Dynamic Neighborhoods – Browsing the World Wide Web Together

Michael Merz, Konrad Froitzheim, Heiner Wolf
Ulm University, Germany

ABSTRACT

We introduce an original scheme that turns the World Wide Web (WWW) into a social place. Our vicinity-based approach is to show the Web's user space to the people surfing on the WWW. Being aware of other Web surfers, people browsing the Web may invoke synchronous communication by a single mouse-click – surfing the Web turns from a lonely affair into a joint experience. We present the basic theoretical concepts of our approach and give a short overview over CoBrow[1], a collaborative browsing service that implements the concepts presented in this paper. We also try to assess the social impact of services like CoBrow, which bring people together in a virtual world.

Keywords: collaborative browsing, neighborhoods, synchronous communication, user space, vicinities, WWW

1 INTRODUCTION

The Internet has become an indispensable database of knowledge and information and the World Wide Web has become the most popular tool for dissemination of information in the Internet. There have been numerous publications on how to collaborate and to communicate on the Web (see e.g. [9], [10]). However, thousands of people browsing the Web – possibly even the same Web site at the same time – are not aware of each other. The Web seems to be of a lifeless nature, which is actually wrong: the Web itself consists of static documents linked by hyper-links, but there are many people browsing Web pages and hopping from one page to another. Thus there is plenty of life on the Web, which is unfortunately hidden. Recently, efforts have been made to reveal these activities and to allow Web users to interact with each other synchronously.

1.1 Dynamic Directory Services

A straight-forward approach to the social Web are *dynamic directory services* (DDS; see e.g. [8], [13]). DDSs allow people to fill personal buddy-lists with friends, associates, etc. The DDS notifies registered users when somebody on their buddy-list enters the Web. Users may start synchronous communication, such as text-based chat, or video-conferencing, at any time with anybody on their buddy-lists who is on-line.

DDSs provide their users with a glimpse of the Web's lively nature. But people have to maintain buddy-lists. Therefore they pre-select their potential communication partners: they can only meet people they know already, people they have put on their buddy-lists. Moreover, only people who have registered with the same DDS may be added to the buddy-lists. Thus with using DDSs, there is no way of meeting new people by chance, like they meet in the supermarket in the real world.

1.2 Static Neighborhoods

Another approach to the interaction of Web users are *virtual meeting rooms* (VMRs). A VMR consists of a set of well-defined Web pages (see figure 1). Somebody browsing a page that is part of a VMR can see all the other people who are browsing pages belonging to the same VMR. What has to be specified in order to create a VMR is the set of Web pages where people can meet, but – in contrast to DDSs – not who can meet.

Hence, VMRs allow their users to meet new people, people they have never met before. And the users do not have to register explicitly with a service, they just have to enter a virtual room. However, the VMRs' view of the Web's user space is still very centralized: the creator of a VMR is the only one who can lend shape to it by specifying of which pages the VMR consists. Therefore people browsing a VMR's Web pages simultaneously are said to share a *static neighborhood*. Due to this static structure, people can meet each other only in predefined parts of the Web, but not anywhere on the Web. In order to meet other people, the users have to know a VMR and how to get there. Neither can they adapt their static neighborhoods to their individual needs, nor specify the people they want to see in the neighborhood. [12] gives a description of static neighborhoods and outlines an implementation proposal.

[1] CoBrow (Collaborative Browsing in Information Resources) is a project in the ”Telematics for Research“ framework of the European Union and the Swiss Department of Education and Science.

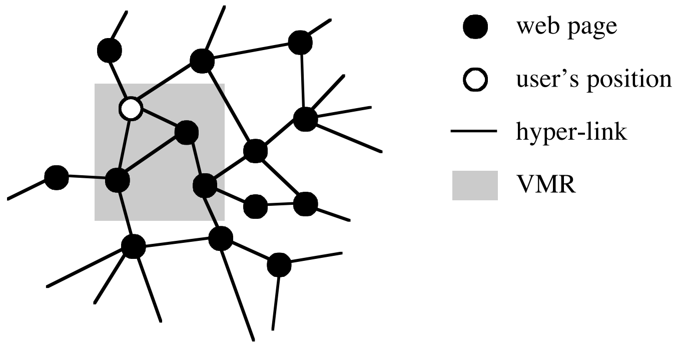


figure 1: example of a virtual meeting room (VMR);

1.3 Dynamic Neighborhoods

We propose a more general approach to user interaction on the Web, which comprises these two concepts, fills the whole range between them, and even allows for real chance meetings of people who do not know each other (almost) anywhere on the Web. This approach is based on *dynamic neighborhoods*, also referred to as *vicinities*. The vicinity of a user consists of a set of people – similar to static neighborhoods. Unlike static neighborhoods, our approach allows the users to select the people in their vicinities by specifying personal preferences. This leads to personal vicinities providing each user with a very personal view of the Web. Typically, each user's view of the user space, i.e. the vicinity, is different from the other users' views – even if the others are browsing the same page. In this approach the whole Web is considered a virtual meeting place.

