Vol. 43 No. 3 IPSJ Journal Mar. 2002

Regular Paper

Autonomous Integrated Services Allocation

Hélène Arfaoui† and Kinji Mori†

Even though computing and telecommunications technologies are continuously improving, critical business applications such as electronic commerce foster an urgent need for high-assurance information service systems in rapidly evolving situations. One solution has been proposed in the form of a distributed service replica network sustained by Push/Pull mobile agents, called Faded Information Field (FIF). In this system, user requests are autonomously driven by Pull-MA in charge of finding the relevant service. In the case of a mono-service request, the system is designed to reduce the time needed for users to access the information and to preserve the consistency of the replicas. But when the user requests joint selection of multi-services, synchronization of atomic actions and timeliness have to be assured by the system. This paper proposes a new technology to increase the potential of service consumption by adding new classes of services, called *integrated services*. An autonomous integration protocol for heterogeneous FIFs is established on the basis of correlation ratios on the three views of content, property, and category of information services. The integrated services have been shown to efficiently reduce access time and increase service consumption.

1. Introduction

Requirements for information systems continue to become more stringent and complex to cope with changing telecommunication needs and to adapt to new technological advances[1]. The Internet, for example, is continuously subject to modifications and evolutions that affect its functioning. Users have varying informational demands that service providers need to cope with. With the emergence of electronic business, informational content is continuously updated according to the current sale offers. The network suffers from constant alterations. Wireless data networks, in particular, are overwhelmed by fluctuant connection stability, and unpredictable availability in an environment with high latency and low bandwidth. As perceived by Internet users, frequent unavailability, poor responsiveness, and data corruption can tarnish a provider's reputation and decrease its service consumption[2]. This situation consequently fosters an urgent need for high-assurance[3]. It is therefore necessary that in continuously changing situations, dynamic binding between the actors be maintained and that reliability, availability, and safety be guaranteed during transactions.

Service utilization varies in quantity and quality, and consequently the complete service offer is generally irrelevant to users. Analyzing the server logs of some HTTP servers, a study[4),5)] has shown that more than 80% of the generated traffic was generally caused by accesses to less than 6% of the actual available information on the central server. In another study conducted by Gwertzman and Seltzer[6], analysis of service consumer worldwide revealed well-defined concentration areas. A selective replication of server information toward its consumer location is therefore one good way to reduce network traffic and load-balance service providers. However, the success of introducing service replicas at distinct sites depends on the ability of the system to maintain consistency between replicas and to dynamically direct users to the right node. Obviously, the previously proposed static techniques are not applicable to rapidly changing environments. A better approach is to integrate users and SPs in one heterogeneous system where each collaborative entity is autonomous. Such a system has been established in the form of an innovative architecture called *Faded Information Field* (FIF) and sustained by a *Pull/Push* mobile agent infrastructure.

In FIF, the service provider allocates its popular information to adjacent nodes, closer to service users. For mono-service requests, this architecture manages to reduce user access time while preserving a consistent and maintainable replicated informational area. However, most requests on the Internet involve two or more services at the same time. If a request can be split into two mono-service requests, different Pull-MAs can be sent in parallel and execute their

† Tokyo Institute of Technology

tasks. For complex multi-service requests to be performed, arduous synchronization operations are necessary, leading to further delay. Additionally, such synchronization is usually impossible when a critical decision needs to be taken when purchasing, for example, an outstanding product or when making a bid. In this paper, we propose that services available in a common replication zone be integrated when they can be demonstrated to be highly correlated. We define how functionally and semantically close two services are on the basis of the three views of their contents, properties, and category, and describe how two FIFs can be dynamically integrated to optimize their synergy in rapidly evolving situations.

The remainder of this paper is organized as follows. Section 2 explains the need for integrating Internet services into the global market. Section 3 presents the concept of autonomous information service and then introduces the Faded Information Field concepts and architecture. Section 4 proposes three views for measuring the synergistic effect of service integration. Section 5 then exposes how an autonomous integration can be achieved and presents the simulation results. Finally, section 6 outlines our conclusions.

