Regular Paper

Autonomous Reconfiguration Technology for High Response Time in Decentralized Database Systems

Carlos Perez Leguizamo,† Minoru Takaishi,† Sotaro Kimura† and Kinji Mori†

With the exponential growth of the World Wide Web in recent years and the progress of information technology, the business market is changing its strategy for a modern online business environment. Autonomous Decentralized Database System(ADDS), based on autonomous coordinating subsystems, has been proposed as a system architecture in order to meet the innovative e-business requirements. Autonomy and decentralization of subsystems help achieving high response time in highly competitive situation and an autonomous Mobile Agent based coordination has been proposed to achieve flexibility in a highly dynamic environment. Further, this paper proposes the autonomous reconfiguration technology in ADDS in order to adapt the system to changing heterogeneous requirements. Based on the complementary requirements of the subsystems, heterogeneity distribution has been defined to accomplish the reconfiguration process. This autonomous reconfiguration technology consists of autonomous division and integration of the system, based on the ever-changing situation, to assure high response time. Simulation of the proposed technology shows its effectiveness for a large scale system and rapidly evolving situations.

1. Introduction

In recent years, the advent of information technology has affected the consumer and business world markedly. With the wide spread use of the Internet, while on the one hand, a service provider is able to offer global services, a customer on the other hand, is powered by global access ability. As a result, the customers are ought to be more demanding and the conditions of service provision are becoming more and more stringent. Moreover, due to the wide choice range, the users' trends have become highly dynamic and unpredictable and the total system is becoming chaotic. Also, under this situation, even a temporary failure of service provider leads to business failures. In short, the companies are unable to predict the users' demands and are threatened of business failures due to the intense competition around the globe. In this background, the need for cooperation among multiple companies and the strategic alliances -such as the Supply Chain Management System- is ever increasing [1].

The environment and the market trends being so diversified and dynamic, new system architectures are required by the heterogeneous e-business entities for mutual cooperation to enhance individual capabilities in regard to diversified customer's trends. Considering the dynamic environment, intense competition and possible business losses, the system must provide high response time, fault-tolerance and flexibility.

A new system architecture, Autonomous Decentralized Database System (ADDS), has been proposed in order to satisfy the enhanced requirements of current online e-business applications. This system architecture is inspired by Autonomous Decentralized Systems [2),3)], where a number of autonomous subsystems are loosely connected through a Data Field (DF). This system has been shown to have properties of real time, fault-tolerance and online expansion in high-assurance systems [4),5)]. Similarly, the Autonomous Decentralized Database System [6)] is based on the concept that each site in the integrated system is completely autonomous. The attribute Allowable Volume (AV) has been defined to provide decentralization and thus to achieve real time updates on each site [7)]. A Mobile Agent (MA), an autonomous entity, is devised for coordination by the continuous adjustment of AV among the sites [8)]. The completely autonomous architecture provides the flexibility in the system and the decentralization offers high availability.The coordination among the sites,however, requires finite time, and as the number of sites in the system increases, the average response time of the system also tends to increase. For such a situation, an autonomous reconstruction technology has been

† Tokyo Institute of Technology

Table 1 Application and system requirements.

Application needs	System needs
Heterogeneous needs	Heterogeneity
Dynamic environment and preferences	Flexibility
One-click response	Timeliness
Continuous Service	Fault Tolerant

proposed for achieving high response time. This reconstruction involves the heterogeneity distribution of complementary requirements of the sites, in order to improve the average system response time.

2. Requirements

2.1 Application Requirements

As stated earlier, the environment in which a consumer accesses a service provider is continuously changing. The service providers offer a variety of services, and customer is free to access the service from any where in the world. This situation introduces high competition as customer wants one-click response to his request. Similarly even a temporary failure in service provision may distract the customer to other service providers. Moreover, considering the dynamic market environment, the customers' demand at the retailer sites is changing in time and space. The service providers always need to adapt to this change in customer trend. On some sites, due to the increase in the customers demand, increased amount of products may be required to decrease the risk of opportunity loss, i.e., the loss of business by missing the business opportunity. On the other hand, some sites may need to dispose off its extra stock or inventory in order to reduce the stock cost risk due to the over inventory or the excessive stock at the site compared to the customers demands. So there exist heterogeneous needs, not only among the makers and retailers but also among multiple retailers itself, changing with time.