In figure 2, we give an example of a user's vicinity. Depending on their personal preferences, the people browsing the Web can see all the others browsing the same page or any other page in the gray zone[2]. Following a hyper-link on the current page means changing the position on the Web. As a consequence, the personal view of the Web changes, too. In our example, the vicinity is just like a spotlight with its owner in the center. The user can see all the other people browsing the pages in the spotlight.

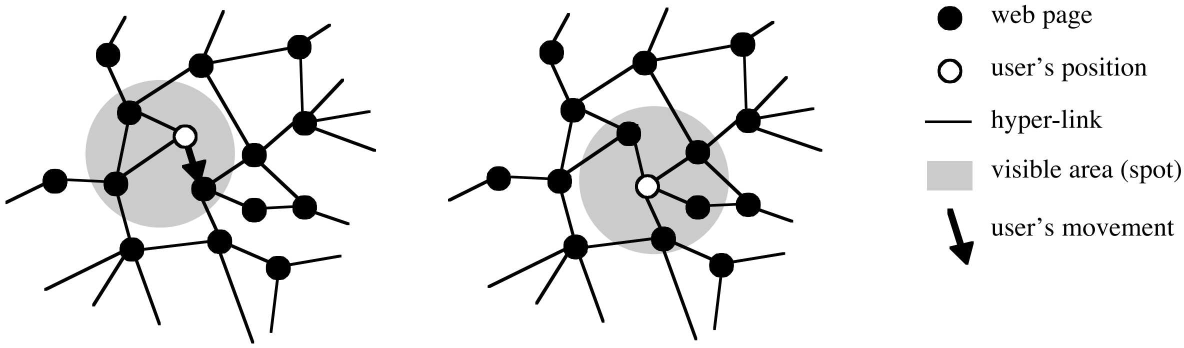


figure 2: example of dynamic neighborhoods with a user moving from one Web page to another;

1.4 Overview

The rest of the paper is organized as follows: First we introduce our concept of vicinities including metrics on the Web, and the visualization of the Web's user space. Section 3 presents a brief overview over CoBrow, a feature-rich collaborative browsing service, which implements the concepts introduced in this paper (see also [15]). We also try to estimate the potential commercial and social impact of collaborative browsing services, such as CoBrow. Finally, in 4 and 5, we summarize and present our future plans.

2 VICINITIES

Our approach to the social Web shows the bustling nature of the Web by displaying the Web's user space in addition to its document space. The basic concept of this approach are *vicinities*. A logical vicinity of a user consists of a set of users. Each user owns a personal vicinity. The vicinities of different people are usually different and depend strongly on the individual settings of their owners. All users can interact with all people in their vicinities. In contrast to the page-centered static neighborhoods, our approach is user-centered: users may configure the shapes of their vicinities on-the-fly by adjusting their personal settings.

---

[2] In figure 2 we assume that the user configured her/his horizon (see 2.2) accordingly;

To evaluate the vicinity of a user we need tools providing the logical distance to the other users on the Web – we need *metrics*. In this section, we outline the fundamentals of metrics and present some of the metrics used in the CoBrow service (see 3):

2.1 Definition

A set of elements S is a metric space (see e.g. [3]) if, and only if $\forall$ (x, y) $\in$ S × S $\exists$ d (x, y):

1. d (x, y) = d (y, x);
2. d (x, y) + d (y, z) ≥ d (x, z);
3. d (x, y) ≥ 0 ∧ d (x, y) = 0 ⇔ x = y;

In the subsequent sections we consider the people on the Web as the set of elements S and we compare methods to measure the distance between these people, i.e. metrics.

2.2 The Distance in (Hyper-) Space

As a first approach we define the distance between people on the Web by the *link-distance* between the closest pages they are browsing. The distance between two people browsing the same Web page is zero, the distance between two people browsing directly linked Web pages is one, etc. Non-negative numbers, weights, may be assigned explicitly to each hyper-link. Hence, the distance between two Web pages can be defined by hand, and shape may be artificially lent to the vicinities of the users: virtually long link-distances between directly linked pages may be pretended as well as a very close virtual neighborhood of pages with many hyper-links in between them.