2. Motivation and Approach

2.1 Coping with the Competition

On the Internet, many application requirements are becoming more severe, in response to the fierce competition taking place between service providers. To strengthen their business position, service providers have been diversifying their offers with value-added services such as advisory services based on collaborative filtering or customized front pages. In the electronic commerce area, for instance, *Amazon.com* supplies recommendations to users based on what other users with the same profile have bought at their site. Search engines have also been gathering satellite-services such free e-mail accounts and on-line news, to render their offers more attractive. Similarly, as an example of vertical integration, virtual manufacturing enterprises[7] have been widely addressed by the research community. As the number of Internet services continues to grow exponentially, it therefore becomes more and more necessary to make a Web site a pole of attraction in order to survive as a service provider.

2.2 Exigency of Demands

With the democratization of Internet usage, users are making more and more diversified and complex requests, involving various criteria and services. A typical example is the hotel reservation/airplane ticket-booking problem. A user is looking for a return flight to Tokyo during his or her vacation, and wants to book a hotel in this city during the same period. Further constraints about prices and possible time periods can also be requested. Obviously, this search induces time-consuming coordination and synchronization of distributed transactions between multiple SPs. However, ignoring this type of request will undoubtedly lead to a loss in service consumption. Coping with the exigency of demands is one of the most critical challenges that SPs need to face.

2.3 Globalization

Hitherto, global service integration has been mainly undertaken by Internet portals to enhance their service offers. *Yahoo.com*, for instance, integrates auction services and on-line shopping along with its traditional search engine. However in rapidly changing environments, such centralized approaches may face saturation and faults affecting the availability, and responsiveness of services. Service integration has to be adapted to continuously varying user demand in terms of quantity and quality requirements and service offers.

In rapidly evolving environments, static and centralized approaches are mostly inappropriate: diversity and distribution have to be preserved to ensure the system requirements. Our approach is to discover how on-line companies sharing common cache replicas can become trading partners, enriching each other's offers, in a dynamic way. In this way, it may be possible not only to decrease the service access time of the users, but also to satisfy a broader range of users. Furthermore, service integration loads off SPs by combining their correlated information on the same node. In autonomous information service systems, autonomy is the milestone for integrating services. It is achieved through a distributed architecture of pruned replicas, called *Faded Information Field*.

3. Autonomous Information Service System

Information service systems are mainly constituted of two actors with conflicting goals and requirements: *service providers* (SP), who bring the informational contents, and *users*, who consume the information services. SPs provide information dynamically, promoting their new information services. Users require timeliness and relevance in varying degrees. The main objective of such heterogeneous system is consequently to ensure the satisfaction of both parties' requirements.

In dynamically changing environments, one high-assurance design approach is to have the architecture provide a way to expand, maintain, and integrate services on-line. The Autonomous Information Service System, inspired by research on demand-oriented access to services[8] and ADS[9], takes this approach.

It ensures the following properties of the system:

- *Flexibility*. The system is self-adapting to new situations.
- *Timeliness*. Dynamic interactions between service consumption and provision take place ensuring timeliness.
- *Reliability*. Information is replicated into multiple distributed nodes to maintain a high level of availability and correctness.

We have developed a system architecture that sustains the concepts of autonomous information services, called the Faded Information Field[10)~12]. Service availability and responsiveness are guaranteed by partially replicating the SP services at adjacent nodes in this architecture. The FIF is supported by a mobile agent infrastructure that stores pushed information services and executes pulled user requests, following two concepts:

- *Autonomous Information Provision*

Each SP pushes its accessibility toward users to promote its contents,

- *Autonomous Information Utilization*

Each user pulls requested data corresponding to his or her requirements.

3.1 Faded Information Field Architecture

The FIF architecture is here explained at a high level by describing its components (see **Fig. 1**). In the FIF model, each actor is given a mobile agent that bridges the autonomous entities. SPs disseminate their information services to *nodes* (replicated information holders) towards *users* relying on a demand-oriented communication protocol. Nodes consist of a stationary *node agent* that provides an execution and routing environment for mobile agents (see **Fig. 2**). *Pull-MAs* embody users' requests to drive them to the correct service, while *Push-MAs* disseminate SPs' information.

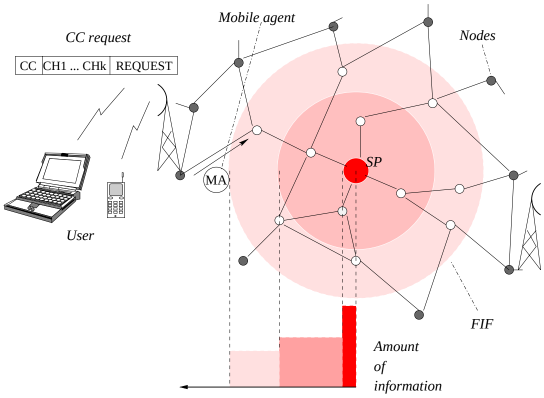

Fig. 1 FIF network structure.

