

# Using Agents to Personalize the Web

**Christoph G. Thomas**

HCI Research Division  
GMD FIT  
D-53754 Sankt Augustin, Germany  
+49 2241 14 2640  
christoph.thomas@gmd.de

**Gerhard Fischer**

L<sup>3</sup>D - Center for LifeLong Learning and Design  
University of Colorado at Boulder  
Boulder, CO 80309-0430, USA  
+1 303 492 1502  
gerhard@cs.colorado.edu

## ABSTRACT

Users build personal information spaces (stored as bookmarks, hotlists, or as a personal page of links) as their WWW-subset and interface to access the World-Wide Web. As the WWW is a living "creature" that evolves and grows permanently, users have to take care that their personal information spaces can be kept manageable and up-to-date.

Our prototype system BASAR (Building Agents Supporting Adaptive Retrieval) provides users with assistance when managing their personal information spaces. This assistance is user-specific and done by software agents called web assistants and active views. Users delegate tasks to web assistants that perform actions on their views of the WWW, on the WWW itself, and on the history of all user actions

In this paper, we discuss aspects of the design-evaluation-redesign cycle of BASAR by focusing on questionnaires, assessment studies, and system evaluations.

## Keywords

agents and agent-based interaction, evaluation of agent-based interfaces, World-Wide Web, information overload, personal information spaces

## INTRODUCTION

The World-Wide Web (WWW) has emerged as a new type of information space. Its lack of central control mechanisms leads to many new interesting features but at the same time has the potential danger that users can drown in irrelevant information. Being lost in space and overloaded with information [9] are two problems users confront: there is more information out there than a single user can manage. The potential benefits of the WWW will not be realized if users cannot retrieve information easily and efficiently.

They need active and user-specific support when they do not know whether potentially useful information exists, where this information is located, how to retrieve this information when it is found, and how to use the information when it is retrieved. Thus, to assist users in making full use of large information spaces such as the WWW, information *access* methods need to be complemented by information *delivery* methods.

The principal techniques that have long served users to retrieve information are browsing and searching. There are a few tools that help users to search and browse the WWW, the most popular being search engines and bookmarks (from NETSCAPE). Search engines support users to locate desired information. Bookmarks provide a fast and easy access to often visited pages. Both are invaluable tools when searching and browsing the web. Moreover, bookmarks allow users to personalize the web, i.e. they create their own personal information space [1] as their "image" of the WWW.

As the WWW is a living "creature" that evolves and grows permanently, users have to take care that their personal information spaces can be kept manageable and up-to-date. This is not an easy job because users have to find the right balance between adding new bookmarks, deleting unused or seldom used bookmarks, and updating existing bookmarks. For example, users often struggle with old or invalid bookmarks and have no idea whether their bookmarks still represent a valid image of the WWW.

Our prototype system BASAR (Building Agents Supporting Adaptive Retrieval) provides users with assistance when managing their personal information spaces by (a) supporting information link updating; (b) reducing the number of links by deleting old, never used, or seldom used information links; (c) supporting the growth of the view by using tools such as search engines more efficiently, and, finally, (d) supporting the relocation of information by referring to the history. Software agents called *web assistants* provide user-specific assistance for these tasks based on a user, task, and domain model.

In this paper, we discuss the design-evaluation-redesign cycle of BASAR by focusing on the role of bookmarks and introducing our concept of active views. Then, we present some examples of web assistants followed by an outline of our technical framework for implementing BASAR. Finally, we report on assessment studies and discuss the role of agents in BASAR. We finish the paper with future directions of our research on using agents for personalizing the web.

### **BASAR: SUPPORTING PERSONAL INFORMATION SPACES**

BASAR is a prototype of an extension to the WWW that embodies the following characteristics:

- **Active views:** BASAR provides users with a concept for creating personal information spaces along a semantical meaning that is independent of the WWW-viewer. An active view is defined as a set of bookmarks belonging to the same header in combination with a set of web assistants attached to it:  $\langle \text{active view} \rangle = \langle \text{bookmark header} \rangle + \langle \text{bookmark links} \rangle + \langle \text{web assistants} \rangle$ .
- **Web assistants:** BASAR provides users with an environment for delegating tasks to agents called web assistants that actively support them in locating, relocating, and filtering information. The assistants are responsible for keeping an active view manageable by adding, updating, and removing information links.
- **Usage profile:** BASAR builds a model of the user (preferences, interests, and tasks) using both explicit (asking the user) and implicit (observing the user) modeling techniques. This enables the web assistants to adapt to the user.

