

Report T98:04

ISSN: 1100-3154

# ConCall: An information service for researchers based on EdInfo

by

Annika Waern

Mark Tierney

Åsa Rudström

Jarmo Laaksolahti

{annika, mark, asa, jarmo}@sics.se

October 1998

Swedish Institute of Computer Science  
Human Computer Interaction and Language Engineering Laboratory  
Box 1263, SE-164 29 Kista, SWEDEN

---

## Abstract

In this paper, we present new types of web information services, where users and information brokers collaborate in creating a *user-adaptive* information service. Such services impose a novel task on information brokers: they become responsible for maintaining the inference strategies used in user modeling. In return, information brokers obtain more accurate information about user needs, since the adaptivity ensures that user profiles are kept up to date and consistent with what users actually prefer, not only what they say that they prefer. We illustrate the approach by an example application, in which conference calls are collected and distributed to interested readers.

## Keywords

Adaptive Information Services, Intelligent Information Filtering, Agents, WWW, Adaptivity, User Modeling, User Profiling.



## INTRODUCTION

The rapid development of information sources such as the World Wide Web has left readers with an acute problem of information overflow. The problem is not simply one of information retrieval and information filtering; the user might also require aid in summarizing the retrieved information and judging the accuracy and quality of it.

There is a clear role for the *information broker*, the human expert that gathers, structures, and evaluates information. Information brokering services exist today on the web, and some of these utilize individual user profiles to tailor the information selected for individual users. The approach can also be made self-adaptive, so that the system can adapt automatically or semi-automatically to user preferences. An example service that does this is the personalized web radio station Silver Island [8].

An imminent problem in the information brokering situation is that both the selection and structuring of the information is done entirely from the editor's perspective. The typical information brokering service utilizes a predefined classification schema for information. Readers can individualize the service by selecting from the predefined categories. This approach has many disadvantages:

- The individual readers must select between classes of information that may be orthogonal to their real interests.
- The individual reader is also forced to use the broker's classification not only for retrieval, but also for structuring the retrieved information.
- It becomes impossible for users to indicate that they are looking for types of information that are not covered by the service. In other words, the feedback that the information broker obtains from reader profiles describes what readers like and dislike, but not what they really want.
- Finally, as the information changes over time, the classification schema may have to change. But this will require that users change their profiles as well.

We have taken a different approach. The basic idea is that there exist several, parallel classification schemas, which may be specific for users, editors or organizations. The schemas need not be stable over time. With this as our basic assumption, we utilize intelligent filtering techniques to allow information brokers and readers to communicate and synchronize their classification schemas.

## EDITOR SUPPORT FOR OPEN ADAPTIVE INFORMATION SERVICES

Any system that adapts automatically or semi-automatically to the individual user must make its inferences based on some type of stereotypic knowledge about the user [3]. There are two possible sources for such knowledge [7]: either, a human expert uses his or her knowledge about the users to equip the system with pre-coded rules of inference, or else, the rules can be learnt by the system from the behavior of groups of users, or from the individual user's repetitive behavior.

Both sources of information have their weaknesses. The disadvantage with pre-coded knowledge is that it relies on an advance analysis of the user behavior, but once the system is running, users may very well change their behavior. This problem is most apparent in domains where the information is rapidly changing or highly unstructured. How could we, for example, analyze and represent the widespread needs of users of the web in such a way that it would be possible to filter information or adapt navigation to an individual user?

Automatic means of inference can potentially deal with these problems. Most approaches to inferring user preferences are based on the actions of a single user. A problem with such approaches is that it will be hard, if not impossible, for the system to deal with new information. This problem has been addressed in systems relying on group user modeling and user recommendations, so-called recommender systems ([2], [8]). One well-established example of such a service is Firefly [5]. The preferences of an individual user are compared to those of the full user society, so that the system is able to suggest new information based on the fact that other users with a similar preference pattern have liked this information.

However, it may not always be the case that the preferences of a whole group of people will be sufficient to satisfy a particular user's needs. In fact, that user might be much more interested in what a single expert would regard as important information, rather than in the recommendation of a large group of peers. In the general case, users may want to judge the relevance of a piece of information

based both on quality (the expertise of the recommenders) and on quantity (the number of people recommending it).