2.2 System Requirements

The system needs for the integration of DB systems which can cope with the application needs mentioned above are summarized in **Table 1**. For a distributed database system in an environment like the Internet, in order to satisfy the heterogeneous needs, such as the reduction of the opportunity-loss and the stock-loss cost; and to realize the adaptability of the system, timeliness, fault tolerance, and flexibility are needed.

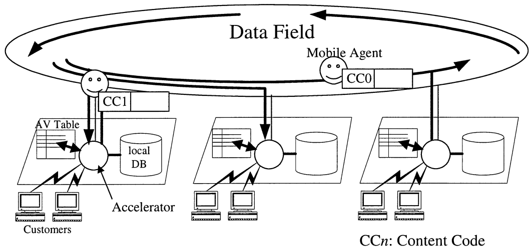

Fig. 1 ADDS architecture.

3. Autonomous Decentralized Database System

Considering the application and system requirements in a dynamic and unpredictable environment Autonomous Decentralized Database System (ADDS) has been proposed in order to satisfy the requirements and to cope with highly dynamic environment [6]. In ADDS the operational database systems of autonomous business sites with heterogeneous needs can be integrated into a common environment without violating each other's characteristics as compared to conventional approaches [9),10)]. The system architecture of ADDS is shown in **Fig. 1**.

Sites in the system are completely autonomous in local update process without communicating with any master site, hence providing the real time capability to the system. The Mobile Agent (MA) is responsible for autonomous coordination among the sites. The interaction of autonomous sites and autonomous MA forms a Data Field where the sites and the mobile agent can communicate autonomously, hence allowing sites to negotiate their needs flexibly [11),12)]. The detailed mechanism of the ADDS architecture is given in the following.

3.1 Loosely-Consistency Management Technology

In ADDS, each site holds Allowable Volume (AV) which defines the maximum permissible volume with in which each site can have local updates autonomously. This AV is defined on all numeric data and represents a flexible amount that can be managed by negotiation among the autonomous cooperating entities. Considering a typical example example of online Supply Chain System, where autonomous sites of makers and retailers are integrated, if there is some request of a product (defined by AV) on one site and this requested amount is less than the AV on this site, the sites can immediately update this request without communicating with any other site and the consistency of the distributed database is also preserved [7]. Hence the real time in updates is achieved. On the other hand, if the requested amount is more than the local AV available at his site, the site is required to coordinate and arrange the lacking amount of AV from other sites, described in the following.

3.2 Background Coordination Technology

A background AV coordination, performed by an autonomous MA, is devised to adapt the system to evolving situations by allocating AV in advance among the sites that has different time constraints for updating. The consumers' needs at each site change dynamically and independently, and therefore the amount of AV at each site is required to be optimized. In this process some sites may run into frequent shortage of volume and some other sites might have surplus volume. Thus, an adjustment of AV among the sites is required. The performance of the whole system greatly depends on the strategy utilized by the MA for the allocation of AV. For this paper we utilize the supply-based mechanism which has been proposed for a timeliness coordination. In this strategy the amount of AV that the MA can coordinate is freely determined by the current AV demand/offer from the sites [8].

3.2.1 Autonomous MA Circulation Mechanism

In ADDS, the MA moves among all the sites in order to maintain the consistency of the system, thus forming a logical dual-ring network. Each site in the system is furnished with a Site Information Table (SIT), that records the information of the neighboring sites directly connected to it (in front and behind).

When the MA visits a site, it can get the address of the next site from the SIT. For example, once the MA finishes the adjustment of AV in site 3, it can know according to the SIT that the next site it must visit is site 0. Thus, the MA gets the address of the site 0 and moves to it. In the case of a new site integrates to the system, such as site 4, this site has to start an initializing process by sending an *initial MA* for searching a site that belongs to the system. When the *initial MA* reaches such site, such as site 3, this MA keeps the data contained in the SIT of the site and adds the information of its *host site*. Then *initial MA* goes back to the original site and rewrites the SIT with the information that has carried from the *founded site*. Finally, it finishes the reconstruction process of the dual-ring network by modifying the SIT of the site that was linked as *next site*, site 0, to the *founded site*.

