Educational Technology International
2005, Vol. 6, No. 1, pp. 65-79.

An Information Filtering Agent in a Flexible Message System

Youngcook JUN*
(Sunchon National University)

Norio SHIRATORI
(Tohoku University)

Abstract

In a widely distributed environment, many occasions arise when people need to filter informationwith email clients. The existing information agents such as Maxims and Message Assistant have capabilities of filtering email messages either by an autonomous agent or by user-defined rules. FlexMA, a variation of FAMES (Flexible Asynchronous Messaging System) is proposed as an information filtering agent. Agents in our system can be scaled up to adapt user's various demands by controlling messages delivered among heterogeneous email clients. Several functionalities are split into each agent in terms of component configuration with the addition of multiple agents'cooperation and negotiation. User-defined rules are collected and executed by these agents in a semi-autonomous manner. This paper demonstrates how this design is feasible in a flexible message system.

Keywords : CMC tool, network interaction, software agent

* Sunchon National University, Department of Computer Education, 315 Maegok-dong Sunchon city, Chonnam, Korea, 540-742, ycjun@sunchon.ac.kr

Ⅰ. Introduction

In distributed information environments, information flow reveals potential problems in terms of network states and human user's demands. Intelligent agent technology provides a mechanism for information systems to act on behalf of their users. To be able to adapt huge data of the Internet to the needs of a user, a tool such as a personalized agent for Internet usage content to its user. Thus building users' profiles and personalization based on profiles becomes the key issues in agent technology.For more detailed review of such technology, refer to the following papers (Konstan , et al., 1997; Shahabi, 2003; Y-W. Seo, & B-T. Zhang, 2000).

Agent technology can be applied to Web-based hypermedia learning system. Henze (2000) proposes the concept and realization of an adaptation component for an open, adaptive hypermedia system which implements advanced teaching strategies and enables integration and adaptation of learning material found in the WWW. The adaptation component uses an indexing concept for describing the content of the various information resources. This indexing concept is also taken as a base for constructing a model of the application domain. A Bayesian inference mechanism calculates estimations about the user's knowledge on top of this domain model.

Various needs requirements are taken place when the number of incoming and outgoing messages is increasing everyday. For example, a user may want to transfer incoming messages whose subject fields contain “compute education” into a separate folder. This task can be carried out automatically by an agent on behalf of the user (Nwana, 1996). Message Assistant is such a program that belongs to a category of semi-autonomous agents. These agents mainly filter email, news articles, web documents, etc., based on the user-defined rules.

Another category of information filtering agents is classified as autonomous agents which gradually learn the patterns of a user's behavior regarding email messages. To make agents be intelligent, researchers employ algorithms such as memory-based or inductive learning. In this category, there are several agents that deal with message filtering, news articles, meeting scheduling, entertainment recommendations, web document filtering and indexing (Petrie, 1996).

In this article, we describe a different approach based on a flexible network framework in terms of ADIPS (Agent-based Distributed Information Processing System). In our case, electronicmessages are treated independently by different email clients such as Eudora, Outlook Express, InternetMail, etc. By agentification of existing email clients, our system not only adapts the user's specific demands but controls the flow of messages, taking advantages of controllability, scalability and intelligence of ADIPS (Fujita, 1996). By focusing on how to scale up the current email agents with user-defined rules, this paper shows implementation issues and describes the system's empirical behavior compared with other information filtering agents.

Ⅱ. Information filtering agents for electronic messages

There are several approaches of adapting a user's various demands when email client is used. Eudora was experimented by agent-based approaches; semi-autonomous (Message Assistant) and autonomous (Maxims) agents. In this section, we first compare Message Assistant and Maxims and further point out limitations of Message Assistant. ADIPS-based email agent called FAMES is described in sequence.

1. Message Assistant and Maxims

Message Assistant was aimed at adding extra functions such as filtering messages and hypertextually linking messages to Eudora 2.x versions. In Message Assistant, users can define their own rules in IF-THEN format. With these rules, Message Assistant carries out tasks of replying incoming messages, assigning priorities, transferring mails into different folders, etc. Once created, these rules explicitly perform tasks for the user either by hand or automatically. As Pattie(1994) points out, semi-autonomous agent such as Message Assistant does not fully satisfy competence criterion. In other words, the user of Message Assistant needs not only to initiate when to create rules but also to maintain those rules over time. One advantage of semi-autonomous agents is that the user can trust his or her own programmed rules in a satisfactory way.

