

INVITED SURVEY PAPER

Trend of Autonomous Decentralized System Technologies and Their Application in IC Card Ticket System

Kinji MORI^{†a)}, *Fellow* and Akio SHIBASHI^{††b)}, *Member*

SUMMARY The advancement of technology is ensured by step-by-step innovation and its implementation into society. Autonomous Decentralized Systems (ADSs) have been growing since first proposed in 1977. Since then, the ADS technologies and their implementations have interacted with the evolving markets, sciences, and technologies. The ADS concept is proposed on biological analogy, and its technologies have been advanced according to changing and expanding requirements. These technologies are now categorized into six generations on the basis of requirements and system structures, but the ADS concept and its system architecture have not changed. The requirements for the system can be divided in operation-oriented, mass service-oriented, and personal service-oriented categories. Moreover, these technologies have been realized in homogeneous system structure and, as the next step, in heterogeneous system structure. These technologies have been widely applied in manufacturing, telecommunications, information provision/utilization, data centers, transportation, and so on. They have been operating successfully throughout the world. In particular, ADS technologies have been applied in Suica, the IC card ticket system (ICCTS) for fare collection and e-commerce. This system is not only expanding in size and functionality but also its components are being modified almost every day without stopping its operation. This system and its technologies are shown here. Finally, the future direction of ADS is discussed, and one of its technologies is presented.

key words: *autonomous decentralized system, on-line property, assurance, IC card ticket system*

1. Introduction

As computing and communication resources have been gradually decreasing in cost, their roles in the society and business have become more important. Moreover, according to the continuous growth of practical applications and the networking of systems on a large scale, the systems have expanded and become more and more complicated. Therefore, the replacement of an entire system at once is impossible, and thus step-by-step construction without stopping its operation is required. In a large and complex system, it is not permitted to stop operation at any time. Even if a part of the system may fail or be repaired, the application has to keep performing its functions. Less restriction on the computing and communication resource results in more requirements for on-line property consisting of on-line expansion, on-line maintenance, and fault-tolerance.

Manuscript received September 3, 2007.

Manuscript revised August 26, 2008.

[†]The author is with Tokyo Institute of Technology, Tokyo, 152-8552 Japan.

^{††}The author is with East Japan Railway Company, Tokyo, 151-8578 Japan.

a) E-mail: mori@cs.titech.ac.jp

b) E-mail: shiibashi@jreast.co.jp

DOI: 10.1587/transcom.E92.B.445

Conventional computing technologies have been developed under a centralized system concept. Even a hierarchical and functionally-distributed system is based on the viewpoint that the total system structure and the functions have to be determined in advance [1]–[8]. This viewpoint itself is inconsistent with the system; the structure and the functions change continuously in the system although the hardware and software structures are fixed and have little flexibility.

To achieve on-line property, Autonomous Decentralized System (ADS) concept and architecture was proposed in 1977 on the basis of biological analogy [9]–[11]. Since then the ADS concept has been applied to various fields of technology such as networks, including the Internet, communications, multi-computers, software, control, and robotics [12]–[18]. Also, its architecture and technologies have been developed and applied in various fields such as factory automation, transportation, information systems, telecommunications, e-commerce, and so on. In these applications, ADS has improved lifecycle-cost efficiency, software productivity, flexibility, and adaptability. In the last 30 years, the market and users' requirements have been changed and diversified. At the same time, the ADS technologies have also advanced according to these evolving situations. Thanks to its wide range of applications, some of the ADS technologies have been approved as de-facto standard in many consortiums (ODVA, BAS, and OMG) [19]–[21], after which the International Symposium on Autonomous Decentralized Systems (ISADS) was founded (1993) under the sponsorship of IEEE, IEICE, IPSJ, and SICE. It has since been held every two years. As a result, research and development of ADS has advanced in the world not only for the control systems [22]–[28] but also for the information systems [29]–[36]. Recently, researches of ADS have been accelerated in the IT industry. The Autonomic Computing project, which is initiated by IBM in 2001, is an example [37], [38].

In this paper, the trend of ADS is discussed from the viewpoint of the requirements and the system structure. Then, as one application, the Suica system is explained. It is an IC card ticket system (ICCTS) [39]–[44] developed by East Japan Railway Company (JR East) for fare collection of transportation and e-commerce. Finally, the future direction of the ADS will be discussed.

2. Autonomous Decentralized System

Constraints due to cost in computing resources have been

reduced, therefore, the focus is moving from utilization efficiency to ease of use and construction. For these reasons, ADS' objective is to meet the requirements of on-line property. In the following subsections, the ADS concept and its architecture are described.