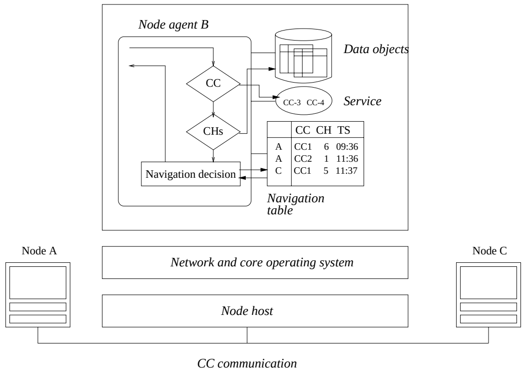

Fig. 2 FIF computation and communication environment.

Information fading is achieved by recursively removing the less popular information and allocating the remainder to storage nodes. Users with different requirements for information attain satisfaction of service at different levels of the FIF. Figure 1 shows the FIF architecture and the repartition of service information on the nodes.

3.2 Content-Code Communication

Conventional communication methods use the destination address to send the data. In a rapidly changing environment, the state of the nodes, the stability of the connections, and the status of the SP are unpredictable, and therefore users cannot specify an SP directly.

In the FIF communication method, the sender does not specify the destination address but only multicasts the data with its *Content Code* (CC), which is uniquely defined with respect to the content of the information service. The node agent selectively receives the requests according to the value of their CC and processes

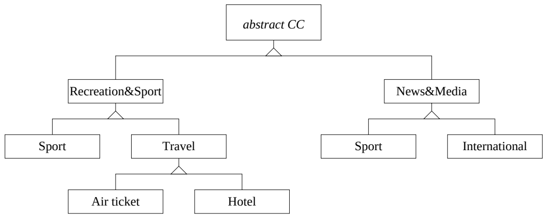

Fig. 3 Content code tree structure.

them autonomously.

A user specifies a request by affixing *characteristic codes* (CH), corresponding to the information properties of the CC. The node agent checks the CC with the entries of the *navigation table* in the local service repository. If the CC is available, it verifies whether relevant CHs are stored in the data object repository. Unsatisfied requests for unmatched CCs or unavailable CHs are further routed to an upper level of information or to another FIF. The user request is then gradually driven to the relevant node agent to process its data (see Fig. 2).

The CCs are organized as a tree structure of inheritance. **Figure 3** shows a part of the CC tree structure. All CCs inherit from the class *Abstract CC*. CHs represent instance attributes of a particular implementation of *abstract CC*. The inheritance is based on the semantics of each CC, so many CCs may inherit from different classes, like the class *Sport* in Fig. 3.

3.3 Push/Pull Mobile Agents

Mobile agent technology[13),14)] is utilized in the FIF as the medium for a dynamic interaction between service provision and service utilization. A reliable mobile agent architecture[15),16)] is available on each node of the network, and a persistent *Node Agent* negotiates with external agents.

- **Push-MA**

Push-MAs are issued by SPs to replicate pruned information toward centers of service consumption. To route the allocation process, they consider the information access effort. Negotiations about the amount of distributed information take place at the node site with the *Node Agent*. When the node accepts the allocation of the Push-MA data, the agent stores CC, CHs, and dated information. Push-MAs thus create a gradient of distributed information that Pull-MAs dynamically follow to get to the relevant Web site replica.

- **Pull-MA**

Pull-MAs are originated by users to embody the CC-driven request. When arriving at a node agent, the Pull-MA checks the available services. If the relevant CC is found, the Pull-MA then checks the data properties available for this CC. If CC and CHs are relevant, the Pull-MA sends back the data to the user and kills itself. If nothing corresponding to its own request is found, the Pull-MA takes a navigation decision checking the content of the *navigation table*. The selection of the node for service consumption is oriented toward satisfaction of the users' multiple criteria for assurance.

A trade-off must be found between the push and pull efforts in terms of access time and quality requirements. The allocation mechanism is driven by comparison of the Pull-MA access time and the information time-to-live. On the other hand, the navigation mechanism results from comparison of the user's quality requirements and the data properties at the node.