In BASAR, users deal with active views, web assistants work on these views, and the usage profile makes the assistance specific to each user. A more elaborate description of the concepts of agents and usage profiles in BASAR can be found in [11].

#### **Questionnaire about Bookmarks**

The development of active views is based on empirical studies of how users personalize their Web space. As bookmarks are the most popular mechanism to remember links, we distributed a questionnaire with 16 questions about the use of bookmarks (for the complete list of questions, see <http://zeus.gmd.de/i3/mmk/basar/bookmarks.html>). The most important results (based on 42 replies) are:

Using the WWW is a periodic task users perform several times a week. When new relevant information sources are found, their links are stored as bookmarks. Most people add bookmarks several times a week. Bookmarks are used to store often-used links or seldom-used links; two-thirds of the respondents use the bookmark feature as a repository for links that may be of interest later on. Some people even "collect links like squirrels collect nuts".

Introducing headers for a list of bookmarks is done by two-thirds of the respondents, most of them having between 20 and 50 links collected in their bookmark list distributed to up to seven headers. With the growing number of bookmarks, the question of which links are still valid becomes critical: old or invalid links still exist in a majority of cases.

Browsing and searching are the two most commonly used techniques to find information, but other sources, such as email from friends and colleagues or hints in magazines and newspapers, are becoming more important.

The questionnaires provided empirical evidence for the fact that people use tools such as bookmarks to create personal information spaces from public information spaces. This is becoming a ubiquitous activity. At the same time, people ask for better and more active support to manage their information spaces. Ideally, such support should consider the following dimensions.

- **Usage:** User's personal information is described by the information that has been selected, its usage, and by the actions that have been performed. Supporting the use of bookmarks requires assistance to identify at least "no more interesting links" (through history) and the invalid links (through checking).
- **Increase:** Empirical evidence shows that personal information spaces never or very seldom shrink; they grow by chance (when getting links by email, finding links in journals, or by surfing) and by intention (searching for keywords). A personal information space is a dynamic object that permanently expands in several directions: the total number of links, the number of headers, and the number of relevant information links for a specific header increase. In other words, with the growth of the personal information space, the request to manage it becomes more and more important.
- **Structure:** A personal information space can be structured by defining headers and by ordering relevant links according to these headers. This can be compared to the directory structure of a UNIX filesystem. With the increasing number of total links, structuring, which is a creative process that has to be done by the user, becomes more important. Structuring the personal information space is an essential task to avoid losing control over that space.
- **Validity:** A user's personal information space is a subset of the WWW. However, it may contain both valid and invalid information (links). Users do not know for sure which information is still valid and which is not. This can only be verified by checking it, i.e., selecting each link.

- **Sharing:** Users tend to distribute their personal bookmark lists to other people.

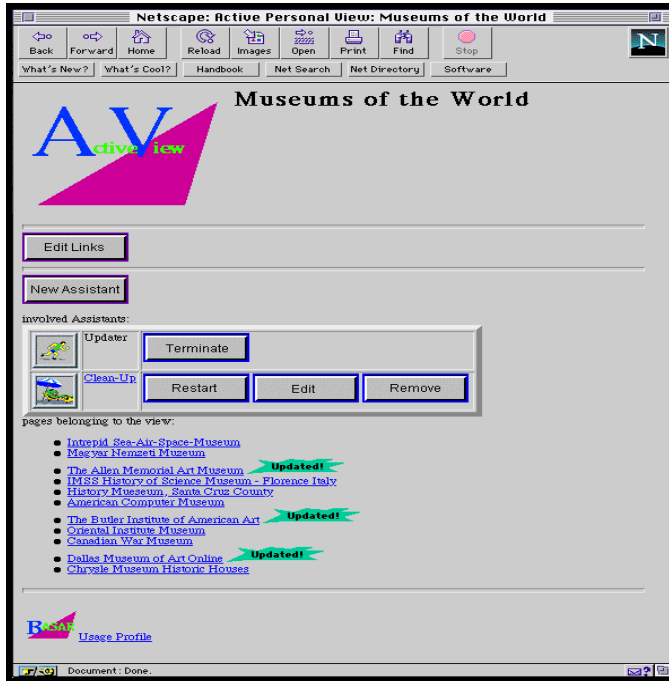


Figure 2. Active View "Museums of the World"