In the category of autonomous agent, Maxims (Pattie, 1994) recognizes the user's patterns by monitoring and learns those patterns by machine learning technique. Provided that there are a substantial amount of repetitive different behavior for different users, autonomous agent can program itself to catch the user's habits and preferences. This approach requires less work from the end-user and gradually adapts to the user over time. Maxims acquires its competence by monitoring the user's actions and finding regular and recurrent patterns with direct and indirect feedback. Besides these sources, an autonomous agent can automate the process of acquiring competence by asking for advice from other agents with the same task. Maxims used memory-based reasoning in a way that the agent memorizes all the situations of message flow taken by the user. Based on the features of situations and actions, memory-based reasoner generalizes particular patterns by finding close matches among nearest neighbors (examples).

The common features of Message Assistant and Maxims are as follows. The two systems employed Apple Events to communicate with Eudora in Macintosh platform.As for a plug-in system with Eudora, Message Assistant was first developed in HyperCardand converted in C later, whereas Maxims was programmed in MCL (Macintosh Common Lisp). On the other hand, Message Assistant was not easily scaled up with substantial amount of user-defined rules. Some conflicts among rules might occur depending on the sequence of rule execution. Maxims also has a limitation of acquiring competence in the initial stage. The process of acquiring competence is slow since it usually takes a large number of examples. To increase bottleneck of competence problem, cooperation among multiple agents can provide an alternative by exchanging information regarding different users in a learning community.

2. Limitations of Message Assistant

Message Assistant as a semi-autonomous agent behaves in a rigid manner plugged into Eudora. With a userinterface, the user can create his or her own rules to filter messages automatically or manually. The current version of Message Assistant lacks flexibility and cooperative work among multiple agents.

Since Message Assistant is particularly plugged into the Macintosh version of Eudora, it cannot be applicable to other software or Windows version of Eudora. Also, rule execution is carried out with interprocess communication protocol, called AppleEvents once email messages are arrived, which usually degrades the execution speed from a user's perspectives. Thus, the Message Assistant lacks a flexible mechanism of filtering messages and reveals performance problem. The second problem results from lack of cooperative mechanism among multiple agents. Since Message Assistant is closely tied with each user's email usage, knowledge about message filtering is acquired separately on each site. In this way, it is not possible for agent managers to gather relevant advice or knowledge from other agents. This prevents multiple agents from sharing knowledge, which reduces system's scalability and distributed intelligence.

Ⅲ. FAMES in the framework of ADIPS

The ADIPS framework is an agent-based computing infrastructure proposed by the authors, aiming at constructing flexible distributed systems (Fujita, 1996). ADIPS framework has the following characteristics to realize the agent-oriented facilities of the system we need; (i) A system is autonomously organized or reorganized in user-driven and event-driven manner, (ii) Agents use domain expertise drawn from designer, administrator and operator of a distributed system, (iii) Legacy computational processes or applications can be used as the reusable components by agentified operation of processes. When user agents request tasks to the ADIPS Repository on users' demands, the most suitable agents in ADIPS Repository are bided by extended contract net protocol (Smith, 1980), and the selected agents are instantiated in a personal user environment called ADIPS Workspace. The instantiated agents in Workspace communicate with each other and adapt to users' demand flexibly.

In this ADIPS framework, FAMES can control message flow, and add extra functions such as message circulation and message cancellation to various email clients such as Eudora, Outlook Express and InternetMail (Sekiba, 1998). By interrupting SMTP and POP, the system sends the list of circulation recipients to the target agent that responds to carrying out circulation task to the next recipient. In this manner, negotiation might occur between two agents once a particular recipient might be absent. The absence of the recipient prevents the message from being circulated without his or her endorsement of the message. The remarkable feature of FAMES is the capability of adding extra functions to different email clients in a uniform way with user-defined messaging configuration. Moreover, this system can bridge functionalgaps among various email clients by abstracting specific functions.