### **The Information Broker Role in Collaborative Filtering**

The main goal of the EdInfo project [6] is to utilize human *information brokers*, or editors, as a resource in adaptive information systems. An information broker can be any of the following:

- The dedicated expert that collects and potentially reviews literature within a restricted area of interest;
- The journalist that produces articles with specific reader groups in mind;
- The librarian that organizes incoming information and directs readers to various sources;
- The professional information broker, that processes specific information requests, seeks for appropriate information sources, and produces summaries of the obtained information.

The common characteristic of these roles is that the information broker has some kind of understanding of what his or her customers want, and is willing to adapt to these needs. Information brokers collect information from various sources, evaluate its relative importance and then choose whether to include the information as it is, disregard it, summarize it, or perhaps rewrite or illustrate it differently than in the original source.

Most scenarios for information brokering can benefit from introducing group user modeling and individual user adaptation. Many existing information services build upon user profiles, for example news services such as CNN Custom News [1] and the Swedish 25timmar [11]. Users are allowed to explicitly set up their profiles by selecting a set of categories and subcategories that fit their interests. This approach gives rise to a number of problems, however. Firstly, the available categories might not fit the users' real interests and preferences. Secondly, the categorization may have to change if a new need occurs, or a new type of information is added, but then all users must change their profiles to adhere to the new categorization. Finally, it is likely that users seldom change their profiles once they are set up, so it is not certain that the profile really reflects the user's true interests.

Individual user adaptation provides a way to deal with these problems, since it can provide user-defined categorizations that are automatically or semi-automatically maintained by learning from the user's actions with the system. But in order to introduce individual user adaptation, we must impose at least two additional tasks on information brokers:

1. to maintain the rules for stereotypic adaptation, used in user profiles;
2. to structure the information in a way that allows for these rules to be applicable.

The essential source of information necessary for these tasks is *feedback* from users, both in terms of which profiles they set up, and how they use the information they obtain. Since this information in itself is of imminent value for information brokers, we believe that they will accept the addition of these tasks. Nevertheless, a definite requirement is that the maintenance and information structuring tasks are made as simple as possible; we cannot assume that information brokers have any particular interest in the details of the algorithms for user modeling.

### **CONCALL: FUNCTIONALITY, ARCHITECTURE AND IMPLEMENTATION**

ConCall is an agent-based system that implements the EdInfo ideas. The system supports the collection, filtering and browsing of conference and workshop calls, but could just as well be used for calls for participation in seminars, courses, etc. Using ConCall, the user (an individual researcher) can review calls and set up reminders for deadlines. To avoid uninteresting calls, the user sets up a filter to retrieve a personal selection of calls and organize them in a personal manner. This filter is maintained by semi-automatic means.

The service is accessed over the web. Reminders are received by email or SMS<sup>1</sup>. The first version of the ConCall service has been implemented and is currently under experimental evaluation.

---

<sup>1</sup> Short Message Service – a messaging service provided in European GSM mobile phone networks.

## Technical Details

The ConCall service is built on an agent architecture that consists of a number of specialized agents. These are:

- The Personal Service Assistant (PSA). This agent handles the interaction with the user and provides a central point of interaction between the user and the agents in the architecture.
- The user profile agent. This agent stores the preferences of the user. The user profile is based on information about the user's actions received from the ConCall agent. The user can inspect and change the profile.
- The ConCall agent itself. This agent does the filtering of calls for each user.
- The reminder agent. This agent provides a reminder service to the user and to other agents. The user may be interested in getting notification prior to a deadline for submission of papers to a conference, and the reminder agent will handle this by sending an email or SMS to the user at the appropriate time.
- The database agent. This agent handles transactions with a database that stores the conference calls. Calls are entered into the database by the editor. (Note that this architecture is a prototype version where the editor is not explicitly represented in the architecture. Calls are collected, classified and entered into the database using tools available to the editor that are not part of the user's collection of agents. In a forthcoming implementation, the two will be integrated).
- The logging agent. This agent is accessible from all other agents and enables agents to keep a record of events.

Agents communicate with each other by KQML messages [4], where the content is represented as ground Prolog terms<sup>2</sup>. The ontologies used are application-specific. Users communicate with the agents by special-purpose user interfaces. The PSA and the reminder agent have their own user interfaces. The user communicates with the user profile agent, the ConCall agent and the database agent through one shared applet (see figure 1).