We formally consider the Web as a directed graph W (P, H) with a finite set of vertices P, the Web pages, and a finite set of edges H, the hyper-links between the Web pages. Furthermore, we suggest the following definition of the link-distance D between the single Web pages, where x, y $\in$ P:

1. For two directly linked Web pages x and y:
   D (x, y) := w (x, y), if a weight w is explicitly assigned to (x, y) $\in$ H;
   D (x, y) := 1, otherwise;
2. D (x, y) := ∞, if there is no path through the Web from page x to page y;
3. D (x, y) := min { D (x, y), D (x, z) + D (z, y) | z $\in$ P };

With this definition of the link-distance D, the Web is a metric space according to 2.1. However, we do not want to measure the distance between Web pages, but the distance between people on the Web. With the help of the above definition of the distance D for Web pages, we define the distance d between two users u, and v:

d (u, v), d (v, u) := min { D (x, y), D (y, x) | user u browses page x, user v browses page y };

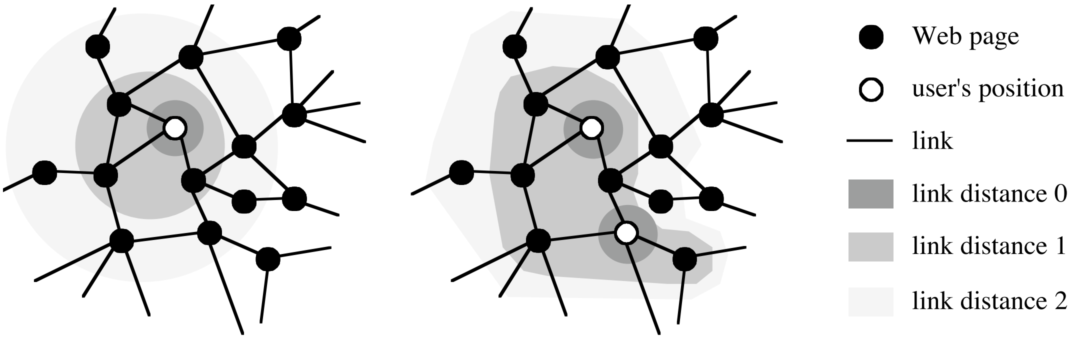


figure 3: pages in the visible area depending on the link distance, assuming each link has a weight w = 1; the shape of the visible area can be complex if a user is browsing several pages simultaneously;

As shown in figure 3 on the left, the shape of a user's vicinity, which is evaluated according to the link-distance metric, depends on the user's settings: the vicinity consists of people browsing Web pages in the dark gray, the medium gray, or the light gray zone. A user may choose how far to see by specifying a *horizon*, i.e. a real number representing the maximum distance where other people can still be seen. Setting the horizon to ∞, a user would theoretically be able to observe all user activities on the whole Web – which is obviously not feasible in practice. The shape of the vicinity changes, i.e. enlarges, when the user opens an additional browser window as shown in figure 3 on the right.

For now we consider the link-distance metric as the basic metric that is not used solely, but in combination with other metrics only[3]. In other systems, however, the link-distance metric may be equal to any other metric, such as the user interest metric (see 2.3).

[3] The CoBrow service's user tracking (see 3.1.3) relies on the people's positions on the Web.

2.3 The Individual User Distance

Our major goal is bringing people together, people who have the same or similar interests. The most interesting metric in this regard is obviously what we call the *user interest* metric: the more interests two people on the Web have in common, the closer they are. We are working on two different approaches to find people with similar interests:

1. The users provide the system with their properties vector, e.g. their hobbies, their age, their shoe size, etc. Additionally, they specify to which degree the respective property applies by providing a number (0.0 = "does not apply at all", ..., 1.0 = "applies exactly"; if no number is provided for a property, the default value is 1.0). In a second step the users specify who they want to meet by defining their preferences vector, e.g. that they want to meet only people who speak ancient Greek, who have green hair, etc. Again they can additionally define how important the single preferences are (0.0 = none of the vicinity members must have this property, ..., 1.0 = all vicinity members must have this property; if no number is provided for a preference, the default value is 1.0).
   The distance between two users according to this metric is defined as the distance between the preferences vector of the first user and the properties vector of the second one. There is no designated way to calculate the distance between the properties vector and the preferences vector; however, we propose the Euclidean distance (e.g. [3]). As a result, the distance between two users according to their properties/preferences yields a real number.
2. A second approach to the matching of users with similar interests is to analyze the Web pages they are browsing. In other words, the more similar the contents of the Web pages the users browse, the closer the users according to this metric. Analyzing the contents of Web pages as well as evaluating the similarity of two documents, are major issues of Web mining and beyond the scope of this paper (see e.g. [6], [14]).