Attached to the view "Museums of the World" are two web assistants, an „Updater“ Assistant that checks the WWW pages, and a "Clean-Up Assistant" that periodically looks for unused or invalid links according to the usage profile.

Our concept of active views supports most of these dimensions. Active views are an active repository for a user's access to the WWW. The basic tasks that have to be supported when dealing with active views are *adding*, *updating*, and *removing* information links. For example, active views can (1) make suggestions to add information links to a view if these links have been visited often by the user in the past, (2) notify the user when information has been updated, or (3) suggest removing links that have become invalid. An active view is user-specific. By being part of the usage profile, it allows analysis of the user's actions on that view, for example, searching for, deleting, or adding links.

### Some Examples

In this section, we describe examples of the Web interface of BASAR (for more examples, see <http://zeus.gmd.de/projects/basar.html>). New views can be created or existing ones can be edited by naming the view and allocating an initial set of links belonging to that view. Once an active view has been created (see Fig. 2), the question is how to refine or extend that view. This task is, according to our questionnaire, done by at least two-thirds of WWW users periodically. One usual way is to call one (or more) search engines that are requested to

return links relevant to the given search key. It is then the user's task to filter the result by hand.

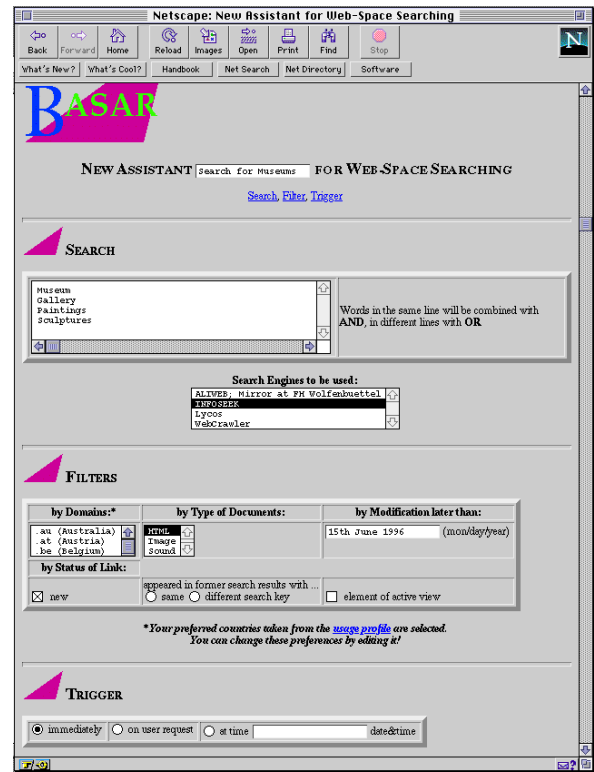


Figure 3. Web Assistant for Search

This web page provides users with a common interface to the most popular search engines. The web assistant will call the selected search engines until reaching success. The „Filters“ section presents a table of the user's preferences on how to sort the search results.

The empirical tests we conducted with WWW confirmed that users have difficulties in making efficient use of search engines. Some of the problems we observed were the accessibility of search engines, the validity of the results, and multiple occurrences of links if more than one search engine has been used (for a complete discussion, see [11]).

These observations led to the development of a search assistant that specifically supports that task, see Fig. 3. The invocation and the results of a search process are stored in the usage profile together with the actions the user performed on these results. The web assistant evaluates the success of a specific search and uses it later for another search task. The extracted information is used when the user calls the web assistant again. If a search engine returns a result, the web assistant filters all the information links according to the usage profile, for example, thereby noticing that the user is most interested in information links coming from .edu or .com server sites.