Consequently, users, nodes, and SPs are loosely connected through the sending and receiving CC requests, realizing flexibility. Timeliness is achieved through the integration of the search environment and the information service system, and reliability is the result of the service replication. The FIF architecture realizes thus an autonomous information service system.

4. Integration Synergy and Correlation

Synergy between two SPs exists if the users accessing their services share the same profiles. Following the three views of content, property, and category, *correlation ratios* are defined to measure the synergy between two services.

- **CC Correlation**

The CC correlation represents the proximity of two services from a business viewpoint. Looking back at Fig. 3, the CC correlation ratio is obviously high between CCs inheriting from the same superclass, but it can also be transversal to the inheritance tree. For instance, the content code "*Sport*" derived from "*News&Media*" is closely related to the CC of the same name but inherited from "*Recreation&Sport*." We consider two types of correlation: semantic correlation and functional correlation. The case of semantic correlation is beyond our research interest, and a great deal of work has already been done on ontology and knowledge management. In the case of the functional correlation, the ratio of correlation is inversely proportional to the number of inheritance connectors separating two CCs in the hierarchical tree.

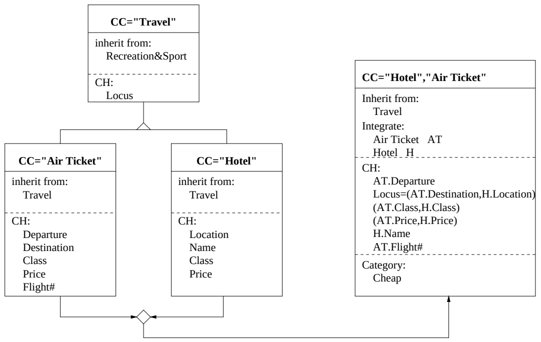

Fig. 4 Service integration.

We call $\mathcal{F}(CC1, CC2)$ the correlation function measuring the functional and semantic proximity of CC1 and CC2. If $\mathcal{F}(CC1, CC2)$ is beyond the integration threshold, their appended CCs, $(CC1, CC2)$, define the integrated service CC satisfying

$$\forall i \in \{1, 2\}, CC_i \subset (CC_1, CC_2)$$

This relationship is denoted by a diamond symbol in **Fig. 4**.

- **CH Correlation**

For services sharing the same business area, the heterogeneous characteristic codes are compared for complementary integration. The CH correlation is linked to the number of common CHs that two different services have. The CH "*Destination*" in the case of On-line Air Ticket Sales and the CH "*Location*" in the case of On-line Hotel Reservation both inherit from "*Locus*" in the superclass "Travel," and must be used as a service comparison factor (see Fig. 4). Other CHs sharing the same type, such as "*Price*," can also be semantically integrated, but no direct comparison can be undertaken in that case. Those semantically related CHs can however be the basis for partitions into categories.

- **Category Correlation**

For service offers having the same business positioning and sharing correlated CHs, a category partition is achieved by horizontally splitting the service offer. On the basis of the values of a correlated CH, a partition of the SP service offer is created.

Let us assume that we have two service providers: SP-1, an on-line air ticket sales site, and SP-2, as on-line hotel reservation site. Let us assume that both offer information about "prices." Splitting the service offer into "Cheap," "Averagely priced," and "Expensive" results in a price-based partition of their service offers. As far as correlation is concerned, the integration is most fruitful when the categorization reveals a common business positioning for both SPs. In the case where the two SPs specialize in bargains in their areas, their common price-based category is "Cheap."

4.1 Synergy Analysis

Consequently, the integration of two SPs is relevant when a common ground for business, CH, and category can be found. Coming back to the travel agent example, their possible integration must concern "cheap travel" with characteristic codes about "flight" and "hotel" (see Fig. 4). Using an object-oriented notation, the integration of two services results in the creation of a new class of service that encapsulates the correlated services.

