figure 2).

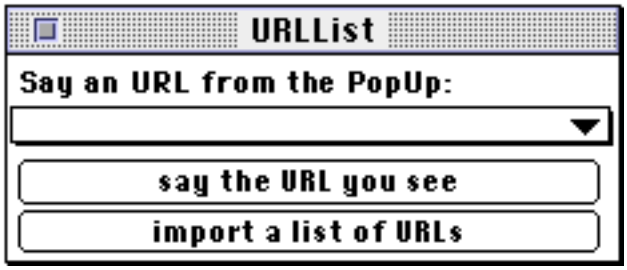


Figure 2: The URL palette from an earlier client version.

The basic idea of this feature was to support student / professor communication when they discussed a set of Web pages the student had created or to point out course material that may be collected in a pointer file already. In the version for the virtual conference we didn't see any similar use of this feature so we eliminated it.

4.5. Pointing out Pages to Objects

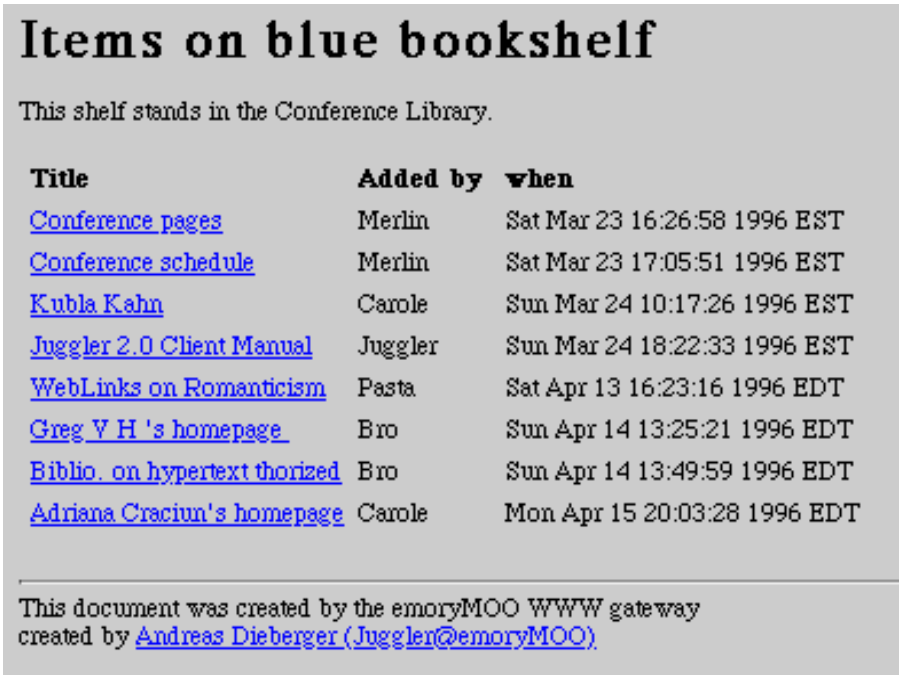
Pointing out information to other users restricts the system to a more or less synchronous communication system -- users have to tell information to people directly or they will not see it. This is where other information objects in the Juggler system become interesting. These objects use metaphors of information carriers to convey URLs and meta-information. Examples are a message on a bulletin board, a MOO internal mail message, an object on a bookshelf or a slide projector that presents a sequence of Web pages with explaining text. Information can be placed in these objects and can be later retrieved by other users at their leisure.

In order to make the system more useful we use the features for pointing out pages also for the creation of information objects. Admittedly it stresses the metaphor when users can point out a Web page to a MOO bookshelf but users coped quite well with this. To insure that only the bookshelf sees that page and not everybody in the room as well, the user first has to notify the bookshelf that she wants to add a book. Again this is not necessarily the ideal solution to this problem but it works well and it keeps the number of features in the system at a minimum. We assume that providing information objects with a *personality*, for instance by making them behave as *agents*, this pointing-out could actually be designed be appear logical.

From conversations with Juggler users we know that users quickly adopted this feature. However due to a bug in the original bookshelf object some URLs were lost and users lost their faith in the use of the point-out button for the bookshelves. This shows that users are more than willing to adopt a simple way to share a pointer even with objects (agents) but only if that feature is reliable as well.

