Architecture for Agent-Mediated Personalised News Services

Marko Turpeinen, Janne Saarela, Mari Korkea-aho, Tuomas Puskala, Reijo Sulonen

Department of Computer Science
Helsinki University of Technology

Corresponding author:

Marko Turpeinen
Helsinki University of Technology/ Room Y309b
Otakaari 1
02150 Espoo
Finland

corresponding email: mtu@cs.hut.fi
phone: +358-50-64084
fax: +358-0-4513293
Abstract

Emerging agent technology can be used to implement a personalised news service. The consumer agent has a model of user's preferences and is able to request information that fulfills these needs. The producer agent has a logical model of the contents of the multimedia objects that are available for service. On the basis of this model, the producer agent can advertise its services in the network to potential consumer agents. The selected multimedia items are offered to the consumer as a personalised newspaper.

This architecture is used in practice in building a personalised newspaper for the World Wide Web, which selects and presents the user with articles that match user's preferences. The article selection process is a combination of keyword searches, semantic matching and social filtering. In our agent-based service, the newspaper editors have an important role in defining semantic categories for the contents of the service, and in providing necessary metadata about the news articles. The implementation is based on object-oriented databases, KQML and Java.
Architecture for Agent-Mediated Personalised News Services

Marko Turpeinen, Janne Saarela, Mari Korkea-aho, Tuomas Puskala, Reijo Sulonen

Department of Computer Science
Helsinki University of Technology

Abstract

Emerging agent technology can be used to implement a personalised news service. The consumer agent has a model of user's preferences and is able to request information that fulfills these needs. The producer agent has a logical model of the contents of the multimedia objects that are available for service. On the basis of this model, the producer agent can advertise its services in the network to potential consumer agents. The selected multimedia items are offered to the consumer as a personalised newspaper.

This architecture is used in practise in building a personalised newspaper for the World Wide Web, which selects and presents the user with articles that match user's preferences. The article selection process is a combination of keyword searches, semantic matching and social filtering. In our agent-based service, the newspaper editors have an important role in defining semantic categories for the contents of the service, and in providing necessary metadata about the news articles. The implementation is based on object-oriented databases, KQML and Java.

1 Introduction

Personalised news service is an ideal and challenging task for autonomous agents. Users are looking for personal assistants that are constantly monitoring the network for interesting information and services. On the other hand, the service providers try to differentiate among thousands of other information production sites (figure 1). How can the producers target their message to the right consumers?

Agents can be considered as mediators [Wiederhold92] that refine and forward information from heterogeneous data sources to the users. Multi-agent intercommunication methods enable message passing between agents in a network environment. The consumer agent transmits user requests for potential producer agents and filters messages according to user preferences. The producer agent acts as an information broker that has a domain model of its own expertise [Fikes95]. The producer agent can advertise the services to the consumer agents in the network. Agents negotiate how, when, and which information items should be transmitted. Agents are also able to consult other agents for suggestions and further information. Finally the agents assist in completing necessary data transfer tasks and financial transactions.

The producer agent presents some descriptors of the multimedia information items to the consumer agent, that acts as an information filter. The consumer agent forwards to the user a subset of items selected based on user profile. Users have the option of providing the consumer agent with feedback either explicitly (i.e. rating mechanism in the user interface) or implicitly (i.e. skipping over the uninteresting items).

1 Email: {mtu, jsaarela, mka, tpu, shs}@cs.hut.fi
A consumer agent can also act as a source of information for another consumer agent. The agents use Agent Communication Language (ACL) to communicate with each other. If the agents move from one host to another they meet at a specified Agent Meeting Point (AMP) [Chess95][White96].

![Figure 1. Agent-mediated information services](image)

### 2 Service architecture

System architecture for personalised multimedia services is presented in figure 2. This architectural presentation is simplified by considering producers of services, brokering services, and other agents that recommend services as single server entity that can answer customer queries. The material can also be distributed to multiple servers and the system performance can be optimised to better balance the load and networking bottlenecks [Korkea-aho95].