Let A_0, A_1, $A_2, \ldots, A_N$ denote all the nodes between two Web sites, where A_0 and A_N are distinguished as the service providers, respectively SP1 and SP2. Let $N_i(A_j)$ denote the amount of information available for service CC_i served at node A_j and $N_{\mathcal{I},i}(A_j)$ the amount of CC_i information for the category $\mathcal{I}$ at node A_j. Furthermore, let Ω denote the subset of common CHs in the integration between SP_1 and SP_2. The popularity of each characteristic code CH_j is measured by its *access effort*, $P_{i,j}(A_k)$. The ratios can be expressed by geometrical means to reflect the integration tradeoff of the heterogeneous SPs. Three ratios are defined to reflect the combinational synergy at node A_k:

$$\mathcal{C}_1 = \mathcal{F}_k(CC_1, CC_2) \quad (1)$$

$$\mathcal{C}_2 = \sqrt{\sum_{j \in \Omega} P_{1,j}(A_k) \sum_{j \in \Omega} P_{2,j}(A_k)} \quad (2)$$

$$\mathcal{C}_3 = \sqrt{\frac{N_{\mathcal{I},2}}{N_2(A_N)} \frac{N_{\mathcal{I},1}}{N_1(A_0)}} \quad (3)$$

$\mathcal{C}_1$ represents the proximity of the two SPs from a business point of view. The function $\mathcal{F}_k$ may contain several parameters and consequently differ from SP to SP and from node to node, so that nothing ensures its symmetry:

$$\mathcal{F}_k(CC_1, CC_2) \neq \mathcal{F}_k(CC_2, CC_1).$$

$\mathcal{C}_2$ reflects the importance of the shared CHs. If the integration of service is achieved upon unpopular service properties, there is little probability that complex requests will contain them as parameters of a joint selection. Therefore, the more attractive the common CHs are, the more fruitful the integration promises to be.

$\mathcal{C}_2$ is also symmetrical, which means it has the same value for SP_1 and SP_2.

$\mathcal{C}_3$ expresses the relative size of the category $\mathcal{I}$ compared to the overall service offer. An insignificant size of $\mathcal{I}$ obviously tarnishes the synergy. This ratio is also symmetrical as both SPs benefit from each other's category scope.

Each SP and each node have their own integration requirements. The $(\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3)$ vector is analyzed case by case, following a given integration threshold. A negotiation process takes place at a node agent involving the three ratios and the number of accesses generated by each service. Once an integration decision is taken, the merging of the two services not only benefits from dual provision, but also attracts requests composed of joint selections, usually requiring synchronization of its atomic actions.

5. Autonomous Integration

The autonomous integration protocol is responsible for binding the information service allocation of two SPs to satisfy a majority of joint requests and reducing the access time of Pull-MAs. Once the integration of services 1 and 2 is concluded, Push-MAs are in charge of balancing the information repartition, taking into account this new potential increase in service consumption. To achieve this, we use an analysis of the request trends for the combined services.

5.1 Integration Requirement

Considering Pull-MAs complex requests for services 1 and 2, we partition them between those having as a primary request service 1 and those having as a primary request service 2, ordering them according to the number of CHs requested for both services. Let $C_1(A_i)$ denote the number of Pull-MAs consuming information service 1 at node A_i and carrying a joint request for services 1 and 2. Those Pull-MAs have various needs in terms of SP2 information. An average pattern of those needs is established by compiling Pull-MAs requests for service 2. Let $N_1^2(A_i)$ denote the average amount of information 2 requested at node A_i of *FIF1*. Let $N_1^1(A_i)$ be the amount of information 1 contained at node A_i of *FIF1* as a result of the fading process. $N_1^1(A_i)$ is also the amount of information 1 that satisfies $C_1(A_i)$ Pull-MAs with joint requests.

Let us consider a Push-MA MA_1 emitted by SP1 following the path $\{SP1, A_1, A_2, A_3, \ldots, A_{N-1}, SP2\}$. The amounts of information 1 and 2 satisfying a majority of Pull-MAs having as primary request service 1 are therefore:

$$I_1^1 = \frac{\sum_{i=0}^{N} C_1 N_1^1(A_i)}{\sum_{i=0}^{N} C_1(A_i)} \tag{4}$$

$$I_1^2 = \frac{\sum_{i=0}{}^{N} C_1 N_1^2(A_i)}{\sum_{i=0}^{N} C_1(A_i)} \tag{5}$$

Equations 4 and 5 define the integration requirement of FIF1, and can be expressed as follows: "*FIF1 requires at least I_1^2 of information 2 where I_1^1 of information 1 is available.*"

5.2 Steps of the Protocol

Autonomy is actually the most important requirement in a rapidly changing environment. We propose a protocol that meets this requirement by adapting the information level of each FIF according to its counterpart needs. The protocol is divided into three parts: handshake, monitoring, and allocation.