The virtual conference used the point-out button only in conjunction with the bookshelves. It is very conceivable to provide similar listening abilities in all other objects that can contain URLs, like in room or mail editors. For bulletin boards Juggler provides a local editor with a point-out button inside the editor.

Objects on the bookshelves are available as pointer lists both inside the MOO and through the WWW gateway of the MOO system (see figure 3).



Items on blue bookshelf

This shelf stands in the Conference Library.

Title	Added by	when
Conference pages	Merlin	Sat Mar 23 16:26:58 1996 EST
Conference schedule	Merlin	Sat Mar 23 17:05:51 1996 EST
Kubla Kahn	Carole	Sun Mar 24 10:17:26 1996 EST
Juggler 2.0 Client Manual	Juggler	Sun Mar 24 18:22:33 1996 EST
WebLinks on Romanticism	Pasta	Sat Apr 13 16:23:16 1996 EDT
Greg V H 's homepage	Bro	Sun Apr 14 13:25:21 1996 EDT
Biblio. on hypertext thorized	Bro	Sun Apr 14 13:49:59 1996 EDT
Adriana Craciun's homepage	Carole	Mon Apr 15 20:03:28 1996 EDT

This document was created by the emoryMOO WWW gateway
created by Andreas Dieberger (Juggler@emoryMOO)

Figure 3: A bookshelf in the WWW gateway.

4.6. History Enriched Environments

Another important aspect of social navigation is information about how people think about information, which information they use most and so forth. On many Web pages authors nowadays place so-called hit counters that show how many people accessed that particular page within a certain time span. Such a counter gives a good indication on how popular a page is. Related to this simple feedback of the popularity of Web pages are approaches that implement voting schemes like described in (Rose, Borenstein et al. 1995) for a news system or the PHOAKS system (Hill and Terveen 1996).

Within the MOO we used an access-counter to turn the MOO environment into a history-enriched environment showing *read wear* (Hill and Hollan 1992). The exits (connections) between MOO rooms count how often they have been used and use this information to point out which navigation paths in the MOO are most popular. Contrary to the Web where hit counters show only point information this facility highlights paths to interesting and frequently used rooms as described in (Dieberger 1996).

An example is the following room description which points out frequently used exits:

```
The Entrance

You are in the entrance of the emoryMOO world. The entrance is a large
plaza with a tall statue of an angry looking man in the middle. Hallways
lead in four directions. Directly (North and North ) is the conference
center. To the [west] lies the library and [east] is the entertainment
area. If you are new here and especially if you never used a MOO before go
[south] to the information area. Somebody hung a bulletin board on the
statue's big toe. Fixed only with leather straps it gently sways left and
right. You can {look at board} or {list board}.

The exits (North and North ) and [east] seem to be used above average.

You see a Bulletin board on Prometheus' big toe, a statue, and a sign here.
```

Following these frequently used exits eventually will guide users to heavily used places and objects that contain pointers to often used information. For this *read wear* to be really useful it must provide a decay function so that old information does not overshadow new information. In the virtual conference we used a function that halved the count of each room exit every two weeks and this function seemed to work quite well.

Especially for new users of our virtual conference this feature helped them to quickly find the areas of most interest, which in this case meant the areas where they were most likely to encounter other users.

Similarly to this concept is the *read wear* we used on the bulletin boards. A listing of a bulletin board always contains an indication of a message's age and of how often is has been accessed. In the MOO itself this information was conveyed using textual descriptions. This was not really helpful as these messages used a misleading metaphor: pages that were new but accessed very often were described as "old" whereas old pages that were scarcely ever read appeared as "new".

```
list board

There are 11 messages on the Bulletin board on Prometheus' big toe.
1  Juggler       Juggler 2.0 released      recent  (Apr  8 16:56)
2  Carole        Who said?                 normal  (Apr  8 21:49)
3  WWW:Carole    A Promethean invitation   normal  (Apr  9 15:31)
4  WWW:R.A.      For the Promethean challe recent  (Apr 11 15:03)
5  WWW:Laura R.  Re:Who said?              recent  (Apr 12 11:18)
6  WWW:Carole    Re:Who said?              recent  (Apr 13 16:11)
7  Pasta         MOO Events                recent  (Apr 18 14:42)
8  WWW:pasta     Re:MOO Events             recent  (Apr 18 14:43)
9  Carole        We need your help!        new     (Apr 24 21:33)
10 WWW:BigAl     Frustration               recent  (Apr 30 11:58)
11 Mark          Re:Frustration            new     (May  5 14:07)
```

The color code we used on the Web gateway for the bulletin boards proved to be more intuitive. The code -- although not necessarily a direct indication for *recentness* faded from red (new) to gray and then changed into a more greenish gray when a message was old but accessed a lot (see figure 4).

