Agent Technology for Personalized Information Filtering: The PIA-System

Sahin Albayrak, Stefan Wollny, Nicolas Varone, Andreas Lommatzsch, Dragan Milosevic
DAI-Lab of TU of Berlin
Franklinstraße 28/29, 10587 Berlin, Germany
Fon: +49 30 314 21761
{sahin.albayrak, stefan.wollny, nicolas.varone, andreas.lommatzsch, dragan.milosevic}@dai-labor.de

ABSTRACT
As today the amount of accessible information is overwhelming, the intelligent and personalized filtering of available information is a main challenge. Additionally, there is a growing need for the seamless mobile and multi-modal system usage throughout the whole day to meet the requirements of the modern society ("anytime, anywhere, anyhow"). A personal information agent that is delivering the right information at the right time by accessing, filtering and presenting information in a situation-aware matter is needed. Applying Agent-technology is promising, because the inherent capabilities of agents like autonomy, pro- and reactiveness offer an adequate approach. We developed an agent-based personal information system called PIA for collecting, filtering, and integrating information at a common point, offering access to the information by WWW, e-mail, SMS, MMS, and J2ME clients. Push and pull techniques are combined allowing the user to search explicitly for specific information on the one hand and to be informed automatically about relevant information divided in pre-, work and recreation slots on the other hand. In the core of the PIA system advanced filtering techniques are deployed through multiple filtering agent communities for content-based and collaborative filtering. Information-extracting agents are constantly gathering new relevant information from a variety of selected sources (internet, files, databases, web-services etc.). A personal agent for each user is managing the individual information provisioning, tailored to the needs of this specific user, knowing the profile, the current situation and learning from feedback.

Categories and Subject Descriptors
[**Information Systems**]: Information Systems Applications – *general, Communications Applications.*

General Terms
Algorithms, Management

Keywords
Intelligent and personalized filtering, Ubiquitous access, Recommendation systems, Agents and complex systems, Agent-based deployed applications, Evolution, Adaptation and Learning.

1. INTRODUCTION
Nowadays, desired information often remains unfound, because it is hidden in a huge amount of unnecessary and irrelevant data. On the Internet there are well maintained search engines that are highly useful if you want to do full-text keyword-search [1], but they are not able to support you in a personalized way and typically do not offer any "push-services" or in other words no information will be sent to you when you are not active. Also, as they normally do not adapt themselves to mobile devices, they cannot be used throughout a whole day because you are not sitting in front of a standard browser all the time and when you return, these systems will treat you in the very same way as if you have never been there before (no personalization, no learning). Users who are not familiar with domain-specific keywords won't be able to do successful research, because no support is offered. Predefined or auto-generated keywords for the search-domains are needed to fill that gap. As information demands are continuously increasing today and the gathering of information is time-consuming, there is a growing need for a personalized support. Labor-saving information is needed to increase productivity at work and also there is an increasing aspiration for a personalized offer of general information, specific domain knowledge, entertainment, shopping, fitness, lifestyle and health information. Existing commercial "personalized" systems are far away from that functionality, as they usually do not offer much more than allowing to choose the kind of the layout or collecting some of the offered information channels (and the information within is not personalized).

To overcome that situation you need a personal information agent (PIA) who "knows" the way of your thinking. and can really support you throughout the whole day by accessing, filtering and presenting information to you in a situation-aware matter (figure 1). Some systems exist (FAB [2], Amalthaea [3], WAIR [4], P-Tango [5], TripMatcher [6], PIAgent [7], Letizia [8], Let's Browse [9], Newt [10], WebWatcher [11], PEA [12], BAZAR [13]) that implement advanced algorithmic technology, but did not become widely accepted, maybe because of real world requirements like availability, scalability and adaptation to current and future standards and mobile devices.

In this paper we present an agent-based approach for the efficient, seamless and tailored provisioning of relevant information on the basis of end-users' daily routine. As we assume the reader will be familiar with agent-technology (see [14], [15] for a good introduction), we will concentrate on the practical usage and the real-world advantages of agent-technology. We describe the design and architecture in the following section and afterwards depict the system in section three.