Handshake

(1) Eventually, Push-MA MA_1 encounters another service on its route. The correlation ratios $(\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3)$ are evaluated and compared with SP1's requirements. If the ratios satisfy SP1's requirements, MA_1 launches an integration request.

(2) Eventually, a Push-MA MA_2 from SP2 reaches the node where an integration request has been launched. MA_2 undergoes the same process as MA_1 and, if the ratios satisfy SP2's requirements, the integration process is started.

(3) The node is tagged with a new content code (CC1, CC2).

(4) Gradually, each node in the FIFs starts the integration process for SP1 and SP2.

Monitoring

(1) For each CC_i, the process monitoring N_i^1, N_i^2 and C_i is activated at each node in the new FIF (CC1, CC2).

(2) The functionality for analyzing integration requirements at each node is attached to the emitted Push-MAs of SP1 and SP2.

Allocation of Integrated Information

(1) At node A_i, MA_1 collects the service consumption logs, $C_1(A_i)$, $N_1^1(A_i)$, and $N_1^2(A_i)$ of *FIF1*.

(2) MA_1 computes the local integration requirement $I_1^1(A_i), I_1^2(A_i)$ based on the logs it collected on its path.

(3) Eventually, its counterpart, MA_2 reaches

A_i. MA_2 brings the peer requirements, $I_2^1(A_i), I_2^2(A_i)$ based on its own path until A_i.

(4) MA_1 compares $I_1^1(A_i)$ and $I_2^1(A_i)$. The minimum of the I^1's is kept as the amount to be allocated. If the current amount of allocated information $N_1^1(A_i)$ is greater than this minimum, the distribution is left unchanged. Otherwise, more information 1 is allocated to A_i.

(5) MA_2 operates similarly with I^2 of FIF1 and FIF2.

(6) MA_1 continues its path to A_{i+1} and continues the allocation process.

(7) When MA_1 reaches SP2, the overall integration requirements of FIF1, I_1^1 and I_2^1, are transmitted. When SP2 emits a next Push-MA, it will integrate those requirements with the Push-MA to alleviate the effects of intermediate results.

(8) Recursively, the integration process continues.

The process continues autonomously. The level of information in the integration area is recursively adapted to complex request needs. The size of the overlapping area is modified in the limit of the individual SP requirements for consistency, depending on the increase in service consumption.

5.3 Simulation

We tried our protocol on a grid network bestowing 6-ary connectivity to each node and containing two SPs: an Air Ticket site and a Hotel Reservation site, as previously described. The two SPs are separated by a minimum of 6 hops, and we suppose each FIF to be composed of 4 levels of information. The user distribution on the network is uniform. We consider only joint-selection requests such as "*Find a Hotel and two air return tickets to Paris for one week in May at the best price*."

For comparison, a client-server environment with a hop-by-hop routing protocol was adopted. Each server centralizes its services and mobile agents are dynamically guided towards the relevant service going from node to node.

We studied the two systems in rapidly evolving environments where faults, disruption, and congestion occur frequently. We concentrated mainly on two items: the dynamic behavior of the integration protocol and the scalability of the integration process with the number of nodes in the network and the number of requests per second in the system.

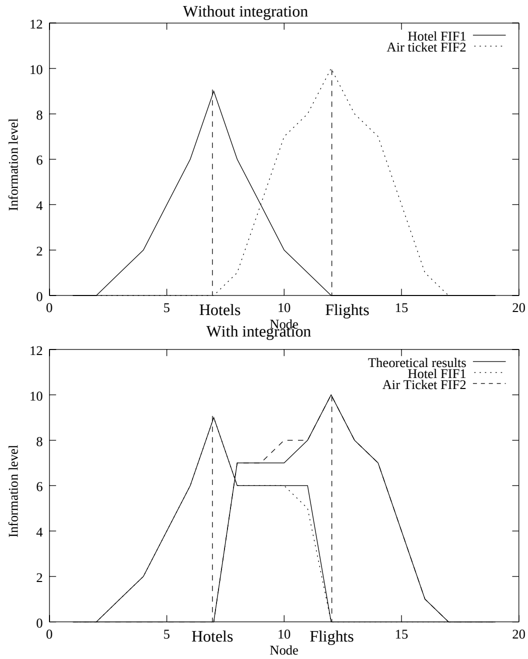

Fig. 5 Result of simulated integration.