Note that the author names on some postings are prefixed with "WWW:". These messages were written using the Web gateway. Without reaching deep into the technical bag of tricks it is not possible to guarantee the identity of the user creating a posting through the Web gateway. The prefix indicates this insecurity and thus alerts the user that the identity of the author has not been verified. In a setting where social processes govern the exchange of information and pointers to information knowing the identity of a sender is very important. In cases where people deliberately want to stay anonymous they still can make use of *guest* characters but care has to be taken that a certain user name can be associated with a certain person or that users at least know when this association cannot be guaranteed.

The wear on bulletin board postings automatically provides read wear for Web pages associated with them. Note however, that this read wear is not *global* like in a hit counter but that it shows the usage by a selected group of users -- all those who access that Web page through this particular posting on the bulletin board.

Although Juggler initially was not meant to be a social navigation system some of its features come close to what we feel a true social navigation system should look like. Obvious improvements would include better handling of meta-information for exchanged pointers, the possibility to log conversations with the URLs in them, more control of who to point out pages to and more consistent and improved use of read wear.

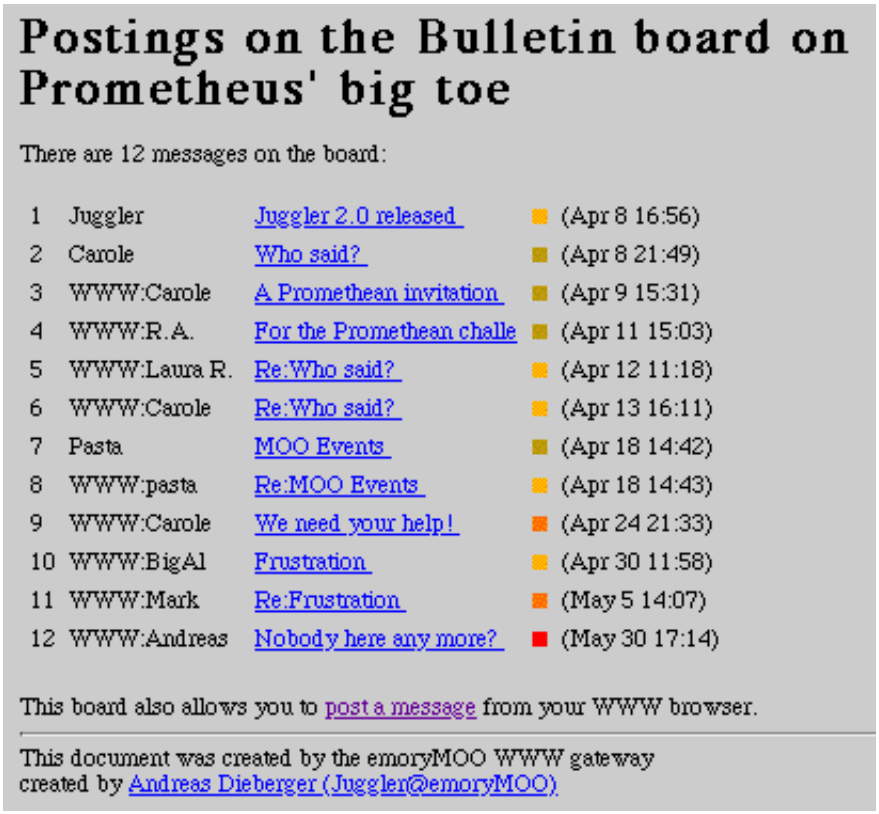


Figure 4: Bulletin board in the WWW gateway with the color-coded read wear.