By determining the distance between a designated user and the other people on the Web according to personal preferences, the vicinity-based approach is a powerful search engine for people: the whole World Wide Web can be searched for people matching individually configurable properties, be it the names of the people, their e-mail addresses, or any other personal information; vicinity servers (see 3.1.1) watch out for other people on the Web according to the respective properties – similar to common search engines searching the Web for documents according to keywords (e.g. [7], [17]). Moreover, vicinity servers allow to specify the significance of each property for the search for other people. The properties of people in a vicinity do not have to match exactly to the designated user's preferences: even "fuzzy" queries are possible. However, the number of people in a vicinity may be reduced by specifying preferences: the more exact and restrictive the preferences, the fewer matches, i.e. the fewer users in the vicinity. In other words, browsing the Web's user space according to individually configurable preferences is similar to browsing the Web's "conceptual" document space (see [11]).

2.4 Disappearing from a Vicinity

In order to set up synchronous communication all participants must be available. Clients of DDSs (e.g. [8], [13]), for instance, just have to be online to be able to participate in synchronous communication. In a vicinity-based system, however, the communication partners additionally have to be visible in each other's vicinities. Thus it is hard to set up synchronous communication in such highly dynamic environments, with all the people appearing on and disappearing from each other's vicinities very quickly.

We allow people to specify the point of time when they become invisible for other people. When a user leaves the vicinity of another user, e.g. by following a hyper-link, he remains still visible for a certain period of time. The point of time when the user actually becomes invisible for the other is just another preference to be configured personally by the user. We define the *time* metric with the help of this preference: the distance between two people according to the time metric is the difference of the current time and the time the user actually disappeared from the other's vicinity. At the point of time when the user wants to become invisible for the other user the distance is ∞.

As a consequence people may specify how long they want to remain visible for the others after actually leaving the vicinities of the others. And people can still invoke or participate in synchronous communication although they have left the virtual meeting place already.

2.5 Simulating DDSs and VMRs

The vicinity approach is generic and therefore comprises various mechanisms to model the Web's user space – including both DDSs and VMRs.

In the narrower sense a DDS is not a real Web service: users have to log onto a central server, which in turn matches the logged users' preferences and properties. A service like that can easily be simulated with dynamic neighborhoods by specifying the personal preferences respectively; to find Mr. Ford Prefect on the Web, for instance, the realname preference would be set to "Ford Prefect". Buddy-lists may be simulated by maintaining lists with the buddies' names. The vicinity server sends notifications to all interested users if somebody on their buddy-lists enters the Web.

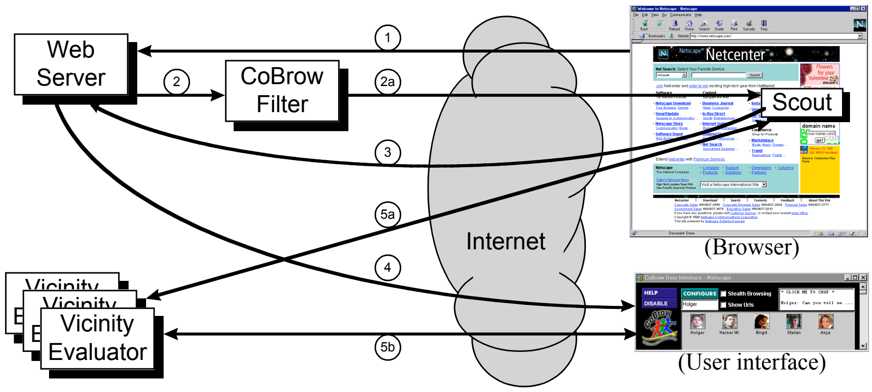
VMRs may also be simulated easily using vicinities: Web masters may define VMRs by setting the link distance attributes of the Web pages making up the VMR to zero. Hence, all people browsing any of these pages simultaneously can then see each other – no matter how many hyper-links are actually between the pages. This is exactly what VMRs allow their users to do.

In other words, we have shown that DDSs as well as VMRs provide only a subset of the functionality provided by the dynamic neighborhood approach. Both of them may be simulated by setting the respective preferences accordingly.

3 COBROW – A SHOWCASE IMPLEMENTATION OF DYNAMIC NEIGHBORHOODS

In this section we outline the CoBrow service, which visualizes the Web's user space. We show the system architecture, sketch the internal communication protocol and discuss the interplay of the components. Any further information on the implementation or the system itself may be obtained at the CoBrow Web pages (see [4], [5]).

CoBrow is a complete collaborative browsing service with built-in communication facilities. It is based on vicinities as introduced in the previous section; however, it supports also static neighborhoods. Explaining all features in due detail is beyond the scope of this paper. Instead, we focus on vicinity management related issues. Neither do we consider the group and conference management here, nor the communication infrastructure.