The agent architecture used in ConCall is designed to be open and extendable to other kinds of information services. Waern further discusses this aspect of ConCall in [10]. Agents in ConCall have at least two interfaces: one towards users, and one towards other agents. In figure 1, this is denoted by each agent consisting of two parts, one half devoted to user interaction, and the other to interaction with other software agents.

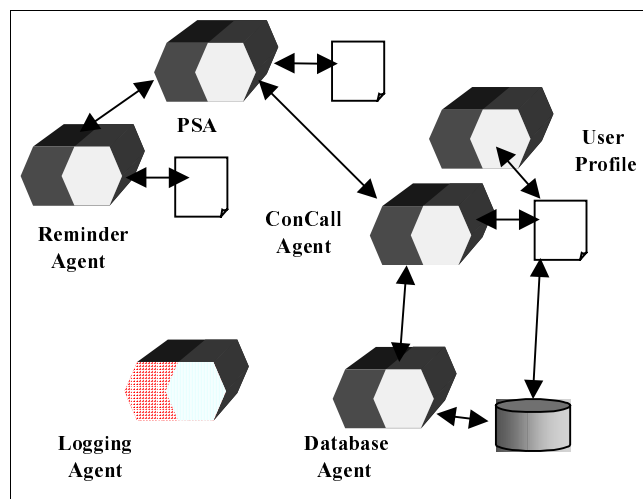
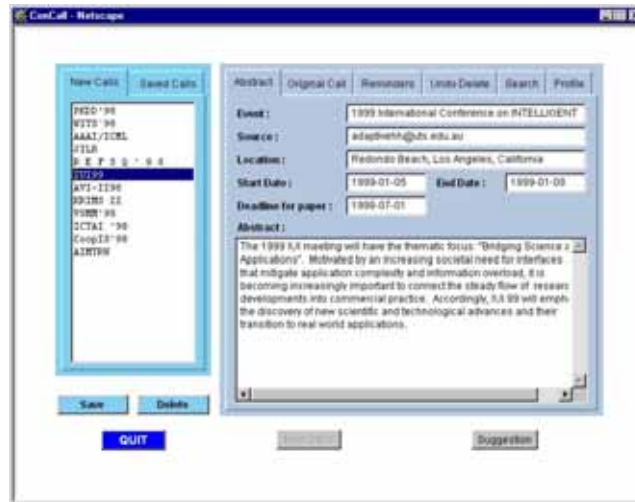


Figure 1: ConCall architecture

Each agent has a span of activities that it can perform, although some of them require interaction in a particular format that may only be accessible to one type of agent (e.g. a user). The reminder agent has complete overlap in services: both agents and users can set up reminders. The ConCall agent uses this

<sup>2</sup> Only the “ask-one”, “ask-all” and “tell” performatives are used.

functionality to set up reminders for conferences when the user asks for it. For most other services, the set-up and control role is reserved for the user. For example, the user profile agent maintains the user profile based on information about the user's actions that it receives from the ConCall agent. The user is given a means to set-up, inspect and control this profile; in particular, the user decides when to activate or deactivate the inferred profile.



**Figure 2:** Screen shot of the presentation of a call

In addition to user profile management, ConCall provides functionality for browsing calls or parts of calls, searching the database, and for saving, deleting and putting reminders on calls. Figure 2 shows a screen shot from the ConCall system, in which a summary of a particular call is presented.

### "BUZZWORD" PROFILING

An immediate concern for the design of the ConCall user profiling was that it should be open-ended and adaptive, but yet be as simple as possible to maintain. To achieve an open-ended design, we required that neither users nor editors should be bound to any predefined ontology in the formulation of filters and the structuring of calls. To achieve simplicity, profiles and annotations should be as simple as possible to still work as a filter mechanism. Users and information brokers are essentially free to define their own classification schemas, but to enable information brokering, these schemas must be at least partially synchronized.