Agents may need to access only descriptors of the multimedia items, not their actual contents. Information source, producer or other agent, provides logical addresses for the selected items, and the agent presents these to the user. The selected items will then be fetched from the information producers and presented by the user's multimedia presentation control module. The cost of information to be delivered can also be presented to the user, before the actual delivery is done.

Multimedia products are stored as objects in a database. The database includes also a logical description of the contents of the service (i.e. metadata), as well as the actual multimedia objects to be served. These two functions are provided by two different servers, because the producer agent needs to access only the metadata server to keep track of available items. The metadata database is very small in comparison with the actual multimedia object repository available at the storage server. The contents of the metadata database can be easily cached and replicated between server sites. Storage services can also be provided by specialised companies (for example network operators) who have the needed storage and bandwidth capacity.
Metadata includes information about the location and characteristics of the data to be retrieved. Metadata must also be available about the logical structure of the data objects and relationships between them. This information is used to create dynamic hyperlinks between the data objects. For example, a newspaper article may be accompanied with a video presentation about the news topic.

The server includes the security and session control layers. User authentication and authorisation is carried out before the consumer agent is allowed access to database contents. In some services, the metadata server can be accessible to everybody, but the object repository is accessible only to authorised users.

The session control layer is responsible of recording all the transactions for usage analysis and billing. It can also help in intelligent resource allocation (user distribution, caching, bandwidth, server load, etc.) to enhance the scalability of the system.

The consumer and producer agents can be implemented as agencies, where there are multiple filtering agents, each with their specific task. Here we have treated them as single agents. The agents are mobile, so the consumer agent can traverse to the host where the producer agent resides, and vice versa. This is not necessary in all inter-agent communications. The agents can also move from one producer host to another.

Figure 2. An architecture for personalised multimedia news services
To choose personal multimedia presentations, the consumer agent monitors and analyses continuously consumer's preferences and updates the user model. The user can always examine and change the contents of the user model. The user can also teach the consumer agent by giving explicit feedback.

An example of data flow in this system architecture is presented by numbered arrows in figure 2:

1. **User modeling.** Consumer's preferences are maintained in a user model. The maintenance can be done explicitly by the user or automatically by a learning mechanism in the consumer agent.

2. **Content queries and promotion.** Consumer agent sends a query to the producer agent to receive items that match the user interests. Also parts of the consumer's user model can be sent to be used in social information filtering performed by the content producer. Producer advertises its services to consumer agents.

3. **Dynamic presentation.** The matching objects are ordered from the object repository. A schedule of retrieval times for multimedia presentation is created dynamically at the server. This schedule will be used by the client to request individual objects. This has an effect on the caching decisions at the server. Cache usage can be optimised, when the presentation times are known in advance [Ramanathan94].

4. **Delivery.** The user receives multimedia data objects from the storage server according to the scheduled presentation. The user can cancel the delivery, and has also the possibility to re-schedule the presentation.

5. **User feedback.** The user gives feedback to the consumer agent by rating the delivered presentations. The agent can also detect implicit feedback.

### 2.1 Consumer agent

The high-level goals of the consumer agent in personalised news services are

- to minimise the amount of undesirable information;
- to model the user's requirements;
- to present the information to the user in a convenient form;
- to minimise cost.

A consumer agent consists of several sub-modules presented in figure 3:

**Request Engine**
Request engine formulates queries to information sources based on user's information needs. It interprets the query results when they are available. The request engine can also answer queries made by other similar agents.

**Message Handler**
Any messages in agent intercommunication are handled by the Message Handler. It composes messages to be sent, and decomposes messages sent by other agents and passes the information to other modules.

**Learner**
Learner is responsible for maintaining the user profile. Learner receives all the user feedback. It should be able to modify the user model based on the analysis of the feedback to better match user's interests.
Figure 3. Modular structure of a consumer agent

*User Interface*
There should also be standard features in the user interface, such as mechanism to give explicit feedback and tools to examine and change the contents of the user model.