3.2.2 Autonomous Fault Detection and Recovery

Due to the inherent decentralized nature of the architecture, the ADDS provides high availability and fault-tolerance. In case of the failure of some sites in the system, the rest of the system keeps on working normally. Even in case of failure of MA the sites still can work normally with their allocated amount of AV. But under a highly changing environment, the demand of AV at different sites may change rapidly and an efficient coordination among the sites is inevitable. Under such a situation, an immediate recovery of the failing MA and the corresponding AV has been presented in this section. In this technology, each site in the system detects the failure of the MA if it is not able to receive it in a specific time. The site that detects the MA's failure generates a new MA who will start moving around the DF instead of the MA that had a failure. The new MA can recover the AV by visiting each site and grasping the site's stock C_i [of site i], the AV value(AV_i) and the increase or decrease of the AV value ΔAV_i during MA failure duration. Once the new MA completes one cycle, the lost AV carried by the failed MA can be recovered using the following expression.

$$AV_{MA} = \sum_{i=0}^{n}(C_i + \triangle AV_i - AV_i) \quad [nsite]$$

In this scheme the multiple MA may exist momentarily due to multiple fault detection at autonomous sites, only the MA generated by the site with highest ID in the system will exist finally as the site having an ID higher than the MA will suppress that MA and only the MA with highest ID will exist in the system, hence the consistency of the updates is achieved.

The MA timeout at each site is set by the mobile agents's circulation time based on MA's actual visits on each site. Under this condition, on the initialization stage of the system no site has knowledge of number of sites and the MA circulation time. If during its first cycle, the MA fails then no site can at all detect the MA failure time. For this reason, a minimum initial timeout, in the initialization stage is set by each site.

4. System Size and the Response Time

This paper examines the critical relation between the user's response time and the number of sites in a typical integrated system and proposes a dynamic solution for the reconstruction of the system.

If Ru represents the user demand at any particular instant on a particular site in the system, AV_i is the amount of product volume on this site, and AV_{all} is the total AV in the system then on the arrival of a user request on site i there exist two cases:

i. $AV_i \geq Ru$
In this case site i has enough AV to satisfy the user's demand immediately. Therefore the response time to the user T_w is just the local processing time T_{CAV} on this site, i.e.,

$$Tw_1 = T_{CAV}$$

ii. $AV_i < Ru$
When the site does not have the enough AV to satisfy the user's demand the site collects the lacking amount of AV from the mobile agent carrying extra AV from other sites in the system. If T_{MA} is the time to complete the MA one cycle among the sites, the average time for the MA to reach the demanding site can safely be stated as $T_{MA}/2$. If the probability that MA is already carrying the required amount of lacking MA when it arrives the demanding site is p_1 then the user's response time in this case Tw_2 can be written as follows

$$Tw_2 = T_{CAV} + \frac{T_{MA}}{2} + \sum_{k=0}^{\infty}(1-p_1)^k \cdot p_1 \cdot kT_{MA}$$

Moreover, if the probability that $AV_{MA} \geq Ru - AV_i$ is represented as p_2 then the user's average response time Tw is given by

$$Tw = T_{CAV} + (1-p_2)\left\{\frac{1}{2} + \sum_{k=0}^{\infty}(1-p_1)^k \cdot p_1 \cdot k\right\}T_{MA} \qquad (1)$$

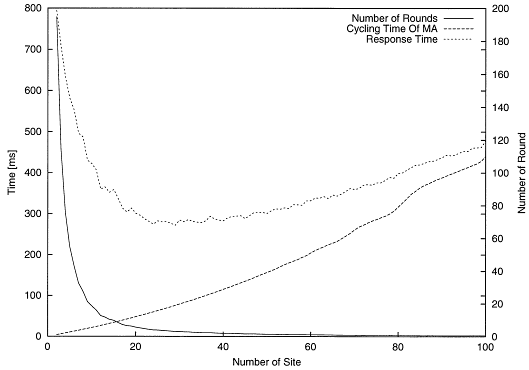

Fig. 2 Response time vs. number of sites.