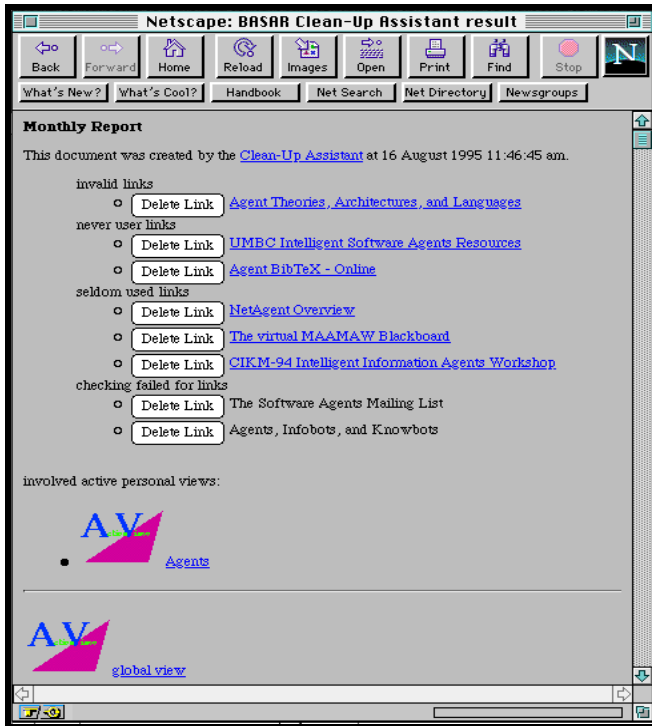


Figure 4. Clean-Up Assistant Results

Screen hardcopy of the monthly report of the web assistant "Clean-Up," which has been attached to an active view called "Agents." It provides the user with a facility to easily delete links that are part of the active view that have become invalid, have never or very seldom been touched by the user, or whose checking has failed due to too much net traffic, for example.

### CONCEPTUAL FRAMEWORK

Two main areas of research have influenced the work presented in this paper. The first area of research involves *indirect management* [3] as the fundamental model of interaction integrating web assistants as active software that supports users in doing their tasks at hand. Both users and web assistants initiate communication, monitor events, and perform tasks instead of having unidirectional interaction via commands and/or direct manipulation.

The second area of research is *information retrieval* and information filtering. In the WWW, search engines are the most popular tools to support retrieval and filtering. Another area has been named "collaborative information filtering," a technique to support information consumers in finding relevant information by making use of what others have already found and evaluated [6]. For example, HOMR [10] is a collaborative information-filtering system based on learning agent technology [5].

In contrast to search engines, systems such as HOMR build a user profile, called an interest profile, and make personalized recommendations based upon values assigned by other people with similar tastes. Such systems can be

used for any database. But they do not contain WWW-specific knowledge, analyze user's dialog history, or build a user-specific usage profile that supports the managing of WWW personal information spaces. The same holds for commercially available products such as WebCompass (arachnid.qdeck.com/qdeck/demosoft/webcompass\_lite/) or WebArranger (www.cesoft.com/info/webarrangerinfo.html). Both tools provide watchdog-like agents to observe a set of Web pages.

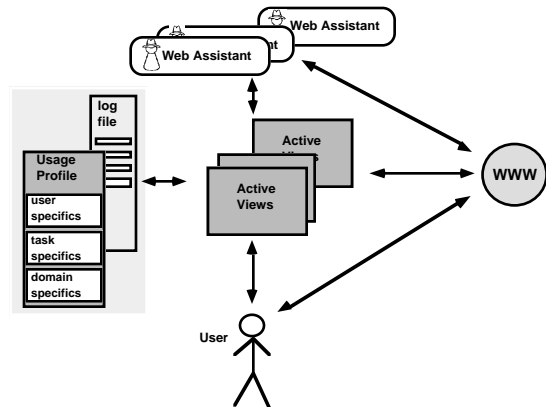


Figure 5. Conceptual Framework

Web assistants have access to both the functionality and the resources of the WWW. Also, web assistants have to communicate with the user. This is done either through an active view or via email, for example.