Written in Java, FAMES can achieve the following mechanisms rooted from the ADIPS framework:

(M1) Realization of adaptive service reconfiguration: A personal environment of user is realized as an organization of agents. According to the agents' characteristics such as self-recognition and adaptability, agents can recognize the environment autonomously. These agents reconfigure their organization, and provide the necessary and sufficient services to users.

(M2) Realization of user-centered flexible messaging: Making whole system as an organization of agents, the message flow can be controlled with respect to users' demand via cooperative problem solving capability of agents.

(M3) Realization of function abstraction: Reforming a conventional e-mail client as an agent (agentification operation), each e-mail clients' peculiar functions are abstracted by the agentification operation, and can be connected together with common interfaces among agents.

Figure 1 illustrates the ADIPS based architecture of FAMES. Each agent in FAMES has its own role concerning message handling. The full-time agents reside in the workspace permanently and control the organizational activities of agents. While part-time agents are instantiated from repository on demand in run-time, all the agents dynamically operate according to the situations of user requirements and system resource availability. MMA (Message Manager Agent) manages agent organization in a personal environment. UIMA (User Interface Manager Agent), FCMA (Flow Control Manager Agent) and MTMA (Message Transfer Manager Agent) are function oriented manager agents which are responsible for user interface, message flow control and message transfer, respectively. Under the control of these manager agents, part-time agents such as UIA (User Interface Agent), FCA (Flow Control Agent) and MTA (Message Transfer Agent) are dynamically organized. UIA is defined by agentification of existing e-mail client software, and additional user interfaces are needed for some peculiar functions. FCAs

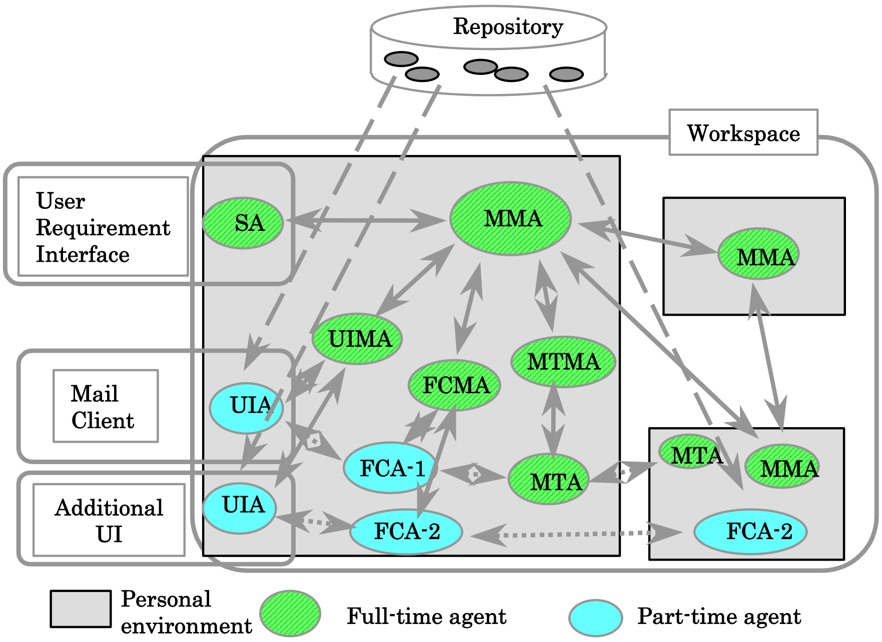

Figure 1. Agent organization for FAMES

handle message flow control such as circulation and cancellation of messages. SA (Secretary Agent) is an agent that gathers user requirements. These agents have cooperative problem solving capability with same functional agents in other personal environments with sophisticated negotiation protocols.