2.1 Concept

Opportunities and challenges for realizing highly complex, efficient, and dependable business-and-control systems have been steadily increasing. They are driven by continuous growth in the power, intelligence, adaptivity, and openness of technologies applied in computing, communication, and control systems. Dynamic changes in social and economic situations demand the next-generation systems to be based on adaptive and reusable technologies and applications. Such systems are expected to have the characteristics of living systems composed of largely autonomous and decentralized components. Such systems are called Autonomous Decentralized Systems (ADSs). Such a system is characterized as follows:

- (1) The system is the result of integration of subsystems.
- (2) In the system, being faulty is normal.

First, a subsystem exists, and the system is the integration of the subsystems. The objectives, structure, and functions of each subsystem should be clearly defined, but the total system cannot be clarified ahead of time. Some of the subsystems may be faulty and need to undergo repair and construction.

On this standpoint, a system is defined as an ADS if the following two properties are satisfied:

- (1) Autonomous Controllability: Even if one subsystem fails, is repaired, and/or is newly added, the other subsystems can continue to manage themselves and perform their own functions.
- (2) Autonomous Coordinability: Even if one subsystem fails, is repaired, and/or is newly added, the other subsystems can coordinate their individual objectives among themselves, and can operate in a coordinated fashion.

These two properties assure on-line property consisting of on-line expansion, fault tolerance, and on-line maintenance. They suggest that every "autonomous" subsystem requires an intelligence to manage itself without directing to or being directed from the other subsystems and to coordinate with the other subsystems [9], [45].

The ADS is realized with autonomous controllability and autonomous coordinability, and for that purpose, each subsystem is required to satisfy the following three conditions:

- (1) Uniformity (in structure): Each subsystem is uniform in structure and self-contained, so that it manages itself and coordinates with others.
- (2) Locality (in information): Each subsystem manages itself and coordinates with the others based only on local information.
- (3) Equality (in function): Each subsystem is equal in function. No master-slave relation exists among subsystems.

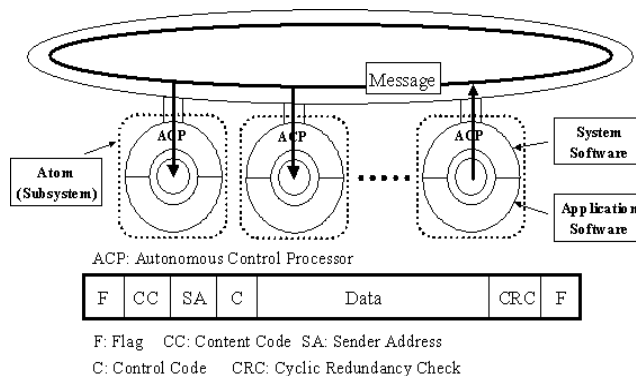


Fig. 1 Data Field architecture and message format.

2.2 Architecture

ADS is realized based on Data Field (DF) architecture without any central operating or coordinating system. The DF architecture is composed of two technologies: content-code communication for autonomous coordinability and data-driven mechanism for autonomous controllability. Each subsystem has its own management system, namely Autonomous Control Processor (ACP) to manage itself and coordinate with the others. Each subsystem, called an "Atom," consists of application software modules and ACP. The DF in the Atom is called the Atom Data Field (ADF).

(1) Content-code communication

All subsystems are connected only through the DF with a uniform interface (Fig. 1); all data are broadcast into the DF as messages. Individual datum includes a content-code defined uniquely by content. A subsystem selects to receive a message on the basis of its content code (Content-code communication). The sender does not need to indicate the receiver's address. This content-code communication enables each subsystem to be autonomous in sending and receiving data. That is, subsystems do not need to know the relationship among the sources and the destinations. This feature of the content-code communication ensures the locality of information which is necessary for each subsystem [9], [45].

In the conventional systems applied in P2P communication, the number of messages grows with the number of receivers increased. However, under DF architecture, all nodes can receive the broadcast message in one communication. Compared to the P2P communication, the number of messages can be reduced in the condition of multiple receivers. Moreover, with the increase of subsystems and communication traffic, the system can be divided into small DF structures as described in Sect. 3.2.3.

(2) Data-driven mechanism

Each application software module in the subsystem starts performing after all necessary data is received (Data-driven mechanism). This mechanism loosely couples modules. Each subsystem independently judges and controls its own action. Required content codes for application soft-

ware modules are pre-registered in the ACP, which can dynamically assign content codes based on changes in application software modules. The subsystem does not need to inform other subsystems if the content codes assigned to the ACP are changed. Each ACP has functions of managing the data, checking the data, and supporting the test and diagnosis (Fig. 1). The function of the application software module is characterized by the relation between the content codes of the input data and the output data.