2. Design of PIA: The Personal Information Agent

To meet the discussed requirements and to support the user in that matter, we designed a multi-agent system composed of four classes of agents: many information extracting agents, agents that implement different filtering strategies, agents for providing different kinds of presentation and one personal agent for each user. Logically, all this can be seen as a classical three tier application (figure 2). Concerning the information extraction, general search engines on the one hand but also domain-specific portals on the other hand have to be integrated. Additional information sources (Databases, Files, Mailinglists etc.) should also be integrated easily at run-time.

Several agents realize different filtering strategies (content-based and collaborative filtering [16], [5]) that have to be combined in an intelligent matter. Also agents for providing information actively via SMS, MMS, Fax, e-mail (push-services) are needed. A Multi-access service platform has to manage the presentation of the results tailored to the used device and the current situation. The personal agent should constantly improve the knowledge about "his" user by learning from the given feedback, which is also taken for collaborative filtering, as information that has been rated as highly interesting might be useful for a user with a similar profile as well. As users usually are not very keen on giving explicit feedback (ratings), implicit feedback like the fact that the user stored an article can also be taken into account [18].

A "keywordassistant" should support the user to be able to define queries even if he is not familiar with a certain domain. Keywords predefined by experts should be offered and also the possibility to point to a "basis-paper" serving as an example. The keywordassistant will extract automatically the most important keywords of that paper and will provide them for searching. The whole system is designed to be highly scalable, easy to modify, to adapt and to improve and therefore an agent-based approach that allows to integrate or to remove agents even at run-time is a smart choice. The different filtering techniques are needed to provide accurate results, because the weakness of individual techniques should be compensated by the strengths of others. Documents should be logically clustered by their domains to allow fast access, and for each document several "models" [19] will be built, all including stemming and stop-word elimination, but some tailored for very efficient retrieval at run-time and others to support advanced filtering algorithms for a high accuracy.