Ⅳ. Design of FlexMA using FAMES

1. Extending Message Assistant with FAMES

Message Assistant (MA) can be easily extended based on FAMES by resolving its problems described in section 2. Firstly, function abstraction feature of FAMES solves the problem of platform dependency of MA. By dividing and agentifying the message filtering function of MA in accordance with ADIPS framework, it can work with not only Eudora on Macintosh environment, but also other e-mail clients on heterogeneous platform. This advantage is clear because both e-mail client and filtering function are treated as agents in ADIPS workspace, and they can cooperatively work. Secondly, the system performance can be improved by adaptive service reconfiguration feature of the FAMES. By instantiating the most appropriate agents, the most adequate filtering function can be used according to the user requirement and system resource status in a flexible manner. For instance, when a user requests functional rich filtering regardless of response time on powerful platforms such as workstation, rule based filtering agents with full functionality will be instantiated. While a user requires quick response time without intelligent feature of filtering, the light weight agents may be selected.

Moreover, cooperation abilities of agents will make the MA more sophisticated in terms of filtering capability. Because the original MA is closed in a personal messaging environment, filtering capability is specialized but limited to a particular user. However, in general, a user participates in several societies on the net, the knowledge on information selection can be shared in a group. To accelerate such knowledge sharing and exchanging, cooperation among agents is necessary. Constructing MA based on FAMES, filtering knowledge of different users can be utilized in a group of members who have same interests.

2. Design of Message Assistant with FAMES

Figure 2 represents the architecture of FAMES-based MA. The original MA functions are divided into three parts, namely knowledge acquisition interface, mail client and filtering function. Each function is then agentified to incorporate features of the FAMES.

(i) Extension of SA: Knowledge acquisition interface, whichhad a role of rule editor in MA, is integrated as a user interface of SA (Secretary Agent).

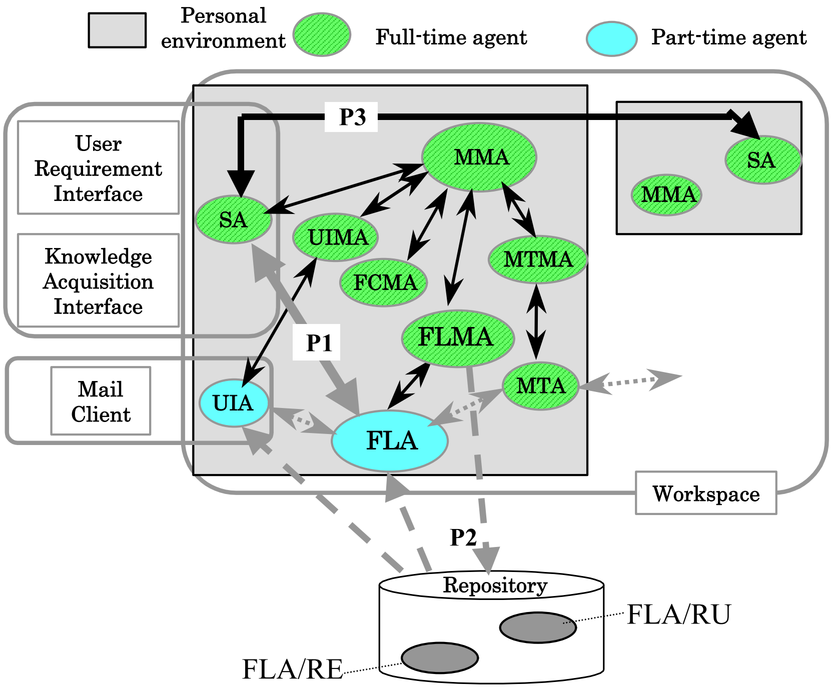

Figure 2. Architecture of FAMES-based MA

Using the interface, SA acquires the user model on message filtering. Moreover, SA is extended to keep and maintain the knowledge.

(ii) Agentification of mail clients: The e-mail client such as Eudora has already been agentified into UIA in FAMES. Thus, flexible combination of e-mail clients and filtering functions can be accomplished.

(iii) Filtering function: The actual filtering function is realized by two class agents, FLA/RU and FLA/RE. FLA/RU stands for a Filtering Agent with rule-base system, whichhas intelligent and powerful filtering capability. FLA/RE stands for Reactive Filtering Agent whose response is quick, but it's reasoning ability is limited. To utilize rule-based reasoning capability in ADIPS agents, extension of agent description language based on the production model has been developed (Bunne, 1998). In the new version of

ADIPS framework, JESS (Java Expert System Shell) can be available in Domain Knowledge base of the agent to provide powerful reasoning capability. One of these two types of agents is selected and instantiated to Workspace according to the user requirement and system resource status. To control the behavior of instance agent of these two class agents (FLA: Filtering Agent), FLMA (Filtering Manager Agent) resides as a full-time agent in Workspace.