5.3.1 Integration Protocol

Figure 5 shows the information distribution results we obtained after execution of the protocol. The first graph represents the information distribution before executing the protocol. We modeled a request repartition that led to the values $I_1^1 = 8, I_1^2 = 7, I_2^1 = 5, I_2^2 = 7$, according to equations 4 and 5. Theoretical computations following the exact overall distribution of the requests led to the information distribution represented by a solid line in Fig. 5.

5.3.2 Network Scalability

We measured the scalability of the integration according to the number of nodes in the network. The results revealed a greater scalability of integrated services. The search time reduction for Pull-MA is more than 40% (see **Fig. 6**). The reasons for such an observation are twofold. First, the integration has for effect to increase the size of the common area where both services are provided. Pull-MAs find the corresponding service more easily. Second, the information-allocating mechanism leads to the satisfaction of a greater number of complex requests. A new class of service is created that satisfies new types of users.

5.3.3 Scalability of the Number of Requests

We modeled the scalability of the system ac-

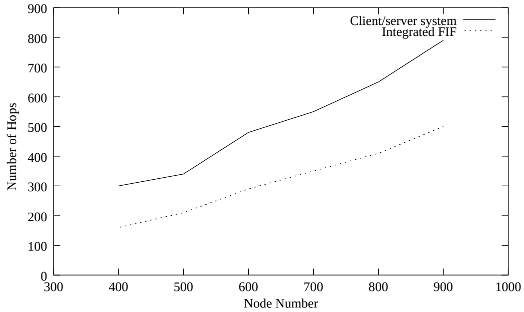

Fig. 6 Scalability with number of nodes.

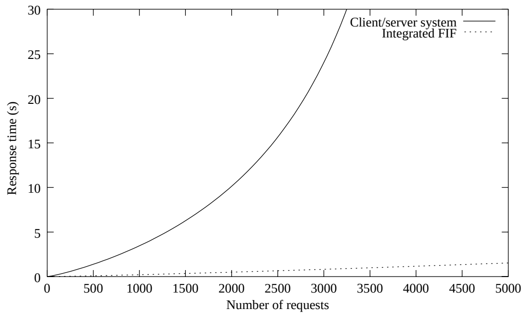

Fig. 7 Scalability of the response time with the number of requests.

cording to the number of requests arriving at the nodes. The experiment was conducted for nodes displaying infinite latency when the number of requests per second exceeds 5,000. We then computed the average response time for numbers of requests per second between 0 and 5,000, neglecting the propagation time. The test showed a response time reduction of more than 90% (see **Fig. 7**). The introduction of service redundancy and the dynamic binding of the user request to an appropriate site replica are the underlying reason of this result.

Globally, the integration of services was shown to facilitate the searching process in rapidly evolving environments. Pull-MAs navigate in integrated FIF structures as if only one CC were present, going directly to the levels satisfying both its requirements.

6. Conclusion

Recently, the necessity of high-assurance in information service systems has been stigmatized as a result of the problems of saturation and unavailability. Providing reliability, availability, and timeliness in such systems is of major importance. For this reason, an architecture of pruned information replicas, called *FIF*, has been designed to satisfy the heterogeneous requirements of SPs and users in rapidly changing environments. In this architecture, the most popular information services are autonomously disseminated towards centers of service consumption to reduce user access time and increase SP reliability. For mono-service requests, the FIF succeeds in greatly reducing user access time while maintaining data integrity. However, service provision has to be adapted to satisfy joint multi-services requests. It is necessary that services that are usually requested jointly should offer a common interface for service provision. in this article, we have proposed to autonomously-integrated information services that share a common replication area in an FIF environment.

By this means, SPs that are ordinarily confined are offered tremendous opportunities for expansion. They can increase their service consumption by attracting new types of users with complex multi-services demands. They can also push towards a homogeneous group of users information selected according to their profile. Finally, they can add significant value-added services to their original offer to attract more users. By autonomously adjusting their service offers to fit with one other, an area of synergistic provision of multi-services is dynamically created on the basis of the three views of content, property, and category.

Experiments demonstrate the potential savings from autonomous integration of correlated SPs. The efficiency of the autonomous integration protocol has also been confirmed by simulation. The user access time is improved by up to 40%, and congestion due to synchronization communication is avoided.

References

1) Hunt, R.: Evolving Technologies for New Internet Applications, *IEEE Internet Computing Magazine*, Vol.3, No.5, pp.16–26 (1999).