The filtering mechanism for the first version of ConCall relies on a particular type of meta-data annotation that we have chosen to name *buzzwords*. The buzzwords for a conference call are simply a set of terms that have been chosen by the editor as a useful characterization of the conference or the call. Users set up their profiles in terms of which buzzwords interest them. Note that we do not intend buzzwords to mean the same as keywords. Keywords typically aim to describe the topic or style of a conference according to some pre-defined categorization schema. The buzzwords, on the contrary, aim to reflect the subjective interests of the editor and the users. They are not picked from any stable categorization (and need not be predefined at all), they are expected to have a "fad" quality and often reflect trends in a research community. Furthermore, they need not be restricted to describing the topics of the conference, but may include names of people in the program committee, the place of the conference, as well as meta-information like "final call" or "extended deadline", etc. The buzzwords are rather to be seen as an open-ended communication channel between users and information brokers.

### User Profile Handling

Figure 3 shows a screen shot of the user profile maintenance screen. In the left part of the right-side frame, the active profile is shown as a list of positive and negative buzzwords. The user can delete buzzwords or add buzzwords to the profile. The user is given several tools that aid him or her in adding buzzwords. He or she can select a buzzword from a menu of buzzwords in use in the service (the menu right below the active keywords), select a buzzword from the candidate profile (shown to the right of the active profile), or type in an arbitrary buzzword (field below the candidate profile). Finally, the user

can accept the system's suggestion for a profile as it is, replacing the old profile with the system's suggestion.

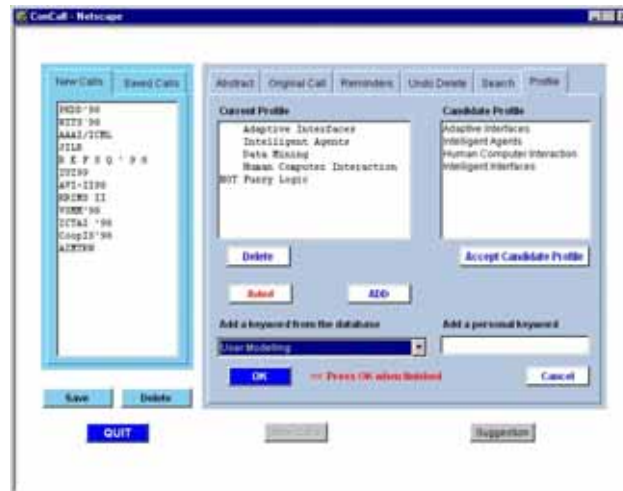


Figure 3: Screen shot of the user profile management

### Broker Profile Handling

The information broker maintains the list of buzzwords in use in the service. The list is automatically updated when new buzzwords are introduced into the system.

In the current version, only buzzwords introduced by the editor are added to the list. The next version will also include buzzwords that users have added to their profiles.

The broker is also free to inspect and modify this list by hand: adding buzzwords that may show useful, modifying the exact formulation of buzzwords, and deleting buzzwords. The task of maintaining the list of buzzwords in use is a simple variant of the editor task of maintaining the rules for stereotypic adaptation, since the buzzwords determine what the system may learn about user preferences.

The buzzword list is used when annotating incoming calls. This is done in two steps:

1. The broker uses information retrieval tools (initially simple free-text search) to extract the subset of buzzwords that are found in a call text.
2. The broker inspects the resulting annotations, and may add or delete buzzwords.

Usually, this is only done for new calls that the broker enters into the database. As the list of buzzwords in use changes, it might also be necessary to maintain the annotations of old calls. We expect that this will happen a lot during a start-up phase, but since conference calls have a limited lifetime, it will probably become less frequent once the set of buzzwords in use reaches a certain stability.

### Local and External Usage of Profiles

The user profile is used in two ways in the system. Firstly, it is used as an external filter, determining which calls to retrieve from the database. The ConCall agent asks the database agent to retrieve all calls that have at least one matching (positive) buzzword. Secondly, once the calls have been retrieved, they are ordered locally by the ConCall agent according to how well they fit the user's interest. Calls that match several of the keywords are listed above those with lower number of matching buzzwords. As shown in figure 3, the user profile also contains disinterest buzzwords. These are seen as exceptions, i.e. terms that make a user less interested in a call where many of the positive keywords fit. They are only used in the local sorting.

This double usage of the user profile is implemented as a first step towards splitting the user profile into a private, local part and a public, external part [3]. Such a split may be important both for privacy and performance reasons.