(iv) Protocols: To integrate MA in FAMES, three types of protocols are mainly used. P1 in figure 2 is a new protocol between SA and FLA, and in charge of transferring and tuning the filtering knowledge in FLA. FLA can be an instance agent of either FLA/RE or FLA/RU; this implies that tuning of knowledge transition is needed in the initial sequence of FLA. For example, when system resource is not enough to perform condition matching of all the rules in a reasonable time limit, only the selected rule sets can be transferred from SA to FLA, and no conflict resolution may be specified. P2 in figure 2 is a contract-net based protocol (Smith, 1980) that is normally used in ADIPS framework to select the suitable agent in Repository. P3 is an inter-SA protocol that enables multiple agents to exchange filtering knowledge among those agents, which have same interests in a particular domain.

3. Agent behavior of FlexMA

The agents dedicated to FMAES-based MA in figure 2 work as follows;

(1) Filtering knowledge acquisition: SA interacts with its user using knowledge acquisition interface, and keeps the filtering knowledge in its knowledge base.
(2) User requirement acquisition: When a user requests to use the messaging system, user requirement is issued through user requirement interface, and SA translates the user-level requirement to inter-agent communication language. TaskAnnouncement message is issued from SA to MMA. In the

TaskAnnouncement, requirements for filtering in terms of capability, response and are included.

(3) Task decomposition by MMA: MMA divides the TaskAnnouncement into some pieces of TaskAnnouncement. MMA also sends information on the system's status such as CPU utilization to system sensor agent. Based on such information, MMA propagates the TaskAnnouncement to appropriate manager agents. TaskAnnouncement concerning filtering is issued to FLMA.

(4) Task announcement to Repository: FLMA issues the TaskAnnouncement to the Repository. In Repository, class agents begin with negotiating to select the most adequate agents in a given condition represented in the TaskAnnouncement. Either FLA/RE or FLA/RU can be instantiated as FLA in the Workspace.

(5) Filtering knowledge transfer: In the initial sequence of FLA, SA and FLA cooperatively work to give appropriate filtering knowledge under the condition of system resource and user requirement, using protocol P2. After transferring knowledge, FLA begins to perform filtering function.

(6) Message filtering: An outgoing message is transferred from UIA to FLA, filtered by FLA, passed to MTA, and sent to other environment. After spooling incoming messages in MTA, FLA and MTA negotiate to filter the spooled messages based on the filtering knowledge. By requesting from UIA, the remaining messages are delivered to user via UIA.

(7) Reconfiguration of agent organization: When some changes occur according to user requirements or system situations, FLA can be deleted or replaced with other agent. For instance, when system resources are suddenly degraded, the rule-based FLA can be dissolved and a reactive FLA may be instantiated instead.

Ⅴ. Discussions

FAMES is a new approach, consisting of autonomous and collaborative software agents. Not being plugged into a particular email client, FAMES resides in the middle of message flow relaying messages from email clients to email hosts. By focusing on the problem of network controllability, scalability and intelligence, FAMES was developed in the framework of ADIPS.

Based on FAMES, FlexMA was designed and developed in this paper. In FlexMA, Message Assistant is expanded by including rule agents associated with agents in the FAMES. For each email client side, several agents are usually generated in the Repository. The behavior of all the agents except rule agents such as FLA/RE and FLA/RU is similar to those of FAMES. On the other hand, FlexMA adopts message-filteringconcept similar to Message Assistant so that this approach takes advantage of scalability rooted from ADIPS. In addition, our system can easily control network state by controlling SMTP and POP for the incoming and outgoing messages. However, Maxims-like autonomous memory-based reasoning method is not yet experimented. Thus, FlexMA does not focus on implementing intelligent agents; our approach rather shows how our system resolves Message Assistant's scalability problem and Maxims' competence problem via collaborative software agents using a contract-net based protocol.