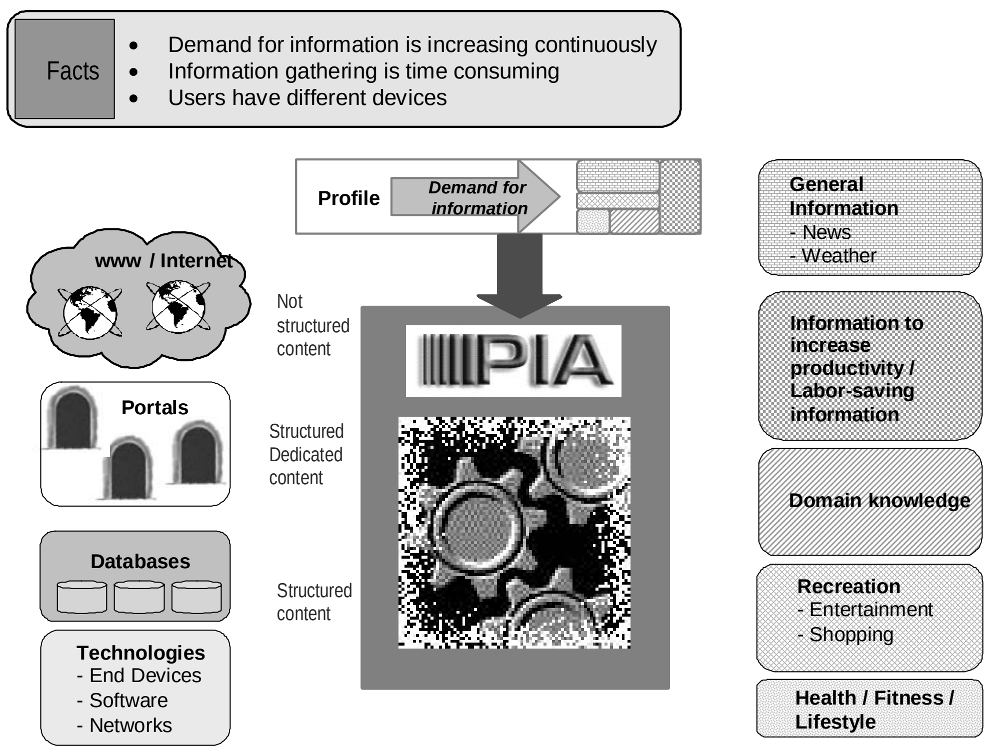

Figure 1: Demand for a personal information agent

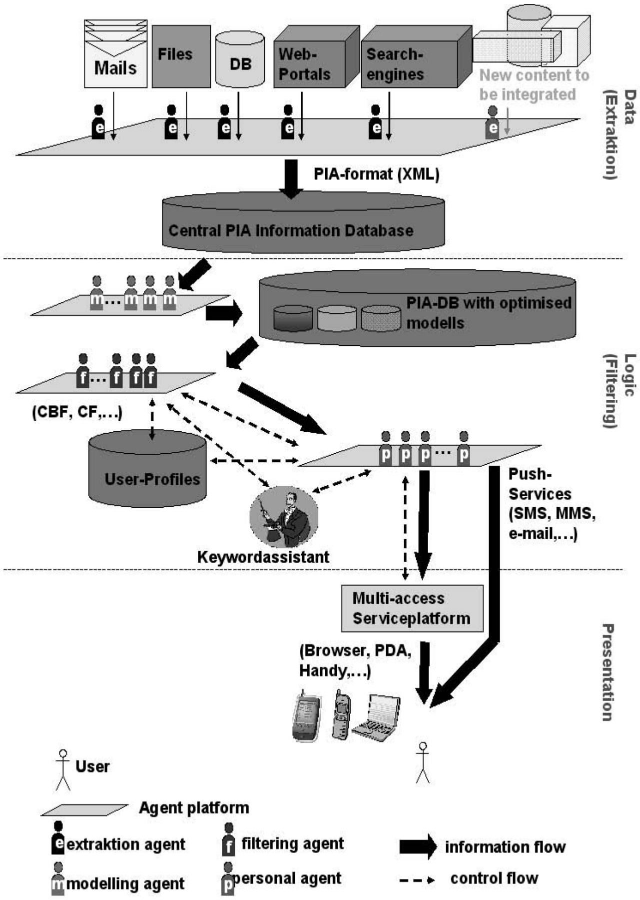

Figure 2: The PIA System seen as a three tier application

If the system notices that the content-based filtering is not able to offer sufficient results, additional information should be offered by collaborative filtering, i.e. information that was rated as interesting by a user with a similar profile will be presented.

With the "push-services", the user can decide to get new integrated relevant information immediately and on a mobile device, but for users who do not want to get new information immediately, a personalized newsletter also has to be offered. This newsletter is collecting new relevant information to be conveniently delivered by e-mails, allowing users to stay informed even if they are not actively using the system for some time.

3. Deployment and Evaluation

3.1 Overview

We implemented the system using Java and standard open source database and web-technology and based on the JIAC IV[1] agent framework [20]. JIAC IV is FIPA 2000 compliant [21], that is it is conforming to the latest standards.

It consists of a communication infrastructure as well as services for administrating and organizing agents (Agent Management Service, AMS and Directory Facilitator, DF). The JIAC IV framework provides a variety of management and security functions, management services including configuration, fault management and event logging, security aspects including authorization, authentication and mechanisms for measuring and ensuring trust and therefore has been a good choice to be used from the outset to the development of a real world application.

Within JIAC IV, agents are arranged on platforms, allowing the arrangement of agents that belong together with the control of at least one "manager". A lot of visual tools are offered to deal with administration aspects. Figure 3 shows a platform, in this case with agents for the building of different models specialized for different retrieval algorithms.

The prototypical system is currently running on Sun-Fire-880, Sun-Fire-480R and Sun Fire V65x, whereas the main filtering computation, database- and web-server and information-extraction is placed on different machines for performance reasons.