## Local User Modeling

Internally, the user model contains three different profiles: the candidate buzzwords, the candidate profile, and the actual profile. The candidate profile and the actual profile are shown in figure 3. The set of candidate buzzwords is internal to the system and not shown to the user. The set of candidate buzzwords and the candidate profile are maintained locally by the profile agent, by inferring from implicit information about user preferences indicated by user actions. Some of the actions that users make on calls are seen as indications of interest or disinterest - the system currently makes use of when the users set reminders, save, or delete, calls.

The set of candidate buzzwords and the candidate profile are maintained by a simple learning algorithm. Whenever the user performs an action that the system uses as a key to user interests, all buzzwords of the call that the user acted on are added to the set of "candidate buzzwords". If the action was a reminder or a save, each buzzword is given a slightly positive value, if it was a delete, it is given a small negative value. If the same keyword occurs in another call that the user acts upon, the value will be increased or decreased depending on the type of action. This way, the value of each buzzword will reflect how consistently the user likes or dislikes calls annotated with the buzzword: if the buzzword occurs both in saved and deleted calls, its value will approach zero. The algorithm also ensures that if certain buzzwords become more and more significant, buzzwords that are rarely used at all will also be given a value that approaches zero.

If the value increases or decreases above or below a certain threshold, the keyword is added to the candidate profile. If the value instead approaches zero, the keyword is eventually deleted from the set of candidate buzzwords.

The candidate profile and actual profiles are given an overall support, which is based on how well it predicts the user's actions. Whenever the user makes one of the significant actions (save, delete or remind), the system maps the buzzwords of the call to see how well it fits the profile. If the call gets a high rating according to the profile, the system predicts that the action would be a save or remind action. If the rating instead is very low, the system predicts a delete. If the user's actual action is the same as the predicted action (the system also allows for a certain overlap in predictions) the user's action is seen as supporting the profile. The support for a profile is generated over time, and is simply calculated as the number of user actions that the profile has predicted since it was last changed. The difference in support between the candidate and actual profiles determine if the system should present an active suggestion to the user to change his or her profile.

Despite the simple structure of the local user model, several things need to be adjusted to make it work in practice. In particular, we expect to need to tune the stability of the profiling. When should a buzzword be considered good enough to be added to the candidate profile, or insignificant enough to be deleted from it? How much better than the actual profile must the candidate profile be to motivate an active suggestion to the user? We hope to be able to tune the stability of the system by comparative testing, using logs from actual user interactions with the system.

## FIRST STUDY

One of the central features of ConCall is that filtering and sorting is done with a very simplistic structure: lists of buzzwords. The simplicity is intentional: we believe that if too much structure is used, the burden on creating and maintaining profiles and annotations becomes too large, both for users and for information brokers. But this forms a major risk for the design solution: users and editors might not be able to communicate sufficiently well using only these lists of buzzwords.

The first study performed with ConCall has been designed with the specific aim to find out whether the buzzword structure is sufficient to allow users and information brokers to communicate. The second purpose of the study is to gather information about which, of a number of possible extensions, is the most appropriate to deal with potential problems concerning the communication between editors and users. In addition to these objectives, we will also use the trial to collect log data that will be used to tune the user modeling algorithms. Evaluation of the usability of local user modeling will be performed in a later phase, once the algorithms have been tuned.

To evaluate the usefulness of the buzzword functionality, we will analyze log data from a set of users interacting with the system. We will also allow an expert information broker to inspect the user profiles and utilize this information in annotating calls. Specifically, we seek answers to questions of the type:

- Do readers change their profiles once they are set up?



- Do readers ever type in their own buzzwords, or do they choose from the menu or from suggestions from the user modeling?
- Is it possible to find clusters of user profiles with similar buzzwords?

We will also interview users about their subjective experiences with the system. These interviews will mainly be directed towards finding out the usefulness of possible extensions, such as

- if users need higher expressive power in formulating profiles;
- if users would find it useful to review each other's profiles; and
- if users would make use of a possibility to categorize calls into personal categories, or alternatively, a categorization of calls supplied by the broker.

The study will start in the beginning of August 1998, and the results will be available by the end of the year.

### **THE FUTURE OF CONCALL**

There are many possible extensions of ConCall, and one of the main purposes of the first study is to establish which of them seem most promising. Here, we only list the possibilities we have envisioned so far.