(1) Browser requests Web page
(2a) Web server sends original Web page to filter
(2b) Filter inserts Scout and forwards Web page
(3) Scout requests user interface
(4) Web server sends user interface
(5a) Scout reports user position,
(5b) user provide information for the system with help of the user interface

figure 4: the CoBrow functional components (simplified);

3.1 System Architecture

The CoBrow service is a distributed system and therefore consists of many functional components. The *Vicinity Evaluators* represent the system's kernel; they provide the client modules, i.e. the *user interfaces*, with information on the users' vicinities. Auxiliary components, the *Scouts*, collect the information on the Web and on the users the Vicinity Evaluators need to calculate the vicinities (see figure 4).

3.1.1 The Vicinity Evaluator

The *Vicinity Evaluator*, also referred to as the CoBrow server, receives information from the user interfaces (see 3.1.2) and from so-called Scouts (see 3.1.3). Based on this information, the Vicinity Evaluator evaluates the users' vicinities according to metrics (see 2) and provides the user space dimension of the Web's virtual world.

Actually, the Vicinity Evaluators are probably the most complex CoBrow components. This is not only because they have to deal with vast amounts of data. Many interconnected Vicinity Evaluators may coexist on the Web providing a distributed CoBrow service. They exchange information on Web users and on the Web's structure. However, the inter-Vicinity Evaluator communication is entirely transparent to CoBrow clients: virtually, there is only one CoBrow service, which is actually provided by many interactive, collaborating Vicinity Evaluators.

Since the Web is organized in parts stored on many Web servers, a Vicinity Evaluator is usually responsible for providing only the user space of a part of the Web. Although in typical scenarios one Vicinity Evaluator administers the user space of one Web server, one Vicinity Evaluator may also provide the user space information for a whole group of Web servers.

In other words, the main task of a Vicinity Evaluator is to find the persons to appear in the users' vicinities according to metrics. The current implementation of the CoBrow system employs the space metric and the time metric to calculate vicinities. The Vicinity Evaluator also supports simple synchronous communication, such as plain text, without group management, though.

3.1.2 The User Interfaces

The *User Interface* visualizes the information calculated by the Vicinity Evaluators, i.e. the Web's user space. Moreover, the user interface allows the users to provide the Vicinity Evaluator with their personal properties and preferences.

On the one hand, visualizing the Web's user and document space means to handle and to display vast amounts of information; on the other hand, all this information has to be displayed on the users' tiny screens. Therefore the major challenge is to clearly arrange plenty of information on strongly restricted screen space. Another problem is imposed by the typically low bandwidth data links between the Vicinity Evaluator and the user interface running at the clients' home PCs: first, all the vicinity data has to be transmitted through these channels; second, and even worse, the user interface has to be downloaded before it can be used.

We offer two different user interfaces, both of them Java applets for the users' convenience: neither do users have to download anything explicitly, nor to install anything; the applets are downloaded and started automatically when a user browses a Web page stored on a CoBrow enabled Web site. They share the basic communication modules and layout components. Each of the user interfaces is tailored to special application scenarios: the plain user interface with minimum requirements in terms of processor load, bandwidth in the communication channel, and display resolution; and the virtual reality mark-up language (VRML)-based user interface, which addresses future applications supported by high-end hardware. These two approaches differ in their visual appearance, processor load, and functionality. However, both of them allow for visualizing static and dynamic neighborhoods.

3.1.2.1 The Plain User Interface

The plain user interface (see figure 5) is the lean one, tailored to the needs of common users, who are connected to the Internet via Internet Service Providers (ISP). Its size is about 60 kBytes, which takes up about 20 seconds to download with a 28.8 kbps modem; it requires initially about 500 x 150 pixels of screen space to be displayed; however, this size may be changed of course.

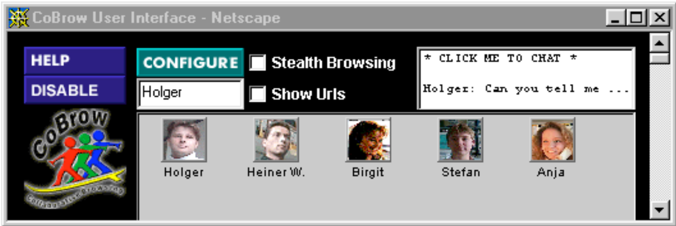


figure 5: the plain user interface;

This user interface lines up the icons of all neighbors in the vicinity along with their nicknames. The URLs the neighbors are browsing may also be displayed on demand. A configure dialog allows to modify the personal properties and preferences. With the help of an additional advanced configure dialog, users may even define very personal new properties, such as their hat sizes. A simple mouse click on another person's icon establishes a communication channel to the respective user. What kind of communication is started depends on the hard- and software on the users' machines.