Figure 2 shows the results of an experiment done under the simulation conditions and parameters described in Section 7.1. As it can be observed, the response time required to completely satisfy the user demands has a trade-off between the MA circulation time and the total number of MA circulations in the system before satisfying the user demand. We observe that the number of sites in the system has a marked effect on the average user's response time, i.e., it is quite susceptible to change in total number of sites and the diversity in their heterogeneous needs.

5. Autonomous Reconfiguration Technology

In this section, the autonomous reconstruction technology is proposed in order to realize the real-time property when the number of sites and diversity of sites' requirements change in the system.

5.1 Total amount of AV, Response Time and the System size

Considering a fixed system size, the number of circulations that the MA needs in order to complete the user request is considered, N_{MA} can be written as

$$N_{MA} = (1-p_2)\left\{\frac{1}{2} + \sum_{k=0}^{\infty}(1-p_1)^k \cdot p_1 \cdot k\right\} \qquad (2)$$

From this relation, we see that p_2, the probability of getting enough volume on the first site, depends on the total allowable volume AV_{all} in the system and increases with increase in AV_{all}. p_1, the probability of MA having enough volume to satisfy a certain site's demand at a particular instant, however is much more complex in nature and also depends on the number of sites with heterogeneous needs besides the AV_{all}. When the number of sites in the system is large, the MA has the opportunity to visit more sites with heterogeneous needs in a unit of time and hence the MA's probability of getting the AV increases. On the other hand, with a system having less number of sites, the MA will visit the same sites repeatedly during the same time interval and the probability to find sites with diverse needs decreases. Therefore, p_1 on the one hand is an increasing function of AV_{all} and a decreasing function of number of sites in the system. And N_{MA} is a decreasing function of both p_1 and p_2. Under this observation our technique proposes an autonomous division of the total system into multiple autonomous subsystems and integration of multiple systems into single system while optimizing the user's response time.

5.2 Division/Integration and the Heterogeneous Requirements

As it is clear by now, the sites in the ADDS take advantage of synergic effects of integration and diversity of different sites in the system help them coordinate each other in a complementary way. For example, some sites want to decrease the stock cost and want to give away the volume, the other sites on the other hand may want to decrease their opportunity loss arising from shortage of volume and want to gain some extra volume immediately. As mentioned in the previous section, the heterogeneity in site's requirements increases with increase in total number of sites. Therefore with a large integrated system the probability of satisfying each other requirements increases but at the same time due to the increase in the MA circulation time the average user's response time tends low. While on the other hand, the MA circulation time in a small system decreases, the heterogeneity of the sites' requirements also decreases, hence the user's response time does not effectively improved. As depicted in **Fig. 3**, an experiment done under the simulation conditions and parameters described in Sect. 7.1 with variable total AV, there exists a trade-off between the user's response time and the number of sites in the system. The system can then be divided or integrated based on this trade off. Considering AV_{all} constant, the number of MA circulations N_{MA} increases with the number of divisions of the total system. In case of

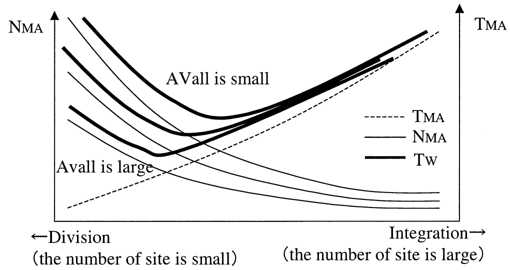

Fig. 3 Total AV and division/integration.

large and equally distributed AV_{MA}, the division of the system improves the user's response time. But if the AV_{all} is short and not equally distributed, though the integration of multiple systems into one system increases the T_{MA}, the average response time is improved. The total amount and distribution of AV among the sites (heterogeneity distribution) play an important role for the average response time. Division of system into multiple small systems is only effective when the heterogeneity distribution among the sites is uniform, otherwise the division does not necessarily improve the response time and rather integration of the system is required.

6. Autonomous Reconfiguration of Autonomous Decentralized Database Systems

In this section we propose the autonomous division/integration of the system based on the heterogeneity distribution evaluation. The following subsections explain the mechanism.