*Presentation Manager*
The multimedia presentations of the personalised service can be formatted dynamically at the user end by the consumer agent as the information becomes available.

*Dispatcher*
Dispatcher is the core of the agent that delegates the unfinished jobs to other modules.

### 2.2 Producer agent

The goals of the producer agent are

- to be aware of the contents of the object repository;
- to find maximum number of consumer agents for the services;
- to optimise the presented information for available bandwidth and user equipment;
- to co-operate with other server modules in the production of metadata.

A producer agent consists of following sub-modules (figure 4):

*Content Monitor*
Content Monitor has access to the logical model of the service and it keeps track of updates in the contents of the object repository. It can also inform other server modules about the
Object Broker
Object Broker accesses the Object Repository, or a separate Metadata Repository, if it exists, for information about the multimedia objects. It allocates the potential customers in the network by sending advertising messages to the "Yellow Pages" servers. It can also contact the consumer agents directly to promote some new service of information.

Message Handler
Any messages in agent intercommunication are handled by the Message Handler. It composes messages to be sent, and decomposes messages sent by other agents and passes the information to other modules.

Scheduler
Scheduler is responsible for synchronisation and orchestration of time-dependent multimedia presentations.

Figure 4. Modular structure of a producer agent

Presentation Generator
The personalised presentations are created dynamically at the server based on the metadata about multimedia objects and the timetable provided by the Scheduler.

Dispatcher
Dispatcher is the core of the agent that delegates the unfinished jobs to other modules.
2.3 User modeling and learning

User modeling and learning components are part of the overall architecture. User modeling process is complicated, since the knowledge about the user can be incomplete, not available, and contradictive. The modeling can be done in a satisfactory way using non-monotonic reasoning techniques. The system must also be able to "unlearn" the knowledge as user's needs change over time. Thus there is a need for a belief revision system. Various learning methods, such as hill climbing, genetic algorithms, or neural networks, can be used in updating the user profile.

The sensitive nature of the user model implies that there should be a security mechanism controlled by the user to allow and disallow access to this information. The contents of the user model should be encrypted. The dissemination of non-sensitive information in the user profile can be implemented by organising the model in three security levels of information:

- **Private.** These information items are accessible to the user only, they can not be given to outside agents. When private information is used in finding services in the network, the identity of the user should not be revealed.
- **Trusted.** The user can specify trusted agents to whom these items can be accessible. Trusted agents can be grouped and the authorisation may done on a group level.
- **Public.** This part of the user model is accessible to anyone who is allowed access to user's environment.

The user model should be stored in a compact and portable format on a PC card or a local hard disk.

3 Personalised news service

The principles outlined in this architecture are tested in practice in the OtaOnline project at the Helsinki University of Technology. We have approximately 300 incoming news articles daily from three major Finnish newspapers. These articles are offered as a World Wide Web -service to the readers in the campus area. Currently the service has about 4000 registered users. Our project team works in close co-operation with the newspapers in developing their online products for the Web.

Agent technology is used to provide each user their own personalised version on-demand of material available in connection with OtaOnline. Each OtaOnline user has a possibility to launch a consumer agent for information filtering. The consumer agent is capable of modeling the user's interests in a user profile and sending information requests to OtaOnline producer agent.

The producer agent processes the requests based on the user profile. It sends a notification to the consumer agent when the personalised pages are ready to be browsed. The consumer agent forwards this information to the user.

The system uses a combination of content-based filtering and social filtering techniques [Malone87, Shardanand95]. The news selection service is based on a user profile that consists of:

- keyword-based query profile on user-specified topics;
- semantical matches based on predefined categories;
- trusted agents that send recommendations to each other.