In the conventional systems, data flow architecture is based on a centralized controller, such as a distribution network [46]. In those systems, the sequence and time of software execution are determined ahead of time. In ADS, each subsystem autonomously implements the software according to the content-code data in the DF.

3. Trends in ADS

3.1 The Evolution of Requirements and Technologies

From the view of the total system, conventional systems were required to achieve high performance, be highly reliable, efficient, and so on. However, with the scale of the system increasing so rapidly, the total system cannot be determined ahead of time, and the system structure itself is also changing constantly. As a breakthrough in systems, ADS was proposed to achieve on-line property.

Along with advanced information technology and changing demands, the technologies based on ADS' concept and architecture are also evolving gradually. In Table 1, the trend of ADS is seen from the viewpoint of system requirement and system structure.

(1) Requirement: From Operation to Service

Requirement has changed from system operation to user services. As the users' requirements have become more and more diversified, it is difficult to achieve user satisfaction even if the system operation is guaranteed. Therefore, the system's service itself has to be evaluated.

(2) Structure: Homogeneity to Heterogeneity

The system structure composition has been changed from homogeneous components to heterogeneous components. To meet users' increasing demands, businesses in various fields have to provide new services. In this business innovation, the systems have to treat the diverse kinds, quality, and quantity of the components and functions. As a result, the heterogeneous requirements can coexist, and the systems should adapt to changing situations.

(3) Trends

In homogeneous systems, the ADS architecture and technologies are applied to deal with continuous operation at system level. Moreover, with the advancement of information technology, ADS has focused on the integration of heterogeneous systems to assure the system operation. However, in the 90s, along with changing requirements, ADS was extended to the information service level to provide fair service and customized service in different applications. Currently, the ADS architecture and technologies have been

evolving in the service infrastructure to improve the end-users' quality of life. It is expected that, in the near future, the users not only get services but also actively create the services. Then, the ADS concept will play a more and more important role in the paradigm shift.

3.2 On-Line Property

3.2.1 Background

With globalization in the 1980s, intensity in competition increased. Therefore, it was not sufficient for companies to win a place in a highly competitive market simply by reducing costs and improving quality. For example, in the steel production process control system, it is necessary to meet the various types and quantities of production demands in global scale. Moreover, in the systems which dealt in seamless manufacturing processes, from raw material to the final product, non-stop system operation was required. The ADS technologies were originally proposed to achieve on-line property. The consumer requirements, technologies, and application shown in Table 1 are described below.

3.2.2 Requirements

A system may need to change according to user requirements. However, its operation cannot be stopped anytime since the system becomes economically and socially important. As a result, the role of on-line property of on-line expansion, on-line maintenance, and fault tolerance in a system, meaning that a system can continue operation during partial expansion, maintenance, and failure, became more and more important.

3.2.3 Architecture and Technologies

In the 1980s, as shown in Table 1, the ADS was targeted to be applied in the area of control. The structure of the control system was composed of homogeneous components. Therefore, the structure of DF(s) was also homogeneous. On-line property is attained by this DF architecture, in which all data are broadcast, and each subsystem selects to receive the data necessary for its application modules on the basis of the content codes (Fig. 2). This feature makes the modules loosely coupled. Even if some subsystems are under construction or fail, the system can continue its operation. On-line expansion, on-line maintenance, and fault tolerance technologies were thus proposed.

(1) On-line expansion

There are three levels of on-line expansion: module, subsystem, and system. In module level expansion, the application software module and database in one Atom are newly installed into or moved to another Atom. Then they need only to register their necessary content codes into their own ACPs and do not need to inform the others. In subsystem level expansion, a subsystem can be constructed, modified, added, and deleted during operation of the other sub-

Table 1 Trends in ADS.

Requirement Structure	Operation	Service	
	System	Mass	Personal
Homogeneous	Online property	Fair service	Customized service
	Technologies:		
	On-line expansion, On-line maintenance, Fault tolerance	Autonomous resource allocation, Autonomous stabilization	FIF architecture, Autonomous agents
	Application:		
	Steel production process control system	Data center
	Time frame: 1980s	Time frame: 1990s	Time frame: 2000s
Heterogeneous	DF Structure: Homogeneous DF(s)		
	Assurance	Unconscious service	Community service
	Technologies:		
	Heterogeneous integration, Autonomous data filtering	Autonomous cooperation process, Autonomous consistence
	Application:		
	ATOS	Suica
Time frame: 1990s	Time frame: 2000s		
DF Structure:			
Regional DFs	Timed DFs	Active DFs	

systems. The subsystems do not need to know the direct relation with others and need not inform others upon their addition to or deletion from the system. In the system level expansion, different ADSs are integrated into one. Two types of the systems integration are designed. In the first design, the DFs of different systems are combined into one DF. In the second design, the different systems are connected by a gateway, and two different DFs are combined through the gateway.