3.1.2.2 The Advanced User Interface

Basically, the advanced user interface (see figure 6) provides the same functionality as the plain user interface. However, due to its capability to visualize virtual three-dimensional worlds, this user interface provides more space to represent the Web's user space, i.e. a third dimension. Appropriate algorithms (e.g. [2], [16]) provide the users with a three-dimensional view of the part of the Web they are browsing. The clients can see the other Web pages in the vicinity of the page they are currently browsing and they can see the other people, of course, who are browsing these Web pages in the neighborhood. Unlike the plain user interface, the advanced user interface does not simply list the icons of all people. Instead, it arranges the icons next to the Web pages the respective persons are browsing. The Web pages themselves are represented by thumbnail images, the hyper-links between the Web pages by threads connecting one thumbnail image with another. A click on a person's icon invokes synchronous communication with this user, just like with the plain user interface; a click on a thumbnail image opens a new browser window with the respective document in it. Plain text chat of all clients in the vicinity is presented as a text track, which flows from the front to the back of the 3D-scene.

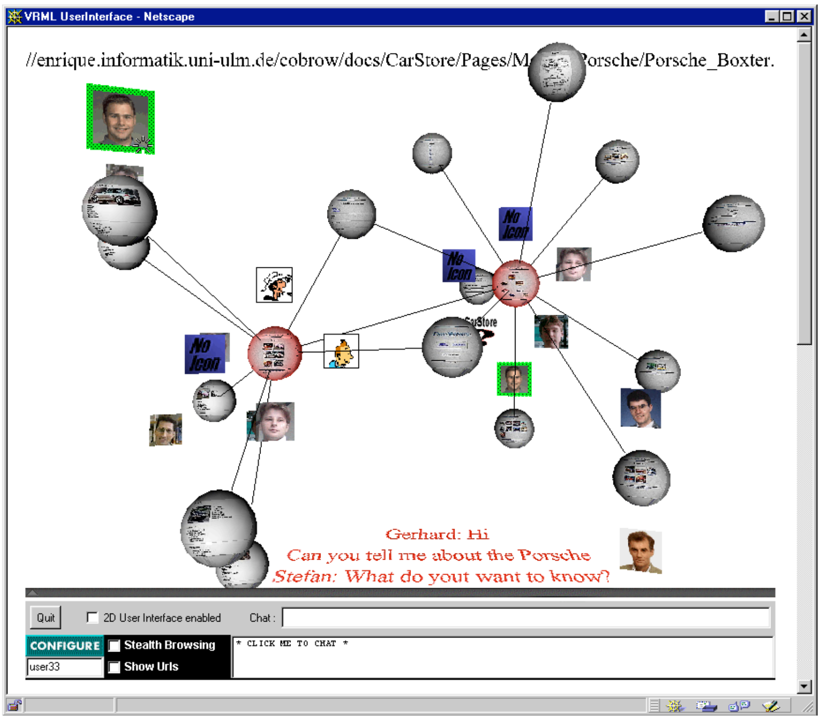


figure 6: the advanced user interface;

Due to this sophisticated visualization this user interface craves for resources: in addition to the Java engine, the VRML player has also to be initialized before the user interface may be used. Then, the Java classes of this user interface are bigger – although they share a lot of code with the plain user interface. And finally, rendering three dimensional worlds takes up more processor power than simply lining up icons and names. Therefore this user interface remains reserved for the clients with high-end hardware and high-speed Internet connections – for the moment at least.

3.1.3 User Tracking

We have explained how the Vicinity Evaluators calculate vicinities, and how this information is displayed by the CoBrow service. But where does all the information needed for the calculation of vicinities come from? We need software that tracks people on their way through the Web[4] and reports any action they take to a responsible Vicinity Evaluator.

This software is called *Scout*. The Scout is a tiny Java applet that is inserted into every[5] Web page sent by the server. The Scout applet opens an HTTP connection to the respective Vicinity Evaluator and reports the user's position in the Web, i.e. the URL of the page this user is browsing. It also notifies the CoBrow server when the user leaves a Web page.

The *CoBrow filter*, i.e. a simple proxy server, parses all Web pages sent by the Web server. It inserts the Scout applet into every Web page dynamically, by adding an HTML applet tag. Since the applet tags are inserted automatically, all HTML-documents stored on a Web server do not have to be edited manually but remain untouched.

3.2 The CoBrow Protocol

Plenty of data has to be exchanged between the functional components of CoBrow to provide the clients with user space information. This is especially true for the CoBrow service, which is a distributed Internet service. To convey all user space related information we propose a special *User Space Information Transfer Protocol* (USITP) in addition to the Hypertext Transfer Protocol (HTTP), the Web's protocol for document space related information interchange.