Firstly, we would like to fuse the roles of the information broker and the reader. It is obvious that ConCall would benefit from a scenario where anyone can add conference calls, as well as read them. It is very simple to extend ConCall this way: what is needed is an annotation on calls that tells who added the call, and some access restrictions on who can make modifications to calls. A more interesting extension along these lines is to make ConCall utilize a distributed database approach instead of a traditional client-server architecture. The aim would be to move the service itself onto the web, connecting several ConCall clients with their own databases of calls. In both cases, one difficulty will be to ensure that users do not obtain too many duplicates of calls. This will call for a more extensive usage of text summarization in information retrieval.

A more elaborate extension is to increase the support for collaborative filtering. Readers may benefit from being able to review each other's profiles, or being presented with a typical profile for a group of users. Editors can get recommendations on new buzzwords that are based on what other people use, rather than on the annotations of calls that they actually read. Also, users may want to sign up for calls entered by a particular editor or a particular organization, rather than arbitrary calls that fit their profile.

Finally, the information may need to be better structured. This may concern both the list of retrieved calls and the formulation of profiles. One possibility is to allow users to sort their incoming calls into individual categories. Instead of setting up one single profile, the user is then given the option of setting up a profile for each of the categories, which is used to select which calls should be sorted into this category. Information brokers may also suggest categories that also may have a "buzz" flavor and possibly change over time. It is also possible to increase the expressive power of profiles by adding types to buzzwords, so that the search for the buzzword is limited to a certain type of information ("Hawaii" is a location buzzword, "Rudström" is a program committee buzzword).

### **CONCLUSIONS AND FUTURE WORK**

The ConCall service shows that edited adaptive information services indeed are possible without imposing large and complicated novel tasks on information brokers. Edited adaptation provides added value for the individual user as well as for the information broker, who obtains better means to organize information to suit the needs of the readers.

The buzzword functionality used within ConCall requires further study. It is possible to envision several ways to enhance it, both in terms of how users and editors can get hints on what buzzwords to use, and in terms of the structure of the buzzword annotations. The key issue here is not to find the most all-encompassing solution, but to find a solution that strikes a balance between expressivity and simplicity, allowing editors and users to communicate efficiently.

### **ACKNOWLEDGEMENTS**

The EdInfo project is funded by the Swedish research institute for information technology (SITI AB) and the Swedish board for technical development (NUTEK).

## REFERENCES

1. *CNN custom news*. (1998) Cable News Network, Inc. Available at <http://customnews.cnn.com>
2. *Communications of the ACM*. (1997) Special issue on recommender systems. Vol. 40, No. 3. March 1997.
3. Cook, R. and Kay, J. (1994): The Justified User Model: A Viewable, Explained User Model, Proceedings of *Fourth International Conference on User Modeling*. The Mitre Corp.
4. Finin, T., Labrou, Y., and Mayfield, J. (1995). KQML as an agent communication language. In Bradshaw (Ed.) *Software Agents*, MIT Press, Cambridge
5. *Firefly: Personalize your network*. Available at <http://www.firefly.com> [Accessed June 26 1998]
6. Höök, K, Rudström, Å., and Waern, A. (1997) Edited Adaptive Hypermedia: Combining Human and Machine Intelligence to Achieve Filtered Information. In Milosavljevic, Brusilovsky, Moore, Oberlander and Stock (Eds.), proceedings of the *Flexible Hypertext Workshop*. Macquarie Computing Report No. C/TR97-06, Macquarie University, Australia. Available at <http://www.sics.se/~kia/papers/edinfo.html>
7. Maes, P. (1994) Agents that Reduce Work and Information Overload. *Communications of the ACM*, Vol. 37, No.7, pp. 31-40, 146, ACM Press.
8. Shardanand, U. and Maes, P. (1995) Social information filtering: Algorithms for automating "word of mouth". In proceedings of the *Conference on Human Factors in Computing Systems - CHI'95*. ACM press, 1995.
9. *Silver Island*. Available at <http://www.silverisland.com> [Accessed June 26 1998]
10. Waern, A. (1998) Service Contract Negotiation - Agent-Based Support for Open Service Environments. To appear at the *4th Australian workshop on Distributed Artificial Intelligence at AI'98*, Brisbane, Australia.
11. *25timmar* (in Swedish). Marieberg Interactive AB. Available at <http://www.25timmar.se/>