2) Ingham, D.B. and Shrivastava, S.K.: Constructing Dependable Web Services, *IEEE Internet Computing Magazine*, Vol.4, pp.25–33 (2000).

3) Yen, I.-L., et al.: Toward Integrated Methods for High-Assurance Systems, *IEEE Computer Magazine*, Vol.31, No.4, pp.32–34 (1998).

4) Bestavros, A.: Demand-based Document Dissemination to Reduce Traffic and Balance Load in Distributed Information Systems, *Proc. SPDP*, IEEE, pp.338–345 (1995).

5) Bestavros, A.: WWW Traffic Reduction and Load Balancing through Server-based Caching, *IEEE Concurrency Magazine*, Vol.5, pp.56–67

(1997).
6) Gwertzman, J. and Seltzer, M.: The Case for Geographical Push-Caching, *Proc. HotOS-V*, IEEE, pp.51–55 (1995).
7) Baker, A.D., et al.: Agents and The Internet: Infrastructure for Mass Customization, *IEEE Internet Computing Magazine*, Vol.3, pp.62–69 (1999).
8) Mori, K., et al.: Service Accelerator (SEA) System for Supplying Demand-Oriented Information Services, *Proc. ISADS*, IEEE, pp.129–136 (1997).
9) Mori, K.: Autonomous Decentralized Systems: Concepts, Data field Architecture and Future Trends, *Proc. ISADS*, IEEE, pp.28–34 (1993).
10) Ahmad, H., Arfaoui, H. and Mori, K.: Autonomous Information Fading by Mobile Agents for Improving User's Access Time and Fault-Tolerance, *Proc. FTDCS*, IEEE, pp.279–283 (1999).
11) Mori, K.: Autonomous Fading and Navigation for Information Allocation and Search under Evolving Service System, *Proc. APSITT*, IEEE, pp.326–330 (1999).
12) Arfaoui, H., Ahmad, H. and Mori, K.: Flexible and Reliable Faded Information Field through Neighbor Discovery, *Proc. IWDCCA*, IEEE (2000).
13) Pham, V. and Karmouch, A.: Mobile Software Agents: An overview, *IEEE Communication Magazine*, Vol.31, pp.26–36 (1998).
14) Fugetta, A., et al.: Understanding Code Mobility, *IEEE Trans. Softw. Eng.*, Vol.24, pp.342–361 (1998).
15) Dalmeijer, M., et al.: A Reliable Mobile Agents Architecture, *Proc. OORTDCS*, IEEE, pp.64–72 (1998).
16) Silva, F.A., et al.: An Approach for Providing Mobile Agent Fault Tolerance, *Proc. 2nd International Workshop on Mobile Agents*, Springer-Verlag, Vol.1477, pp.14–25 (1998).

(Received May 8, 2000)
(Accepted November 14, 2001)

Hélène Arfaoui received M.S. degree in telecommunications technologies from both ENST Bretagne in France and Aston University in England in 1997. In 1998, she joined the Tokyo Institute of Technology, Japan, as a research student and joined the Ph.D. program in 1999 under the supervision of professor Kinji Mori. Her research interests are distributed and high-assurance systems, information services, and autonomy of mobile agents. She is a member of IEEE society.

Kinji Mori received the B.S., M.S., and Ph.D. degrees in the Electrical Engineering from Waseda University, Japan in 1969, 1971, and 1974, respectively. From 1974 to 1997 he was with Systems Development Lab., Hitachi, Ltd. In 1997 he joined Tokyo Institute of Technology, Tokyo, Japan as a professor. His research interests include the distributed computing, the fault tolerant computing, the assurance and the mobile agent. He proposed Autonomous Decentralized Systems (ADS) in 1977 and since then he has been involved in the research and development of ADS. He is one of founders of the International Symposium on ADS (ISADS), was the general chair of the fourth ISADS, and now is the steering commitee chair of ISADS. He served many international academic activities as the general chair, the program committee chair and the member of conferences, the editorial board of IEEE Transaction on Computing, and as the IEEE Technical Board Committee of Computer Communication. He received the Best paper award of IEEE COMPCON fall 1984, the Science and Technology Minister's Patent Award, Japan Minister of Science and Technology, 1994, the Special Distinguished Ichimura Award conferred by the Honorable Prince Mikasanomiya, 1994 and so on. He is a Fellow of IEEE, and member of IPSJ and SICE, Japan.