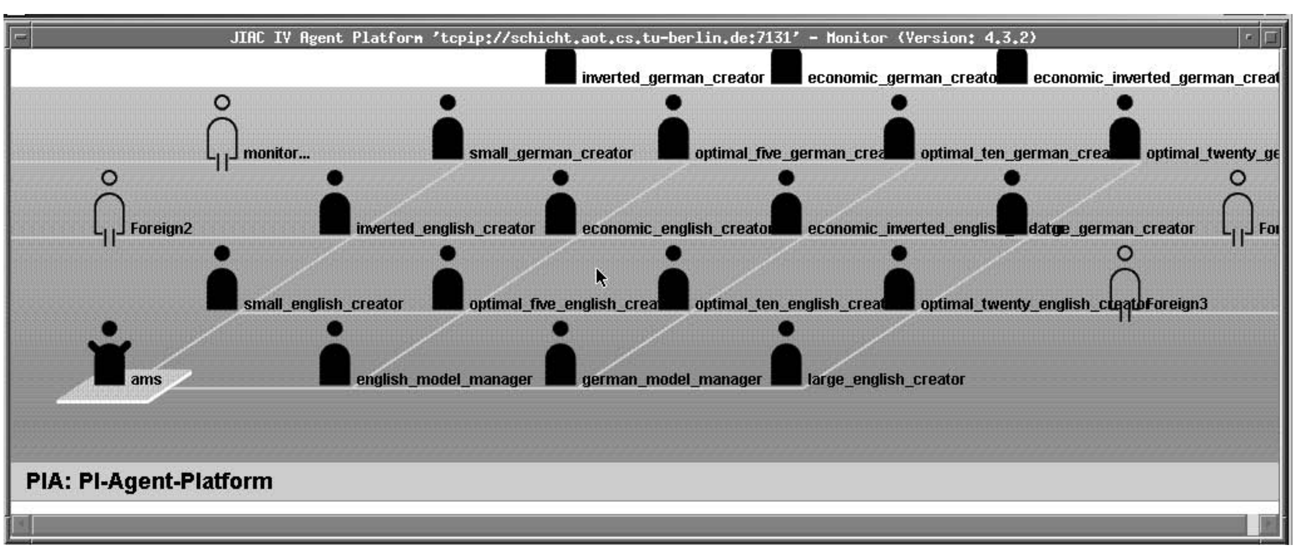

Figure 3: Several Agents are building different models specialized for different retrieval algorithms.

New content is stored, validated and consolidated in a central relational database (update-driven). Information can be accessed by WWW, e-mail, SMS, MMS, and J2ME Clients, where the system adapts the presentation accordingly, using the CC/PP (Preferences Profile) with a tailored layout for a mobile phone and a PDA (see section 3.5). The personalized newsletter and the push-services are sent via e-mail, SMS or MMS. The user can use self-defined keywords for a request for information or choose a category and therefore the system will use a list of keywords predefined by experts and updated smoothly by learning from collaborative filtering. A combination of both is also possible. The keyword assistant is able to extract the most import keywords of a given article using the term frequency inverse document frequency (TFIDF)-algorithm [22].

3.2 Gathering new Information

New information is constantly inserted in the system by information extraction agents, e.g. web-reader agents or agents that are searching specified databases or directories. Additional agents for the collection of new content can easily be integrated even at runtime, as all that is necessary for a new agent is to register himself at the system, store the extracted information at a defined database table and inform the modeling-manager agent about the insertion. As a file reader-agent is constantly observing a special directory, manual insertion of documents can be done simply by drag-and-drop and an e-mail and upload-interface also exists. Source can also be integrated by Web services. New Readers can be created using a easy-to-handle tool and another tool is enabling to conveniently observe the extraction-agents, as this is the interface to the outside that might become critical if for example the data-format of a source is changed.

3.3 Pre processing for efficient retrieval

The first step of pre processing information for efficient retrieval is the use of distinct tables in the global database for different domains like e.g. news, agent-related papers, etc. Depending on the filtering request, tables with no chance of being relevant can therefore be omitted. The next step is the building of several models for each document. Stemming and stop-word elimination is implemented in every model but different models are built by computing a term importance either based only on local frequencies, or based on term frequency inverse document frequency (TFIDF) approach. Furthermore number of words that should be included in models is different which makes models either more accurate or more efficient. Created models are indexed either on document or word level, which facilitate their efficient retrieval. The manager agent is assigning the appropriate modeling agents to start building their models but might decide (or the human system administrator can tell him) at runtime to delay latest time-consuming modeling activity for a while if system load is critical at a moment. This feature is important for a real-world application, as overloading has been a main reason for the unusability of advanced academic systems.