Web assistants in BASAR are small, semi-autonomous software programs able to act and react, to reason, to perform a task, and to communicate and cooperate [11]. They support adaptive filtering, creating active views, and locating and accessing relevant information. They have knowledge about the network, such as location of search engines, available servers, time zones, or different types of server sites such as *.com*, *.edu*, *.de*.

The user interacts with web assistants through email or through web pages within active views. These web pages inform the user about all active assistants with a short description of their tasks. They also provide the user with functionality to create active views, to edit the usage profile, and to create assistants through an editor. BASAR has been implemented under X-Windows in VisualWorks 2.0. The user normally interacts with a World-Wide Web viewer. BASAR has access to the logfile of user actions, to the usage profile, and to the distributed Web space.

### EVALUATING BASAR

We conducted a set of empirical tests to provide us with a deeper understanding of how users deal with the concept of assistants (agents) as implemented in BASAR. In the following, we discuss observations and results along the two main characteristics of BASAR, agents and active views.

## Web Assistants

In [2] we introduce three design rationales for assisting agents: shared context, user's attention, and control of agents. The „agents part“ of the evaluation has been centered around two of them – user's attention and control of agents – complemented with the issue of task delegation. Shared context is discussed with active views, see below.



Figure 6. Control of Web assistants

This extract from a Web page (Fig. 2) presents functionality to control two existing assistants: The „Updater“ assistant - currently running - can be terminated by the user, and the „Clean-Up“ assistant (currently sleeping) can be restarted, edited, or removed.

### *Delegating Tasks to Agents*

BASAR supports the delegation model, i.e., a user delegates typical WWW tasks to agents. With the tests we asked for the good and bad of that principle. As BASAR 's aim is to support typical WWW users with no interest or ability in programming agents we have chosen a so-called „task naming“ approach for editing agents (instead of „programming“ the agent with IF and THEN part the task description can simply be selected from a list of tasks) which revealed to be appropriate. If not necessary, users should not be stressed with a complicated agent editor. However, this may look different if more complex tasks have to be performed by agents.

The test users liked the easy way to specify an agent and to select one of the task description: „*Oh yeah, pretty easy! I expected a somehow more complicated interface.*“ The trigger mechanism was well understood, also the options for type and time of notification.

Speaking names for the agents were chosen without problems, e.g., „Updater“. But also a name like „myAssistant“ were given to an agent – a nice name but it does not tell much about the task to perform.

### *Intrusiveness: Catching User's Attention*

In BASAR, we introduced different ways to catch user's attention when agent's task has been completed. These are „sending an email report“, „popping up WWW viewer“, and „blinking an icon“. It is up to the user to select one of these notification types either as a default for all agents or specific for each agent to be created.

During the tests, all users selected „popping up WWW viewer“ as the type of notification, meaning that BASAR pops up a viewer when results have been generated. Users were not very surprised when the result presentation of an agent was suddenly popped up although most of the users had continued their work after the agent was created. This differs extremely to the experiences we had with the

presentation of suggestions in the first version of our adaptive user interface FLEXCEL [12]. It seems to make a big difference whether the system or the user originally initiates an active behavior: if the system is the initiator (as done in FLEXCEL) then the decision when and how to present something to the user must be made very careful. On the other hand, if the user him-/herself initiates and determines how the system should act, e.g., by *consciously* choosing „immediately“ as the time of notification, then he/she can deal much better with such kind of active system behavior.

### *Control of Agents*

In BASAR, agents created by the user are monitored and controlled within a Web page (see Fig. 6). The question was whether the tools available in BASAR for controlling agents are sufficient and usable. In summary, this page has been revealed as too static to provide ongoing and appropriate information about the status of agents.

„*Why doesn't he say that he is working?*“ This is a typical remark given by a test person after creating an agent with the agent editor. Although most of the people selected „immediately“ as trigger for the agent telling that the agent should start working immediately after its definition, it was not always clear that it really did it without giving any further feedback, for example, as one subject suggested, by a window that says: „I started working.“

Some people asked for a better feedback from an agent that is working, for example, by displaying a running clock icon as it is used, for example, in MacOS when the system is busy. It was not realized that, once an agent has been started, it performs its task as a background activity. And for background activities, there is no reason to use a „busy“ sign.