6.1 Heterogeneity Distribution Evaluation

Under given AV_{all}, the average response time depends on the size of the system. Thus division of system into multiple systems improves the response time in each system provided each reduced system becomes self sufficient in its allowable volume, ie., the sites have high heterogeneity in demands throughout the system. This heterogeneity of sites' demand being a detrimental factor in the system reconstruction process, we have defined Heterogeneity Distribution among the sites. Considering V_k as the allowable volume at site k and D_k the corresponding user's demand on this site, the heterogeneity distribution is given as follows:

$$h = \frac{1}{\alpha} \cdot \sum_{k=1}^{\alpha} \frac{(V_k - V_{k-1})^2}{V_k^2 + V_{k-1}^2} \qquad (3)$$

where α represents the total number of sites. Mobile agent, while moving among the sites,

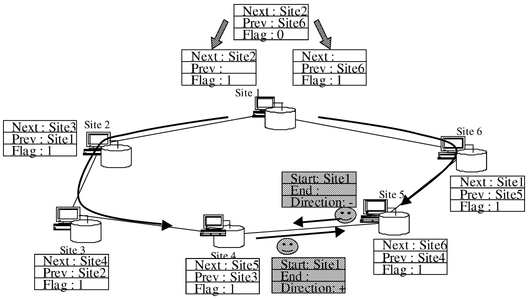

Fig. 4 Autonomous division technology (a).

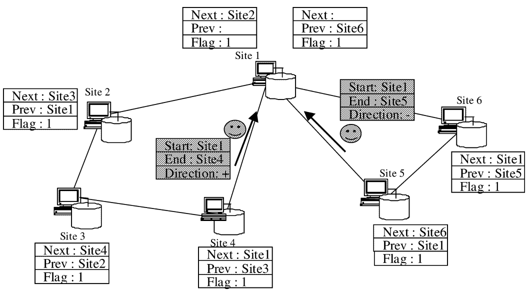

Fig. 5 Autonomous division technology (b).

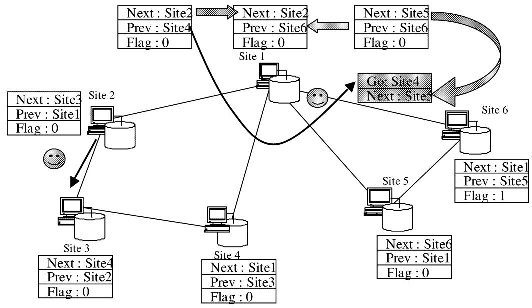

Fig. 6 Autonomous integration technology (a).

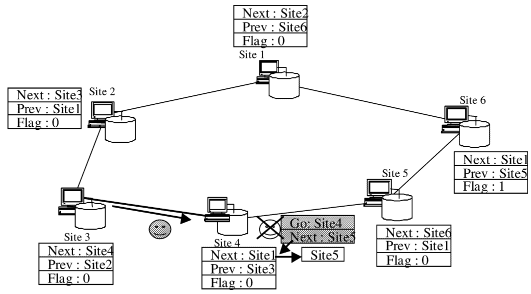

Fig. 7 Autonomous integration technology (b).

gathers this information from each site and determines whether to divide/integrate the system based on some preset threshold value of h, i.e., for a certain threshold values of d and d', the MA can decide to divide if $h \geq d$ and to integrate if $h < d'$. Hence when the heterogeneity among the sites is highly distributed the division of the system improves the response time and on the other hand, when the heterogeneity is only concentrated in certain regions of the system the division makes the response time even worse in some of reduced systems and rather integration of these systems is required. The two level thresholds (i.e., d and d') is used in order to provide a stability in the reconstruction process in a highly dynamic environment.

6.2 Autonomous Division of ADDS

The autonomous MA circulating among the sites reads the each site's demand versus allowable volume information, gathers this information from the sites during its regular visit on each site and evaluates the heterogeneity distribution given by Eq. (2). In case the heterogeneity distribution value is larger than the given threshold value decides to divide the system. As the system being dynamic and unknown the optimum system size can not be determined, the system is divided into half each time until each portion reaches to some optimum size.