3.4 Filtering technology

As the quality of results to a particular filtering request might heavily depend on the information domain (news, scientific papers, conference calls), different filtering communities are implemented. For each domain, there is at least one community which contains agents being tailored to do specific filtering and managing tasks in an efficient way. Instead of having only filtering agents, each and every community has also one so called manager agent that is mainly responsible for doing coordination of filtering tasks. The coordination is based on quality, CPU, DB and memory fitness values, which are the measures being associated to each filtering agent [23]. These measures respectively illustrate filtering agent successfulness in the past, its efficiency in using available CPU and DB resources, and the amount of memory being required for filtering. A higher CPU, DB or memory fitness value means that filtering agent needs a particular resource in a smaller extent for performing a filtering task. This further means that an insufficiency of a particular resource has a smaller influence on filtering agents with a higher particular fitness value.

The introduced different fitness values together with the knowledge about the current system runtime performance can make coordination be situation aware (see also [23]) in the way that when a particular resource is highly loaded a priority in coordination should be given to filtering agents for which a particular resource has a minor importance. This situation aware coordination provides a way to balance response time and filtering accuracy, which is needed in overcoming the problem of finding a perfect filtering result after few hours or even few days of an expensive filtering.

Instead of assigning filtering task to the agent with the best combination of fitness values in the current runtime situation, manager is going to employ a proportional selection principle [24] where the probability for the agent to be chosen to do actual filtering is proportional to the mentioned combination of its fitness values. By not always relying only on the most promising agents, but also sometimes offering a job to other agents, manager gives a chance to each and every agent to improve its fitness values. While the adaptation of quality fitness value can be accomplished after the user feedback became available, the other fitness values can be changed immediately after the filtering was finished through the response time analyses. The adaptation scheme has a decreasing learning rate that prevents already learnt fitness values of being destroyed, which further means that proven agents pay smaller penalties for bad jobs than the novice ones [17].

In the case where the received filtering task cannot be successfully locally accomplished usually because of belonging to unsupported information domain, manager agent has to cooperate with other communities. While coordination takes place inside each community between manager and filtering agents, cooperation occurs between communities among manager agents. Cooperation is based on either finding a community which supports given domain or in splitting received task on sub-tasks where for each sub-task a community with good support exists.

Figure 4 presents a high level overview of the filtering framework being composed of three different filtering communities (FC), where each community has one filter manager agent (M) and different number of specialized filtering agents (F). There are two different databases (DB) with information from different domains, and they are accessed at least by one community. On the figure cooperation is illustrated as a circle with arrows which connect manager agents.

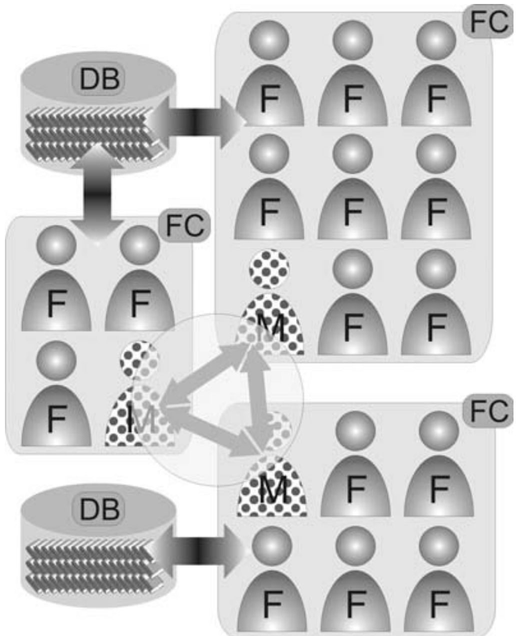

Figure 4: Filtering framework with three different communities

3.5 Presentation

As one of the main design principles has been to support the user throughout the whole day, the PIA system provides several different access methods and adapts its interfaces to the used device (Figure 5). To fulfill these requirements an agent platform ("Multi Access Service Platform") was developed that optimizes the graphical user interface for the access by Desktop PCs, PDAs and smart phones.