5. Example "Vortex"

An entirely different system is the Vortex. It provides an alternative to keeping URLs in the conventional browser hotlist. Vortex uses a simplified desktop on which users can collect and manipulate objects that represent URLs. They can group pointers into sub-vortices or cluster, comment, and annotate them and so forth. Users can change the appearance of pointers to reflect how they perceive the size and importance of a Web page and if a server tends to respond slowly.

Vortex collections can be exported into pointer lists. The system remembers how often each page has been accessed and uses this information to structure the exported hotlist.

5.1. Collecting pointers as icons

Vortex is a HyperCard based prototype that communicates with Netscape Navigator through AppleScript. By dragging a new pointer out of the generator button in the bottom left (see figure 5) a new pointer -- or rather a handle, to use the terminology from section 3.2.1. -- containing the URL information of the currently showing page is created. Presently this stores the name and URL of the page. Additional page attributes (like keywords, page size,...) still have to be set by hand but future versions should retrieve such meta-information automatically and also generate an abstract of each page.

Essentially Vortex treats URLs as document handles. The icon of the handle indicates the size and the degree of usefulness of a page with additional information accessible in an information dialog box.

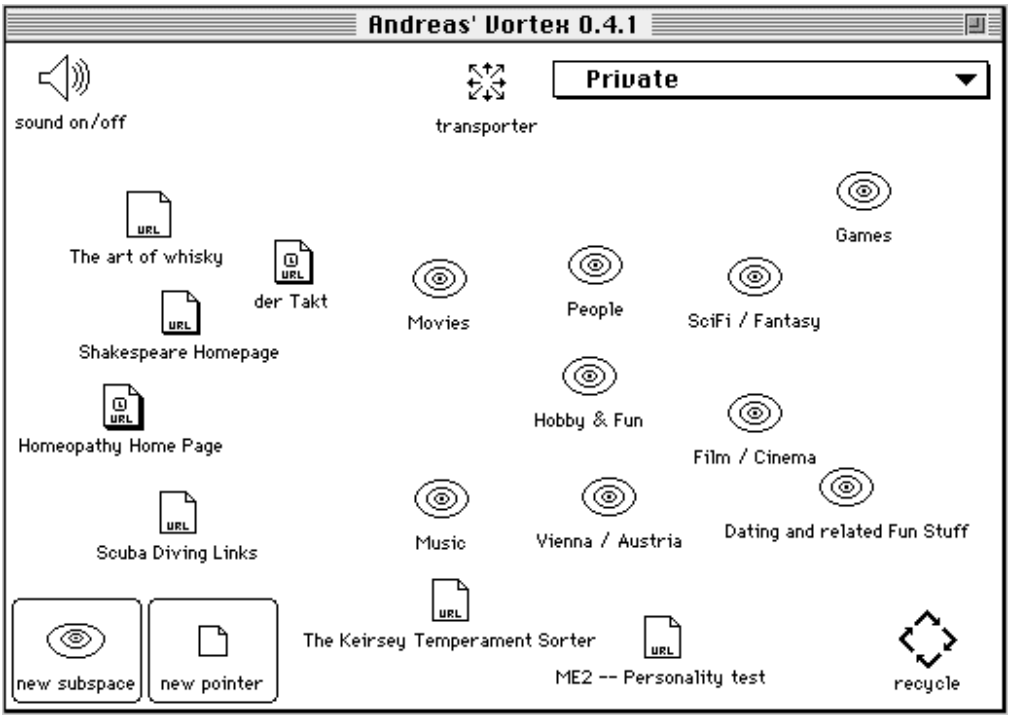


Figure 5: A typical Vortex screen.

Double-clicking a pointer in Vortex loads the corresponding page in Netscape Navigator and the page's meta-information (usage counters, last used date,...) are updated. Pointers can be directly manipulated and each such manipulation treats the URL together with all associated meta-information as one object. For example pointers can be dragged into sub-vortices, copied into the clipboard and so forth. The transporter moves pointers to the top Vortex (home space) and the recycler deletes pointers (and vortices) dragged onto it.

5.2. Support for copying pointers

Vortex originally was not intended as a collaboration or communication tool other than to generate pointer pages in an easy way, which is a type of social navigation as discussed above. However we soon started to incorporate additional features for social navigation. One of them is a function to copy pointers into emails -- by clicking on a pointer while holding a modifier key the URL and the meta-information are packed into a nicely formatted string that is easy to read for the receiver of the message. Ideally this string would be in a standardized format so that a pointer could be converted back to a complete page handle on the receiving end.