(2) On-line maintenance

DF architecture makes it easier for application software modules to be tested while the system is operating. There are two kinds of modes for each module: on-line mode and test mode. On-line data and test data coexist in the DF. There are two kinds of approaches for an on-line test. In the first approach, the module with on-line mode uses the test data to do the test. In the second approach, the module with the test mode uses the on-line data to make the test. Online test is supported by a BIT (Built-In Tester module) in each ACP and by an EXT (External Tester module) which is an application software module. The BIT module in the subsystem sets its application software module in the test mode, and then it generates test data, and checks the test result. The application software module in test mode receives data from the DF and processes it. Then it broadcasts test result data with a test flag to the DF. The BIT of the system in test mode prevents the signal from being sent to output devices such as controllers. The EXT monitors test data and test result data in the DF. By correlating test data with test result data, the EXT checks fault occurrence in the application software module in the test mode and broadcasts fault detection. The BIT independently decides whether to change the test mode to on-line mode based on test results.

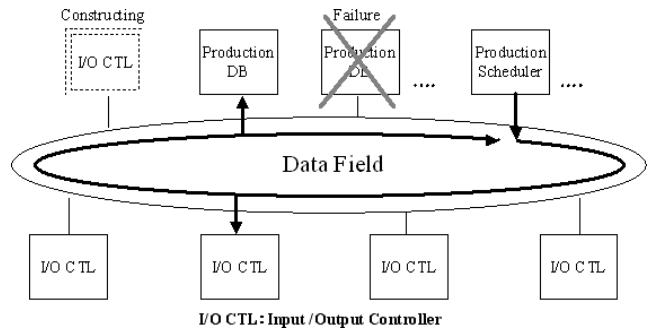


Fig. 2 DF architecture (1).

This test mechanism makes it possible for both on-line and test modes to coexist in the same system, at the same time.

(3) Fault tolerance

The ADS architecture and its data-driven mechanism makes it possible for the subsystems and application software modules to run freely and asynchronously. The subsystems and application software modules are replicated according to the requirements and their level of importance. Replicated application software modules run independently and send out processed data with the same content code to the DF. Faulty data are also sent out to the DF. The ACP in each subsystem receives all data with the same content code from replicated modules and selects the correct data from them. Here, the data consistency management module in the ACP identifies the same data both by content code and event number induced with the data. Correct data are selected from the same data through majority voting logic, which is flexibly adapted to the predetermined time interval or the total number of received data.

In the conventional k-out-of-N redundant system, the

redundancy is only in the subsystem level, and the voter detects the fault as a centralized controller. In the ADS, the redundancy is not only in the subsystem level but also in the module level. Each module autonomously detects the fault based on majority voting logic according to relative redundancy. Under this logic, fault occurrence is detected, and each application software module avoids being affected by fault propagation. A subsystem with a replicated module can intercept any data broadcast from other replicated modules. If the subsystem includes a faulty application software module, it detects the internal faults via this interception. As the results, even if an application software module is faulty, the subsystem continues operation by using correct data received from other replicated application software modules and recovers this fault by itself.

3.2.4 Application

As one application of ADS, a steel production process control system was proposed. To improve steel quality and to reduce cost, the software needed constant modification, revision, and testing. In this system, on-line expansion, on-line maintenance, and fault tolerance technologies were effectively utilized not only for the hardware system but also for the software system.

Production schedule data, broadcast into the DF from the steel production scheduling module, are received by real time In/Out Control (I/O CTL) subsystems. Each I/O CTL has its own responsible control region, and it autonomously adjusts its own schedule according to the situation by communicating with other I/O CTLs through the DF, as shown in Fig. 2. In this system, the application software modules are replicated according to their level of importance. Each module is driven only by the correct and necessary data received by the ACP. This autonomous data-driven mechanism makes it possible to expand, test, and repair the component during operation [47]–[49].