If the user wants to use the PIA system, the request is received by the Multi Access Service Platform (MAP). The MAP delegates the request to an agent, providing the logic for this service. In the PIA system the requests are forwarded either to login agent or the personal agent. The chosen agent performs the service specific actions and sends the MAP an abstract description of the formular that should be presented to the user. For this purpose the XML based Abstract Interaction Description Language (AIDL) has been defined. Based on the abstract description and the features of the used device the MAP generates an optimized interface presented to the user. The conversion is implemented as a XSLT transformation in which the optimal XSLT style sheet is selected based on the CC/PP information about the user's device.

This approach simplifies the creation of optimized user interfaces for different devices. The abstract interface description can be easily transformed into HTML, PDA optimized HTML or WML. If the user wants to have a voice interface, a style sheet for converting the abstract user interface description into VoiceXML has to be added to the MAP. Additional changes at the PIA system are not needed.

Beside of the features provided by MAP the design of the user interface must create an easy to use system even on devices with a tiny screen and without a keyboard. That is why the PIA interface provides additional navigation elements on complex forms and minimizes the use of text input fields. New results matching a defined request are presented first as a list of short blocks containing only title, abstract and some meta-information (as this is familiar to every user from well-known search-engines). This information is also well readable on PDAs or even mobile phones. Important articles can be stored in a repository. This allows the user to choose the articles on his PDA he wants to read later at his desktop PC.

Depending on the personal options specified by the user, old information found for a specific query may be deleted automatically step by step after a given time, that is, there is always up to date information that is presented to the user (we call this "smart mode"). This is for example convenient for getting personalized filtering news. The other option is to keep that information unlimited ("global mode") for a query for e.g. basic scientific papers.

For the "push-services" (that is, the system is becoming active and sending the user information without an explicit request), the user specifies his working time (e.g. 9 am to 5 pm). This divides the day in a pre-, work, and a recreation slot, where the PIA system assumes different demands of information. For each slot an adequate delivering technology can be chosen (e-mail, sms, mms, fax or Voice). If you decide to subscribe to the personalized newsletter, new relevant information for you will be collected and sent by e-mail or fax once a day with a similar layout and structure for convenient reading if you have not seen it already by other pull- or push services. Therefore you can also stay informed without having to log into the system and if you do not want to get all new information immediately.

Figure 5: Information accessed by browser or tailored for presentation on a PDA or a mobile phone

4. Conclusion and future work

The implemented system has an acceptable runtime performance and shows that it is a good choice to develop a personal information system using agent-technology based on a solid agent-framework like JIAC IV. Currently, PIA system supports more than 120 different web sources, grabs daily around 3.000 new semi-structured and unstructured documents, has almost 500.000 already pre-processed articles, and actively helps about fifty scientists related to our laboratory in their information retrieval activities. Their feedback and evaluation is a valuable input for the further improvement of PIA. In the near future we plan to increase the number of users to thousands, and therefore we plan to work on the further optimization of the filtering algorithms to be able to simultaneously respond to multiple filtering requests. Also, we think about integrating additional services for the user that provide information tailored to his geographical position (GPS), a natural speech interface and innovative ways to motivate the user to give precise explicit feedback, as the learning ability of the system is depending on that information.

5. REFERENCES

[1] Brin, S.; Page, L.: *The anatomy of a large-scale hyper textual (Web) search engine*, Proc. 7th International World Wide Web Conference on Computer Networks, 30(1-7), pp. 107-117, 1998.

[2] Balabanovic, M.; Yoav, S.*: FAB: Content-Based Collaborative Recommendation*, Communication of the ACM, Volume 40, Number 3, pp. 66-72, 1997.

[3] Moukas, A.: Amalthaea*: Information Discovery and Filtering using a Multi agent Evolving Ecosystem*, Practical Application of Intelligent Agents & Multi-Agent Technology, London 1998.

[4] Zhang, B.; Seo, Y.: *Personalized Web-Document Filtering Using Reinforcement Learning*, Applied Artificial Intelligence, Volume 15 Number 7, pp. 665-685, 2001.