The division process in ADDS is described in **Figs. 4** and **5**.

6.3 Autonomous Integration in ADDS

If the heterogeneity distribution in a ADDS becomes less than a certain threshold value, the response time in this system starts increasing irrespective of the smaller size of the system. In this situation, the system should be allowed to intermix with more sites having high heterogeneity thus increasing the chances of MAs satisfaction in a larger system. This can be achieved by integration of the existing multiple systems into one large system. In a large integrated system, though the MA circulation time increases, the average response time, however, is improved due to extended chances of MA satisfaction during one cycle. As in the division case, the integration of ADDS is explained in **Figs. 6** and **7**.

7. Evaluation

7.1 Simulation Model

Following our application example, we modelled a typical supply chain system, consisting of a maker and a dynamic number of retailers joining or leaving the system autonomously and total number of sites ranging from 4 to 30 for the simulation purpose. A random number of users were generated at each site, demanding a random amount of AV at different instants of time. The user arrival rate on each site is taken

Table 2 Simulation parameters.

Number of Nodes	4 – 30
Bandwidth	500 – 2,000 [Kbps]
User Timeout	3,000 [ms]
Request Arrival Rate Users	0 – 2,000 [ms]
AV Demand User Users	1 – 5
Number of Makers	1
Produce of AV	1,500
Produce AV Period	50 [ms]
Db Access Time	100 [ms]
MA Size	20 [KB]
MA Processing Time	100 [ms]
Total Initial AV	3,000
Simulation Time	300,000 [ms]
User Requests Dist. (skew)	0.75

randomly between 0 to 2,000 ms. The total AV in the system is supposed to be sufficient and always greater than the total users' demand. Each time, the total AV reduces to 10 percent of the total demands, the maker generates a new stock of the product and supplies it to the retailers. We study the case where servers receive transactions according to a highly-skewed Zipf distribution (0.75) that is the most common case in a very dynamic environment like the Internet. The system parameters used in this simulation are achieved by the implementation of the system on the Internet, using jdk 1.2.2 for the mobile agent and the agent platform implementation on each site. Also each site was having a PostgreSQL 7.1.3 database system. The following table (**Table 2**) gives the realistic measure of the parameters used in the simulation under the above stated implementation.

7.2 Results

The results shown in **Fig. 8** signify the advantage of autonomous reconstruction in ADDS based on the heterogeneity distribution of needs among the multiple sites. The results clearly show the improved response time as compared to an ADDS system without the technology, especially for large number of sites. And the improvement margin increases with increase in the number of sites. For small number of sites (less than 12 in figure), the autonomous reconstruction process has some overhead and therefore the average response time in this case has been shown low. This is due to the fact that the MAs, during the division/integration process, has to reroute them strayed from their normal route thus the response of a small number of forthcoming sites in their normal route is disturbed. But for a relatively large number of sites, this effect is negligible and as seen

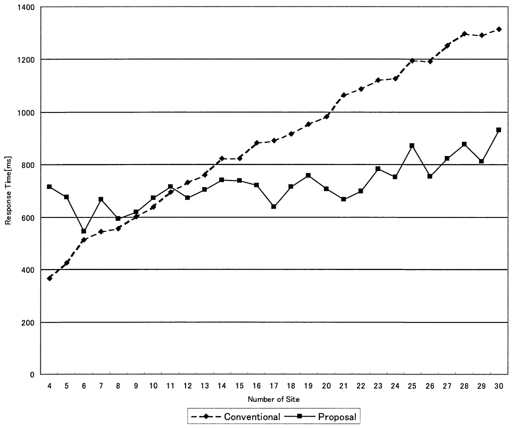

Fig. 8 Simulation result.

from the figure, outperforms the conventional system. In this sense, the proposed technology is only effective in a database system with the characteristics of ADDS in which the data is partitioned among the sites (AV) and there exists an intelligent entity, such as the MA, that while realizing the allocation and coordination of AV it can detect the heterogeneity in the distribution of the data and start the reconfiguration of the system.