However, as BASAR is a prototype developed in the VisualWorks environment, there was one possibility to keep track of an agent's activities once it has been started: there was always a small window visible (partly hidden by the viewer) that listed all SMALLTALK method calls during run-time. It was not intended as a feedback for users but it was used as such a clue: „*I realized that there was something happening in the background. Therefore I was sure that the agent is working.*“

We conclude that the design of user's control and agent's feedback has to go through some more user design evaluation redesign cycles. This will be part of the future work.

### **Active Views**

If we talk about shared context in BASAR then an active view is one of its representatives: users use active views to manage their personal information space and agents work on these views to support the managing process. Beside the active view, the usage profile and the logfile (as logical part of it) are the other parts of the shared context.

### *Views supporting Personal Information Space.*

With BASAR, we claim that the concept of active views supports (a) users in managing their private information space, and (b) helps users in dealing with information overload. Both issues were considered during evaluation. With our bookmark questionnaire we learned that there are five dimensions that are relevant for the managing of personal information spaces: usage, increase, structure, validity, and sharing. We discuss the active view concept along these dimensions, i.e., we asked which of these dimensions are supported by active views (in combination with agents).

*Usage.* The usage of user's personal information space is kept as the logfile of all user actions. A simple but useful support to get informed about the status of usage comes from the „Clean-Up“ assistant that lists all invalid, all never or all seldom used links (see Fig. 4).

*Increase.* As we know from the bookmarks questionnaire and our own experiences, user's personal information space is always increasing and tools that help dealing with that effect are missing. With the increasing number of headers and the increasing total number of links, the time to spend for managing them will also increase. Such a managing process always asks for the suitability of a computational assistance. The time to spend for managing user's personal information space can be reduced if typical and periodically appearing managing functions can be delegated to agents. Assisting agents in BASAR provide this functionality, e.g., by being able to look at user's history of actions or by checking the validity of the space. With this assisting agent concept - complemented with the search agent features - we seem to be on the right track for providing computational assistance, following the premise not to hinder the increasing of the space but to support the managing of it (and for an agent it does not matter whether it has to check 10 or 100 links but for the user this is quite a difference).

*Structure.* Active views support the structuring of user's personal information space basically with the same mechanism as bookmarks do: a list of headers and each header with a list of links. But additionally, active views provide some extensions for people that have to deal with a great amount of links. First, they keep track of all links that belong to more than one view making the removing of these link easier, for example. And second, BASAR agents can perform their actions on more than one view making the update or the clean-up process, for example, easier to handle.

*Validity.* The test persons validated what we found out with the bookmarks questionnaire. They knew about the number of bookmarks and the number of headers they collected in their private list but most of them had no idea about the validity of the links. Even more, they had no idea about the usage (e.g., never or very seldom) of their bookmarks. In consequence, tools are needed that provide the user with feedback about the relevance of the

bookmark links according to their use and their validity. Active views are able to provide that support, giving the user feedback about the state of his/her space, but let him/her make the final decision about adding or deleting so that user's burden to clean up the space decreases.

*Sharing.* If people share (parts of) their personal information space with others, e.g., by creating a private homepage „my favorite places to surf“ and storing it at a Web server site, then they have to realize that this homepage has become a part of the World-Wide Web itself, accessible for people from all over the world. Sharing an information space with others is a kind of „social act“. Other users may like the collection of links, may become angry about too many invalid links, or, even worse, may become offended due to some links to Web sites that reflect user's political interests, for example. Until now, active views not really support that dimension. However, as future work, this will be discussed when active group views come in.

### **Discussing Agents**

We see two major issues how to deal with the information overload and user's limited ability to absorb information, as there were:

- decreasing the cognitive load on the user, and
- increasing the relevance of what is delivered to the user.