Vortex does support receiving URLs via email: it parses text file and placing all found URLs into a list of pointers to create. The user selects which pointers to import and handles for these URLs are created in the currently opened Vortex. Future version of Vortex should recognize keywords and use filter procedures to place new handles in the correct places, similar to the procedures described in the Piles interface described in (Mander, Solomon et al. 1992; Rose, Mander et al. 1993).

5.3. Supporting creation of hot lists and sending collections

In addition to the simple interactions described in the last section Vortex also uses a markup language (Vortex Definition Language -- VDL) which can describe a complete Vortex with all its substructures as a text file.

Vortex is a very early prototype. We currently think of a port to Java and to extend the Vortex' social navigation and pointer management facilities. A Vortex applet would provide a more or less system-independent way to communicate collections of Web pointers between users. The applet would read pointer collections defined in VDL and display them as a hierarchical pointer list, as a list in a multi-frame document or as a graphical rendering of the original Vortex with its desktop-like metaphor.

In addition to this browsing scenario a Vortex applet might allow users to modify a collection by adding, annotating or deleting pointers. The Vortex then could be uploaded to the server to be modified by the next user. With such a setup a distributed group of people could work on one shared collection of pointers. We did not consider problems or exclusive write access problems and read/write access rights here for simplicity.

6. Future developments

The Web is such a fast-moving environment that it was very likely that the first commercial social navigation systems would appear before this paper would be published. Indeed there exists at least one such system already. It is called Powwow and essentially is a chat-based client that allows users to easily exchange pointers and supposedly even provides simple white board functionality (http://www.tribal.com/powwow/).

Many newer MOO clients also provide features for recognizing URLs in the output. One example is the TinkeriView Client (see http://www.tinkeri.com/). However it seems that none of the newer systems provides a feature like Juggler's point-out button yet.

Also the first Java-based MOO clients have been announced. These clients are able to eliminate the bothersome separation into MOO client and Web client and future such clients will provide users with one integrated environment for MOO/WWW browsing (see for example the Tecfa project http://tecfa.unige.ch/moo/short_guide.html).

The new versions of Netscape Navigator and probably also other Web browsers provide support for chat rooms and it is probable that these features soon will also support pointing out Web pages to other users.

Also features like read wear find their way into standard Web design. One example is http://www.minds.com/. On this site there is a collection of discussion forums and on the head page of each forum the *most active discussion threads* are directly accessible -- a beautiful application of the read wear principle.

As mentioned above a standard for the exchange of document meta-information called MCF (Meta Content Format) has been proposed and is used in Apple's Project X system (http://mcf.research.apple.com/). Future social navigation systems will use such standards to provide handles to the documents as described and methods to easily create and maintain such handles. This will include methods to transfer them between various communication and data manipulation tools (databases) to painlessly share pointers to information with colleagues over the network making these future systems true social navigation tools.

7. Conclusions

We described a navigation behavior on the Internet, in particular the World Wide Web which is characterized by free interchange of pointers to information using diverse communication and interaction tools like email, news, chat rooms or textual virtual environments. Contrary to traditional hypertext navigation behavior social navigation is characterized by social processes like asking for pointers and freely sharing information with complete strangers without expecting anything in return. We believe one reason for this willingness of sharing information is that the

Web does not have any visible structure. Users therefore impose structures on the Web by defining structured pointer lists for example.

Social navigation is a very real phenomenon although most communication tools on the Internet today provide very little support for it. We described several requirements for easing social navigation like making URLs less visible, providing handles to Web documents that contain and maintain meta information and we advocated features that make it a very natural and easy process to point out information to other users.

We also described two systems, namely Juggler and Vortex, that although they were not designed as social navigation systems show typical aspects of future social navigation systems. We described experiences with these systems and possible future extensions.

Acknowledgments

Part of this work was funded by a research grant from the Austrian "Fonds zur Förderung der wissenschaftlichen Forschung" (Dr. Erwin Schrödinger Stipendium), Grant J01021-MAT. Further work on the Juggler system was done for the "Prometheus Unplugged?" Virtual Graduate Student Conference on Romanticism at Atlanta's Emory University (April 1996), organized by Mark Ledden and Carole Meyers. We would like to thank Jay D. Bolter, who first suggested to connect a MOO client with a Web client and to Tom Erickson who triggered our interest in what we now call *social navigation*. We also would like to thank the emoryMOO community and Anne Bourne for their input.