[5] Claypool, M.; Gokhale, A.; Miranda, T.; Murnikov, P.; Netes, D.; Sartin, N.: *Combining Content-Based and Collaborative Filters in an Online Newspaper*, ACM SIGIR Workshop on Recommender Systems, Berkeley, CA, August 19, 1999.

[6] Delgado, J.; Davidson, R.: *Knowledge bases and user profiling in travel and hospitality recommender systems*, In Proceedings of the ENTER 2002 Conference, Innsbruck, Austria, January 22-25 2002, Springer Verlag, pp. 1-16.

[7] Kuropka, D.; Serries, T.: *Personal Information Agent*, Informatik Jahrestagung 2001, pp. 940-946.

[8] Lieberman, H.: Letizia: *An Agent That Assists Web Browsing*, International Joint Conference on Artificial Intelligence, Montreal, August 1995.

[9] Lieberman, H.; Van Dyke, N,; Vivacqua, A.: *Let's Browse: A Collaborative Browsing Agent*, Knowledge-Based Systems, 12(8), 1999, pp. 427–431.

[10] Sheth, B.: *A Learning Approach to Personalized Information Filtering*, M.S. Thesis, MIT- EECS dept, USA, 1994.

[11] Joachims, T.; Freitag, D.; Mitchell, T.: *WebWatcher: A Tour Guide for the World Wide Web*, In IJCAI (1), 1997, pp. 770-777.

[12] Winiwarter, W.: *PEA - A Personal Email Assistant with Evolutionary Adaptation*, International Journal of Information Technology, Vol. 5, No. 1, 1999.

[13] Thomas, C.; Fischer, G.: *Using agents to improve the usability and usefulness of the world wide web*. In Proceedings of the Fifth International Conference on User Modelling, pages 5--12, 1996.

[14] Jennings, N; Wooldridge, M: *Agent-oriented software engineering*, Handbook of Agent Technology (ed. J. Bradshaw). AAAI/MIT Press, 2000.

[15] Sesseler, R.; Albayrak, S.: *Agent-based Marketplaces for Electronic Commerce*, International Conference on Artificial Intelligence, IC-AI 2001.

[16] Resnick, P.; Neophytos, J.; Suchak, M.; Bergstrom, P.; Riedl, J.: GroupLens: *An open architecture for collaborative filtering of net news*, Proceedings ACM Conference on Computer-Supported Cooperative Work, pp. 175-186, 1994.

[17] Albayrak, S.; Milosevic, D.: *Self Improving Coordination in Multi Agent Filtering Framework*, IEEE/WIC/ACM International Joint Conference on Intelligent Agent technology (IAT 04) and Web Intelligence (WI 04), Beijing, China, September 2004., (to appear).

[18] Nichols, D.: *Implicit Rating and Filtering*, Proc. Fifth DELOS Workshop on Filtering and Collaborative Filtering, Budapest, Hungary, 10-12 November, ERCIM, pp. 31-36, 1997.

[19] Tauritz, D.: *Adaptive Information Filtering: concepts and algorithms*, Ph.D. dissertation, Leiden University, 2002.

[20] Fricke, S.; Bsufka, K.; Keiser, J.; Schmidt, T.; Sesseler, R.; Albayrak, S.: *Agent-based Telematic Services and Telecom Applications*, Communications of the ACM, April. 2001.

[21] Foundation for Intelligent Physical Agents, www.fipa.org, 2004.

[22] Jing, L.; Huang, H.; Shi, H.: *Improved Feature Selection Approach TFIDF in Text Mining*, Proc. 1st Internat. Conference on Machine Learning and Cybernetics, Beijing, 2002.

[23] Albayrak, S; Milosevic D.: *Situation-aware Coordination in Multi Agent Filtering Framework*, The 19th International Symposium on Computer and Information Sciences (ISCIS 04), Antalya, Turkey, October 2004., (to appear).

[24] Zbigniew, M.; Fogel, D.: *How to Solve It: Modern Heuristics*, Springer-Verlag New York, Inc., New York, NY, 2000.