Do the concepts of agents and active views - as they have been implemented in BASAR - support these two strategies - and by that provide user-specific support to personalize the WWW? Lets summarize some of the support users get from BASAR:

- Users get active support from the system; for example, web assistants notify the user about old, new or updated information relevant to an active view. Without BASAR, bookmarks are passive repositories of information links.
- Users can delegate tasks; for example, users can ask web assistants to look for new links in the result list of a search process that are not stored in the user's personal information space. Without BASAR, users must do that by hand.
- Users can ask web assistants to perform periodic tasks; for example, to look for new information on a specific server every week. Without BASAR, users have to do that themselves.
- Users get uniform access to search engines with a knowledge base in the background that describes how the different search engines work, how to call them, the best time to call them, and how relevant their results were in the past. Without BASAR, users have

to remember the necessary information every time they call a search engine.

- Users have access to the history of all actions. For example, BASAR supports the relocating of sites visited in the past whose links have not been stored as bookmarks, for example. Up to now, it may be a kind of frustrating job for users to find something they liked only by surfing and remembering.

#### *Decrease Cognitive Load by Delegating Tasks*

The purpose of BASAR agents is to provide mechanisms that allow users to delegate tasks to agents that normally have to be done by the user in the World-Wide Web. Delegating a task (to an agent) is an additional effort for the user. But if users are able to delegate, for example, search and filter tasks to agents that may work during night when the user is absent, this additional effort is worth to do.

According to the delegation model, the user can decide if and when he/she will get computational support by agents. Once such a task has been delegated, agents work (semi-) autonomously until task completion. Overall, these characteristics of BASAR reduce the cognitive load on users because agents perform annoying and repetitive tasks for the user giving the user more efficient use of time or let him/her perform the same task in a shorter period of time.

#### *Increase Relevance of Link*

Agents in BASAR increase the relevance of information delivered to the user because they can check (even permanently) the validity of links and perform sort and filter operations thereby reducing the set of information delivered to the user according to user-specific filter and sort mechanisms.

#### *Are agents harmful?*

There is some discussion going on whether the concept of agents is harmful, e.g. [4]. According to [13] this may be a problem, especially for agents „that attempt to converse with the users“. So far, agents in BASAR have been implemented as not being able to converse with the user; they only interact with the user along interface concepts that do not provide real conversation mechanisms.

Maes [5] raised the question of trust and competence of agents. She defines it as one of the most important questions concerning agent technology to be answered for each implemented agent based system. How do BASAR agents deal with these concerns? During our tests we observed that the users had no doubts whether the agents would fulfill their tasks or not. The specification of an agent was so concrete that there was no reason to be doubtful. If assistant behavior covers functionality that is normally done by the user then it strongly depends on the complexity of the task to perform. In BASAR, these tasks were clear, meaning that user's need not built a great

confidence to the agents and had no exaggerated expectations [8] what agents could do and what not.

Also, in [5, p. 40] the question of how the user interface of agents should look like is discussed. „Should there be one or many agents? Should agents use facial expressions? What is the best metaphor for interface agents?“ For BASAR, we argue along the following line.

- First, we think that the user him/herself should make the decision whether to deal with one agent that is created once and that can be edited and changed whenever the user likes. Or, whether the user wants to create for each type of task one specific agent. But user's freedom to decide comes not for free. There is the potential danger that too many agents have been created over time, arising an additional managing problem for the user which is definitely not user's intention. For a system like BASAR this means that we have to think about mechanisms or tools that support the minimizing of the number of agents the user has to deal with. One solution is to extend the current version of BASAR with a task composer for agents, for example. Another solution is to directly offer only a few numbers of predefined and complex agents which then should be able to provide explanations what they are able to do and what not.
- Second, as most users are generally not interested in agents *per se*, but in agents abilities to provide assistance, there is no need to „display“ an interface agent with a facial expression. Therefore, in BASAR, agents do not present themselves as „anthropomorphic“ persons to the user, instead they are simply concentrated on the tasks they are designed for: to perform assisting functions for the user to make him/her free for doing other things. Hence, the best metaphor for agents at the user interface will strongly depend on the purpose the agents are used for, and, moreover, from the domain they are supposed to work in. So, in our opinion, there is no general answer to the question for the „best metaphor“, there will be a set of useful and also different answers based on intensive assessments and empirical studies in the domain of interest.