2 http://otaonline.hut.fi/
The keyword-based selections are defined entirely by the user. These are normally used to cover short-term information needs. Each topic is identified by a topic header, producer agent and a collection of keyword/weight-pairs. The keyword weight is measured as a value in the range between 0 and 1. The weight can be adjusted by the user or by the learning module of the consumer agent. Also exclusive keywords can be entered to discard articles.

The semantical categories are defined by the newspapers and this domain model is used by the producer agent in advertising the service to the consumers. Our approach is a straight-forward two-level hierarchical tree of categories, such as Business/Software, Politics/European Union and Sports/Football. The semantic categories are used in two ways:

- the editors assign values to the articles from producer's point of view;
- the user can set a profile to reflect his/her interests.

<table>
<thead>
<tr>
<th>[User-defined topics]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Producer:</strong> OtaOnline News Agent</td>
</tr>
<tr>
<td><strong>Topic:</strong> Finnish Formula 1 drivers</td>
</tr>
<tr>
<td><strong>Keywords:</strong> Mika+Häkkinen 0.9, Mika+Salo 0.9, Williams 0.5, McLaren 0.7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>[Predefined categories]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Producer:</strong> OtaOnline News Agent</td>
</tr>
<tr>
<td><strong>Category:</strong> Business, 2</td>
</tr>
<tr>
<td><strong>Category:</strong> Business (Computer Industry), 4</td>
</tr>
<tr>
<td><strong>Category:</strong> Sports, 4</td>
</tr>
<tr>
<td><strong>Category:</strong> Sports (Motor Sports), 5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>[Trusted agents]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name:</strong> Janne's News Agent, 5</td>
</tr>
</tbody>
</table>

**Figure 5. An example of a user profile**

The user profile includes values from 0 to 5 for each news category. By default, values for all categories are set to 3. For example, value 4 for sports category makes sure the user gets only articles having representative value greater or equal to 4.

The newspaper editors are responsible for giving the semantical values for each news article. In theory, natural language processing could be used for automatic semantical
analysis of the contents of the article. In our application we expect to see better results when the news articles are categorised by human experts.

The social filtering concept in our application is also fairly simple. User can select trusted agents for recommendations by their symbolic name. A dedicated Agent Name Server (ANS) can be queried for available agents. Any user can give explicit feedback by rating an article. If the user has not asked the consumer agent to keep the ratings private, they will be collected at the producer's site. Other agents will now be able to use these ratings as recommendations for an article. The user will receive an article recommended by a trusted agent, if the user has also shown interest in the semantical categories that the article belongs to.

4 Implementation

An object-oriented database is used to store the logical model of the service and metadata about the available objects. We use Objectivity/DB as our database engine and the tools for the producer agent are coded in C++ and Perl. All the objects have a unique object identifier (OID) in the database.

The consumer agent is implemented in Java\(^3\). The user interface for personalised "front pages" and the tool for user profile management are implemented as Java applets. The user has also the possibility to get the front pages in HTML, with loss of some functionality. When launched, the profile management applet queries the producer agent for available semantical categories and displays them to the user (figure 6). The actual newspaper articles are HTML pages and other standard WWW material stored at the producer's site.

The articles are stored and indexed in a WAIS database, and the producer agent makes the keyword-based queries to the freeWAIS-sf\(^4\) search engine. However, the WAIS engine does not have good support for Finnish grammar, which is important in providing relevant results.

The agent-to-agent communication is based on Knowledge Query and Manipulation Language (KQML)\(^5\), which is a language and protocol for exchanging information and knowledge. KQML has an efficient set of performatives for queries, capability definition and networking. The chosen API for KQML is the Java Agent Template\(^6\), which has been implemented at the Stanford University. It provides also asynchronous exchange of KQML messages via sockets and enables the dynamic exchange of Java classes.