In short, FlexMA is designed and tested to extend Message Assistant based on FAMES framework by trying to resolve Message Assistant's limitations; flexibility, agent cooperation, and scalability. By applying FAMES to Message Assistant, we built a hybrid FlexMA that turns out to bring several research topics in agent-based network information processing. We are trying to implement a prototype system of FlexMA on the latest version of ADIPS framework, reviving design methods and development techniques used in MA. As for future research, these topics are directly related to knowledge sharing and reuse, protocol development among agents, knowledge competence and trust, etc.

One of the advantages of using FlexMA can be applied to filter unsolicited bulk e-mails that produce an insurmountable problem throughout the world. It has recently been argued that serveral classifiers can be used to filter spam mails (Androutsopoulos, et al., 2000; Hall, 1998). In order to filter such unwanted messages, anti-spam filters can be designed using Bayesian classifier algorithm or others from the client side. What the FlexMA design indicate in this direction is that multiple agent cooperation can indentify spam mails by collecting user's information ahead of time based on a contract-net based protocol before reaching the client email software.

References

T. Bunne, H. Hara, S. Fujita, K. Sugawara, T. Kinoshita, & N. Shiratori. (1998). The extension of the ADIPS agent description language based on the production model. *IEICE Technical Report, 98*(200), 9-14. (in Japanese)

S. Fujita, K. Sugawara, T. Kinoshita, & N. Shiratori.(1996). Agent-based Architecture of Distributed Information Processing Systems. *Trans. IPS. Japan, 37*(5), 840-852, (in Japanese)

R. Hall (1998). How to avoid unwanted email. *Communications of the ACM, 41*(3), 88-95.

N. Henze (2000) *Adaptive Hyperbooks: Adaptation for Project-Based Learning Resources.* PhD Thesis at Univeritt Hannover.

JESS. *The Java Expert System Shell.* Url at http://herzberg.ca.sandia.gov/jess/

Y. Jun, J. Levin, M. Jacobson. (1994). *The Message Assistant: A communication tool for educational networks.* Paper presented at KRNet '94, Seoul Korea.

Y. Jun, J. Levin, M. Jacobson. (1995). *Client-server tools for educational network interactions: The Message Assistant and the Learning Resource Server.* Paper presented at KRNet '95, Seoul Korea.

J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon, & J. Riedl (1997). GroupLens: applying collaborative filtering to Usenet news. *Communications of the ACM, 40*(3), 77-87.

I. Androutsopoulos, J. Koutsias, K. Chandrinos, G. Paliouras, & C. Spyropoulos (2000). *An evaluation of naive Bayesian anti-spam filtering.* In Proceedings of the Workshop on Machine Learning in the New Information Age, 11th European

Conference on Machine Learning, Barcelona, Spain.

P. Maes. (1994). Agents that reduce work and information overload. *Communications of the ACM, 37*(7), 31-40.

H. Nwana. (1996). Software agents: An overview. *Knowledge Engineering Review, 11*(3), 205-244.

C. Petrie. (1996). Agent-based engineering, the Web, and intelligence. *IEEE Expert, 11*(6), 24-29.

J. Sekiba, G. Kitagata, T. Suganuma, T. Kinoshita, K. Okada, & N. Shiratori. (1998). *FAMES: Design and Implementation of Flexible Asynchronous Messaging System.* Proceedings of The International Conference on Software in Telecommunications and Computer Networks (SoftCOM98).

Y-W. Seo, & B-T. Zhang (2000). *Learning user's preferences by analyzing Web-browsing behaviors.*Proceedings of the fourth international conference on Autonomous agents, p.381-387, June, Barcelona, Spain.

C. Shahabi (2003). *Web Information Personalization: Challenges and Approaches.* Presented at the 3nd International Workshop on Databases in Networked Information Systems (DNIS 2003), Aizu-Wakamatsu, Japan.

R. G. Smith (1980). The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Trans. on Computers, 29*(12), 1104-1113.

Youngcook JUN

Associate Professor, Dept. of Computer Education, Sunchon National University. Interests: WBI, Adaptive Hypermedia, and Computer Algebra Systems for e-learning

E-mail: ycjun@sunchon.ac.kr

Norio SHIRATORI

Tohoku University.

E-mail: norio@shiratori.riec.tohoku.ac.jp