#### *Summary*

Delegating tasks and supporting the browsing and searching processes enable a user (a) to use less time to search the same information space, and (b) to search a larger space in the same time. These two strategies reduce the information overload problem and lead to a new task distribution between users and computers, allowing users to focus on the more interesting aspects of the task.

#### **FUTURE DIRECTIONS**

So far, BASAR supports individual users. To make it suited for a group of users, we started to extend active views to *active group views*. For example, a view on „lifelong learning“ could be shared by the members of our

research groups at CU and at GMD as a joint information repository. Active group views will support collaborative work practices [7], implying that not all users necessarily have to create their own active views, but power users and local developers will create views and the corresponding web assistants.

Moreover, the future work with BASAR will extend the concepts of active views and web assistants to support *information brokering*. This will be tested in a new project called ELFI (Electronic Funding Information) starting this summer in which we intend to use an extension of BASAR to support the funding consultants at the German universities. A typical task for a funding consultant is to periodically search and browse through more than 50 European sites which provide „grant-relevant“ information. The consultants have to check these sites for their relevance according to their university's research directions in order to apply for new grants. The goal of this project is to provide sophisticated agent support for the searching for, the filtering of, and the dealing with that kind of domain-specific information spaces.

#### ACKNOWLEDGMENTS

The authors would like to thank the members of the Center for LifeLong Learning and Design at the University of Colorado. The research was supported in part by NSF Grant IRI-9311839, "Enhancing Indirect, Long-Term Collaboration with Intelligent Agents". We would also like to thank Achim Nick, GMD, who did most of the implementation of BASAR.

#### REFERENCES

1. Fischer, G., and Nieper, H. Personalized Intelligent Information Systems, Workshop Report, Breckenridge, CO, Institute of Cognitive Science, University of Colorado, Boulder, CO, Technical Report, No. 87-9, May 1987.
2. Fischer, G., and Redmiles, D. Enhancing Indirect, Long-Term Collaboration with Intelligent Agents. NSF-Grant IRI-9311839, Boulder, CO, 1993.
3. Kay, A. User Interface: A Personal View. In Brenda Laurel (Ed.) *The Art of Human-Computer Interface Design*, Addison Wesley Publishing Company, Reading, MA, 1990, 191-207.
4. Lanier, J. Agents of Alienation. *Interactions*, July 1995, 66-72.
5. Maes, P. Agents that Reduce Work and Information Overload. *CACM*, July 1994, Vol. 37, No. 7, 31-40.
6. Maltz, D., and Ehrlich, K. Pointing The Way: Active Collaborative Filtering. *CHI-95 Conference Proceedings*, ACM, New York, 1995, 202-209.
7. Nardi, B.A. *A Small Matter of Programming*. Perspectives on End User Computing. The MIT Press, Cambridge, Massachusetts, London, England, 1993.
8. Norman, D.A. How Might People Interact with Agents. *CACM*, July 1994, Vol. 37, No. 7, 68-71.
9. Schick, A.G. Gordon, L.A., and Haka, S. Information Overload: A Temporal Approach. *Accounting Organizations and Society*, Vol. 15, No. 3, 1990, 199-220.
10. Shardanand, U., and Maes, P. Social Information Filtering: Algorithms for Automating "Word of Mouth." *CHI-95 Conference Proceedings*, ACM, New York, 1995, 210-217.
11. Thomas, C.G., and Fischer, G. Using Agents to Improve the Usability and Usefulness of the World-Wide Web. *Proceedings UM-96, Fifth International Conference on User Modeling*, Hawaii, User Modeling, Inc. 1996, 5-12.
12. Thomas, C.G., and Krogsæter, M. An Adaptive Environment for the User Interface of Excel. *Proceedings of the 1993 International Workshop on Intelligent Interfaces*, Orlando, FL, ACM press, 1993, 123-130.
13. Wroblewski, D. McCandless, T., and Hill, W. Agency Considered Harmful. *AAAI Symposium on Knowledge-based Human-Computer Communication*, Position Paper, 1989, unpublished.