\(^3\) http://java.sun.com/
\(^4\) ftp://ls6-www.informatik.uni-dortmund.de/pub/wais/
\(^5\) http://www.cs.umbc.edu/kqml/
\(^6\) http://cdr.stanford.edu/ABE/JavaAgent.html
5 Related work

There is a wide range of research projects on personalised information delivery and some commercial applications have also started to appear. Fishwrap [Chesnais95] is a personalised news system implemented at the MIT Media Lab. It relies on natural language parsing, full-text search and keyword-based profiles that are updated manually by the readers. Krakatoa Chronicle [Kamba95] uses Java language to build a personalised newspaper for the WWW. It supports precise formatting and multicolumn layout that are typical in printed newspapers, and the layout options are configurable by the users.
WebDOGGIE\textsuperscript{7} is social filtering applications for WWW page recommendation. It employs statistical methods to compare user profiles in filtering the suitable pages for each user. A video selection system has been implemented at Bellcore [Hill95], that takes advantage of the gathered history-of-use information. Our current social filtering approach is based entirely on explicit recommendations by other trusted users.

SIFT\textsuperscript{8} is a keyword-based approach to filter NetNews articles and is based on a customised user profile. CRAYON\textsuperscript{9} and NewsPage\textsuperscript{10} are examples of information brokering services on the WWW that act as filters between the news sources and the readers. CRAYON allows the user to build his own customised front page by selecting from a list of pre-defined items. NewsPage has a multi-level semantic categorisation for the news items, but has no possibility for personalised selection.

\section{6 Discussion and future work}

Our architecture has been designed to ensure the scalability of the overall system. The personalisation of the system requires more CPU time from the servers to process individual user requests. The metadata servers and the producer agents can easily be distributed to multiple sites, so the CPU load can be reduced. The communications between the agents impose extra load on the network, but should be reasonably small when compared to traffic of sending the actual multimedia contents across the network.

The use of a simplified domain model is motivated by the fact that the producer knows what type of information it serves. Also there are different semantic categories for different newspapers (e.g. producers); an evening paper and a business paper do not share the same domain model and do not evaluate their articles in the same manner. However, the concept of a simplified domain model that is imposed by the producer raises numerous questions. What if the consumer wants to organise the information in a different way than the service producer? What happens when the producer restructures the domain model? Are the user profiles capable of adjusting automatically to these changes? OtaOnline is used as a testbed for our approach and we expect to get answers to these questions from real-world experiments with our users.

The issue of user privacy is crucial to a large deployment of personalised services. In our current model the consumer must essentially advertise his/her interests to the producer. We are exploring ways in which the cryptographic protocols could help in making the agent transactions completely anonymous, while still preserving the possibility to charge the user for the service. Also different levels of privacy in the user model will be supported in the future.

Java language has been chosen to ensure portability and easy distribution across the Internet. The current approach does not use mobile agents, such as Telescript\textsuperscript{11}, that could move from one producer site to another and gather the appropriate news items as they proceed. This concept is worth further exploration.

\section{7 Conclusions}

There are many possibilities for creating personalised services in a network environment. The role of the intelligent agents is to act as electronic brokers, and to match the needs of

\textsuperscript{7} http://webhound.www.media.mit.edu/projects/webhound/doc/Webhound.html
\textsuperscript{8} http://sift.stanford.edu/
\textsuperscript{9} http://crayon.net/
\textsuperscript{10} http://www.newspage.com/
\textsuperscript{11} http://www.genmagic.com/
the consumers with the available producers in the information marketplace. An information filtering agent should be able to do sophisticated semantic retrieval of information objects from a multimedia object database. The consumer agent makes a query to one or more producer agents at the servers, which retrieve information and present it to the consumer agent.

Personalisation of the news service is very important as the producers are looking for ways to provide value-added service for their registered users. The emphasis in the OtaOnline project has been in the editorial process of networked multimedia production. In our agent-based implementation the editors will have an important role in providing the necessary metadata for personalisation. Our approach is a practical way to implement a personalised news service for the World Wide Web using multi-agent communications technology.

Acknowledgements

We would like to thank TEKES, Nokia and Aamulehti Group for their financial support for this work.

References