The current implementation of the CoBrow service encapsulates USITP in HTTP, according to "form-url-encoded" requests (see [1]). The most important advantage of this technique is its compatibility with components augmenting the Web, such as proxies and firewalls: there is no need for installing or updating any software or hardware facility in order to run a CoBrow software component. However, we hope to succeed in standardizing a protocol for user space information interchange. This is important with regard to future Web browsers visualizing the user space in addition to the document space and with regard to intranets. We hope either to integrate our protocol seamlessly into HTTP, or to create a new protocol standard coexisting with HTTP.

---

[4] For privacy reasons users may disable the tracking, of course.

[5] On demand designated Web pages, or even whole directories may be excluded from user tracking.

CoBrow handles very volatile user space information, such as the positions of people on the Web. To properly compute and represent the vicinities, all information updates have to be passed between the software components with reasonable delays. HTTP's request/response architecture is not suited for the interchange of volatile user space information. The system asks for updates, for instance, i.e. responses without preceding requests. USITP adds the functionality needed, such as multi-request-sessions, time-outs, periodical information updates, subscription for information, and many more.

3.3 Results

In order to test our system and to assess the user acceptance, we installed the CoBrow service on our department's model railroad server[6] in October 1997. As about 1,000 people a day visit our interactive model railroad, we consider this server to be the ideal site for people to meet each other by chance, people who have never met before, who are interested in our interactive model railroad, in CoBrow, or both.

It has shown that users are excited about seeing each other in their vicinities. Although most users are not equipped with Internet based conferencing tools, they use the built-in chat frequently. However, chat is apparently too cumbersome to keep people in the system for a longer time: while some users try very hard to keep the communication going, others are simply too drowsy to communicate in this admittedly old-fashioned way.

However, the planned user trials (see 5) will provide us with more quantitative information on the acceptance of our new Web service, on its usability, etc.

3.4 Perspectives and Applications

Although it is hard to assess the future perspectives of an original Web service, such as CoBrow, we are sure that systems visualizing the Web's user space will become an integral part of the future Web. Just because this is the natural way of representing the Web: not only the mesh of lifeless documents is displayed, but also the people browsing the Web pages, people eager to communicate. Potential applications include Internet Cafes, meeting points on non-commercial Web-sites, virtual department stores, and many more. The perspectives of services like CoBrow are limited only by imagination and we evaluate only a few examples here:

- *Tele-shopping* can be made much more interactive: all the customers of a company are aware of each other, just as if they were in the neighborhood's shopping center, or in the shopping mall downtown. Virtual malls might even be modeled specifically to improve the impression of a virtual shopping tour.
- Internet technology is the basis for corporate data networks, so-called *intranets*. In enterprises, teams in both R&D and production will be able to form ad-hoc teams to solve problems and improve products.
- People surfing the Web *just for fun* can meet by chance in virtual malls, on homepages, or any other Web pages. They may go to Cybercafes together, discuss with other people interested in the same topics, etc.

Since CoBrow supports and strongly suggests the interaction of people on the Web, specifically of people who have never met before, it will revolutionize the "Netiquette": new social protocols ruling the virtual life in a virtual world will be necessary – similar to the social protocols ruling the living together of people in the real world.

To explore carefully the social impact of new services, such as CoBrow, we reinforced our project team by experts in psychology. They will especially evaluate the planned user trials (see 5).

4 SUMMARY

We have shown the basic model, implementation concepts, and first results of a new service to represent the Web's user space. The CoBrow system, implementing the concepts presented in this paper, is under evaluation on our test Web site. People accessing this site are aware of each other: they can see other people browsing the Web pages in their vicinity; they can even see these people hopping from one page to another. People in a vicinity may communicate with each other by means of the built-in ASCII-chat, or they can easily set up video-conferences by clicking on another person's icon on the user interface – presuming the respective communication partners are provided with the appropriate equipment. The system is updated permanently and the project team always watches out for new tools, features, and concepts to be integrated into the software. However, the success of the service depends crucially on the installation of the server software on as many Web sites as possible. Hence, all readers are invited to download our free software package[7] and to enhance their Web sites with this new service.

5 FUTURE WORK

As mentioned earlier, our vicinity-based collaborative browsing concepts introduce an original approach to bringing together people sauntering through the Web's virtual universe. There are numerous issues left for further research and we subsequently outline some of our plans to give an idea of what will be going on in the near future:

[6] Everybody is welcome to visit our CoBrow demonstration site at `http://rr-vs.informatik.uni-ulm.de/rr/`