8. Conclusion

Due to the wide spread use of information technology in business market, we have examined the very nature and trend shift in the modern online business environment. The system being of gigantic size, the environment is highly dynamic and unpredictable. Under this background, the conventional systems can not sustain in satisfying highly competitive and demanding requirements of service providers as well as customers. Autonomous Decentralized Database System has been presented as an innovative system architecture in order to meet the innovative e-business requirements. The system has characteristic feature of subsystem autonomy, most demanding feature in an unpredictable large scale system. Each database site in the integrated system is provided with autonomy for its local updates, thus providing a real time property. Further, autonomous mobile agent technology is used for address free coordination among all the database sites, this technique resolves the heterogeneous needs in the system and offers a flexibility in a highly dynamic environment. The autonomous reconstruction in ADDS is proposed to adapt the system to changing heterogeneous requirements. In a system where no single subsystem has total knowledge of the system, an autonomous way of division and integration is proposed to improve the average system response time. The technique has been shown effective while using the realistic parameters obtained from a real implementation of the system. Though the applicability of the system has been discussed with a typical e-business application — supply chain system, the architecture is applicable to variety of information service systems either on LAN or on the Internet.

References

1) Lee, H.L. and Whang, S.: E-business and supply chain integration, *Proc. SGSCMF'01* (Nov. 2001).
2) Mori, K.: A proposal of the autonomous decentralized system concept, *IEEJ Trans. EIS*, Vol.104-C, No.12, pp.303–310 (Dec. 1984).
3) Mori, K.: Autonomous decentralized systems concept: Data field architecture and future trends, *Proc. ISADS'93*, pp.28–34 (Mar. 1993).
4) Yen, I.-L., Paul, R. and Mori, K.: Toward integrated methods for high-assurance systems, *Computer*, Vol.31, No.4, pp.32–34 (Apr. 1998).
5) Yen, I.-L. and Paul, R.: Key applications for high-assurance systems, *Computer*, Vol.31, No.4 (Apr. 1998).
6) Leguizamo, C.P., Kato, S. and Mori, K.: Autonomous decentralized database system for assurance in heterogeneous e-business, *Proc. COMPSAC'01*, pp.589–595 (Oct. 2001).
7) Leguizamo, C.P., Dake, W. and Mori, K.: Loosely-concistency management technology in distributed database systems for assurance, *IEICE Trans. Inf & Syst.*, Vol.E86-D, No.10, pp.2104–2113 (Oct. 2003).
8) Leguizamo, C.P., Dake, W. and Mori, K.: Autonomous mobile-agent-based data allocation technology in distributed database systems for assurance, *IEICE Trans. Comm.*, Vol.E87-B, No.7, pp.1818–1825 (July 2004).
9) Perrizo, W., Rajkumar, J. and Ram, P.: HYDRO: A heterogeneous distributed database system, *Proc. SIGMOD'91*, pp.32–39, Denver, USA (May 1991).
10) Fu, A.W.-C. and Cheung, D.W.-L.: A Transaction replication scheme for a replicated database with node autonomy, *Proc. VLDB'94*, pp.214–225, Santiago, Chile (Sep. 1994).
11) White, J.E.: Mobile agents, Brandshaw, J. (Ed.), *Software Agents* (1996).
12) Harrison, C.G., Chess, D.M. and Kershenbaum, A.: Mobile agents: Are they a good idea?, Research Report of IBM (Mar. 1995).

(Received October 21, 2004)
(Accepted June 9, 2005)
(Online version of this article can be found in the IPSJ Digital Courier, Vol.1, pp.362–369.)

Carlos Perez Leguizamo joined the Ph.D. program in 2002 at the Tokyo Institute of Technology, Japan, under the supervision of professor Kinji Mori. His research interests are distributed and high-assurance systems, database systems and autonomy of mobile agents.

Minoru Takaishi is a master student under the supervision of professor Kinji Mori. His research interest are distributed and high-assurance systems.

Sotaro Kimura is a master student under the supervision of professor Kinji Mori. His research interest are distributed and high-assurance systems.

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 System Development Lab., Hitachi, Ltd. In 1997 he joined Tokyo Institute of Technology, Tokyo, Japan as a professor. He proposed Autonomous Decentralized Systems (ADS) in 1977 and since then he has been involved in the research and development of ADS.