References

Andrews, K. and A. Dieberger (1996). Reinventing the Wheels? Usability problems on the World Wide Web. accessible at http://www.lcc.gatech.edu/faculty/dieberger/Usability_and_Web.html.

Bernstein, M. (1993). Enactment in Information Farming. Hypertext'93. Seattle**:** 242-249. .

Bruckman, A. and M. Resnick (1995). “Virtual Professional Community: Results from the MediaMOO Project.” Convergence **1**(1). .

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

Curtis, P. and D. A. Nichols (1993). MUDs Grow Up: Social Virtual Reality in the Real World, Xerox Parc. accessible at ftp://parcftp.xerox.com/pub/MOO/papers/MUDsGrowUp.ps.

Dieberger, A. (1995). Providing Spatial Navigation for the World Wide Web. Spatial Information Theory - Proceedings of COSIT'95. A. U. Frank and W. Kuhn. Semmering, Austria, Springer. **LNCS 988:** 93-106. .

Dieberger, A. (1996). Browsing the WWW by interacting with a textual virtual environment - A framework for experimenting with navigational metaphors. Proc. Hypertext'96. Washington DC**:** 170-179. accessible at http://www.lcc.gatech.edu/faculty/dieberger/HT96.paper.html.

Dieberger, A. (1996). Juggler 2.0 Manual, Georgia Institute of Technology. accessible at http://www.lcc.gatech.edu/faculty/dieberger/Juggler20/Juggler20_docu.html.

Dourish, P. and M. Chalmers (1994). Running out of Space: Models of Information Navigation (short paper). HCI'94 (British Computer Society). accessible at ftp://parcftp.xerox.com/pub/europarc/jpd/hci94-navigation.ps.

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

Erickson, T. (1996). “The World Wide Web as Social Hypertext.” Communications of the ACM **39**(1): 15-17. .

Hill, W. and L. Terveen (1996). Using frequency-of-mention in public conversations for social filtering. to appear in Proc. of CSCW'96. .

Hill, W. C. and J. D. Hollan (1992). Edit Wear and Read Wear. CHI'92. Monterey, ACM Press**:** 3-9. .

Maltz, D. and K. Ehrlich (1995). Pointing The Way: Active Collaborative Filtering. CHI'95. Denver, CO, ACM Press**:** 202-209. .

Mander, R., G. Solomon, et al. (1992). A 'Pile' Metaphor for Supporting Casual Organization of Information. CHI'92. Monterey, ACM Press**:** 627-634. .

North, M. M. and S. M. North (1993). An Information Exploration and Visualization Approach for Direct Manipulation of Databases. Proc. VCHCI'93. Vienna, Springer**:** 417-418. .

Resnick, P., I. Neophytos, et al. (1994). GroupLens: An Open Architecture for Collaborative Filtering of NetNews. CSCW'94. Chapel Hill, NC, Addison Wesley**:** 175-186. .

Rose, D. E., J. J. Borenstein, et al. (1995). MessageWorld: A new Approach to Facilitating Asynchronous Group Communication. CIKM'95 (Conf. on Information & Knowledge Management). Baltimore, MD**:** 266-273. .

Rose, D. E., R. Mander, et al. (1993). Content Awareness in a File System Interface: Implementing the 'Pile' Metaphor for Organizing Information. SigIR'93. Pittsburgh, PA, ACM Press**:** 260-269. .

Starr, B., M. S. Ackerman, et al. (1996). Do-I-Care: A Collaborative Web Agent. CHI'96 Conference Companion. Vancouver, BC, ACM Press**:** 273-274. .

VanBuren, D., P. Curtis, et al. (1995). The AstroVR Collaboratory, An On-line Multi-User Environment for Research in Astrophysics. Astronomical Data Analysis Software and Systems IV. R. A. Shaw, H. E. Payne and J. J. E. Hayes. **77**. accessible at http://www.stsci.edu/meetings/adassIV/vanburend1.html.