[7] The server software may be downloaded from `http://www.cobrow.com/`

- We think that the one of the most urgent issues to be addressed is the enhancement of the system's usability. In other words, we want to adapt the functionality and the handling of the whole CoBrow service to its users' needs. To hear about these requirements at firsthand, we, together with our industrial partners, have set up various *user trials* for the near future.
- The installation of the CoBrow server still requires skills in system administration. Hence, we will *ease the installation* process in order to succeed in distributing the CoBrow service widely and we hope to get some helping feedback from the user trials in this regard, too.
- With the current CoBrow version the user interface software, i.e. a Java applet, is downloaded to the users. However, an approach to the speed-up of the CoBrow service's downloading and initialization process at the client side, and also to a more flexible and more personal configuration of the user interface, would be a *stand-alone user interface*, which resides permanently at the client – a conventional program, e.g. a Java application. These benefits come at the price of a more complex installation and configuration process.
- We have introduced mechanisms to calculate dynamic neighborhoods: a vicinity server computes the members of a vicinity according to a metric, using the users' settings, i.e. the users' preferences and properties. To generalize the vicinity evaluation we are about to introduce personal user agents, so-called *meetlets*. A meetlet is a small piece of code, e.g. a Java applet, which is executed on the vicinity server. Each user may create a personal meetlet to entirely define the users' vicinities, or to modify, or to refine the vicinity lists generated by the vicinity server.

ACKNOWLEDGEMENTS

This work was performed in the framework of the CoBrow project, which is partially funded by the European Community under the program "Telematics for Research" and the Swiss Department of Education and Science. The authors would like to thank their colleagues at Lancaster University, UK, Ulm University, Germany, and ETH Zurich, Switzerland for their contributions.

REFERENCES

[1] T. Berners-Lee, D. Connolly. *Hypertext Mark-up Language – 2.0*. IETF RFC 1866, 1995.

[2] H. Bönisch, S. Fiedler, K. Froitzheim, P. Schulthess. "A VRML-based Visualization of User-Vicinities in the WWW". *Proceedings of ATSMA 6th International Conference on Telecommunications – Analysis and Modeling*. Nashville, TN, 1998.

[3] I. N. Bronstein, K. A. Semendjajew. *Ergänzende Kapitel zum Taschenbuch der Mathematik*. Teubner, Leipzig, 1990.

[4] The CoBrow Project. *Collaborative Browsing in Information Resources*. ETHZ, Zurich, Switzerland, 1997. Software on-line[8].

[5] The CoBrow Project. *CoBrow – You are not alone on the web*. Ulm University, Germany, 1998. Software on-line[9].

[6] R. Cooley, B. Mobasher, J. Srivastava. Web Mining: Information and Pattern Discovery on the World Wide Web. Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence. 1997.

[7] Excite, Inc. *Excite*. Redwood City, CA, 1995-98. Software on-line[10].

[8] Firefly Network, Inc. *Personalize your network*. Cambridge, MA, 1997. Software on-line[11].

[9] R. Frivold, R. Lang, M. Fong, "Extending WWW for synchronous collaboration". *Proceedings of the 2nd International World Wide Web Conference*: pp 69-75. Chicago, IL, 1995.

[10] B. R. Horstmann, T. Trevor, J. "The World Wide Web as enabling technology for CSCW: The case of BSCW". *Computer-Supported Cooperative Work: Special issue on CSCW and the Web*, Vol. 6. Kluwer Academic Press, 1997.

[11] R. E. Kent, C. Neuss. "Web Conceptual Space". *Proceedings of the World Conference of the Web Society WebNet'96*. San Francisco, CA, October 1996.

[12] J. Kothardi, E. Grossman, S. Mehrotra. "Neighborhoods: A Framework For Enabling Web Based Synchronous Collaboration And Hierarchical Navigation". *Proceedings of the 30th Hawaii International Conference on System Sciences*: pp. 666-675. Hawaii, January 1997.

[13] Mirabilis Ltd. *ICQ – World's Largest Internet Online Communication Network*. Tel-Aviv, Israel, 1997. Software on-line[12].

[8] `http://www.tik.ee.ethz.ch/~cobrow/`

[9] `http://www.cobrow.com/`

[10] `http://www.excite.com/`

[11] `http://www.firefly.net/`

[14] M. Pazzani, J. Muramatsu, D. Billsus. ”Syskill & Webert: Identifying interesting web sites“. *Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access*. Portland, OR, 1996.

[15] G. Siedler, A. Scott, H. Wolf. ”Collaborative Browsing in the World Wide Web“. *Proceedings of the 8th Joint European Networking Conference*: pp. 122-1 - 122-8. Edinburgh, UK, 1997.

[16] A. M. Wood, N. S. Drew, R. Beale, R. J. Hendley. ”HyperSpace: Web Browsing with Visualization“. *3rd International WWW Conference, Poster Proceedings*: pp. 21-25, Darmstadt, Germany, 1995.

[17] Yahoo!, Inc. *Yahoo!*. Santa Clara, CA, 1994-98. Software on-line[13].

[12] http://www.mirabilis.com/

[13] http://www.yahoo.com/