3.3 Assurance

3.3.1 Background

By the 1990s, non-stop control system operation had already reached high levels of efficiency. However, maintenance was manual and not yet automated. Moreover, labor costs had risen drastically. Therefore, not only the cost of operation but also the life-cycle cost, mainly consisting of maintenance cost, became a major consideration. Information systems to assist maintenance personnel required more and more details. It then became necessary to construct a heterogeneous system in which the facility's control system and maintenance information system were integrated. This structural shift brought about the trend from homogeneous to heterogeneous operation system, as shown in Table 1(a). For example, the Autonomous Decentralized Transport Operation Control System (ATOS) for Tokyo metropolitan area railway system is such a heterogeneous system in which

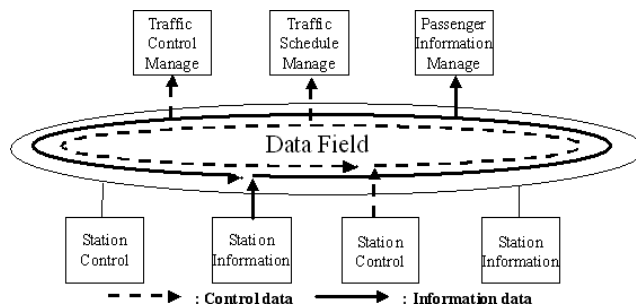


Fig. 3 DF architecture (2).

train control system and information system are integrated.

3.3.2 Requirements

An integrated system consisting of heterogeneous systems is required to keep operations safe and stable under heterogeneous properties and evolving conditions. This requirement is called assurance, which includes heterogeneity and adaptability [50].

3.3.3 Architecture and Technologies

The main feature of this architecture is not only that the structures of the DFs are heterogeneous, but also that various kinds of data of different quality levels are flowing through the DF simultaneously (Fig. 3). The heterogeneous data in the DF need to coordinate with other systems to achieve non-stop operation. Thus, autonomous heterogeneous integration technology and autonomous data filtering technology were proposed.

(1) Heterogeneous integration

Heterogeneous systems and content-code data coexist in the same DF. In this situation, to meet assurance requirements, especially for a mission-critical application in the integrated control and information systems, atomicity of the transaction process must be realized. Heterogeneous integration technology was proposed to guarantee atomicity by making coordination between heterogeneous subsystems both in control system and information system. When performing a transaction process, each control/information subsystem autonomously checks the atomicity of the processed data by cooperating with other correlated control/information subsystems. Based on transaction data flowing in the DF, each subsystem judges the completion of the transaction and commits the process autonomously [22].

(2) Autonomous data filtering

Autonomous data filtering technology is for assuring the different response time requirements in the integrated control and information systems, and not interference by each other in situations of change. In message suppression technology, the gateway monitors both DFs and judges whether or not to pass on the message based on the system's workload [23]. When the gateway receives transaction data,

it calculates the estimated response time based on both systems' workload, and compares it with the requirement. If the gateway decides not to pass it on, it sends suppressed data to the DF. In function filtering technology, to avoid the data function of control system violated by the data of information system, each gateway runs autonomously to avoid passing through unnecessary information messages to the control system.

3.3.4 Application

As shown in Fig. 3, ATOS is an application of ADS for integrating two heterogeneous systems: control systems such as route and traffic control, and information systems such as schedule and passenger information [24]–[27]. This system has been in development over the past 10 years, and some of the current parts will gradually be replaced even before the entire system construction is completed. Therefore, the inherited system, test system, and new system coexist at the same time. This system must be constructed step-by-step without stopping train service and disrupting operation of the currently installed parts of the system.

The system is composed of different regional DF structures. Each train line is composed of station subsystems, train-line traffic schedule management subsystem, and a train-line information service management subsystem. The network connecting both the station subsystems in the train line and the train lines are utilized for both control information missions. The control system is for real-time application, while the information system is required for high performance. In the station subsystem, the computers for control and those for information are divided and connected according to their own mission-oriented networks: namely, the control Ethernet and the information Ethernet through the gateway [28], [51].

In this system, the control system should use the train schedule and train delay data, which is generated by the information system. However, the control system is running in real time and has to assure the safe train operation. By using the autonomous data filtering technology, the control system can autonomously judge when and how to utilize data from the information system. Meanwhile, the information system can utilize control data for monitoring the train condition and rescheduling at any time.

3.4 Fair Service

3.4.1 Background

By the late 90s, on-line property of system operation was achieved. In addition, the Internet became an attractive alternate source of information. The advent of the Internet has generated new requirements for services from users [52]. Therefore, the concept of requirement has shifted from as operation of systems to the user services, as shown in Table 1(b). Moreover, the access from the users cannot be predicted. Each of service providers has a different service

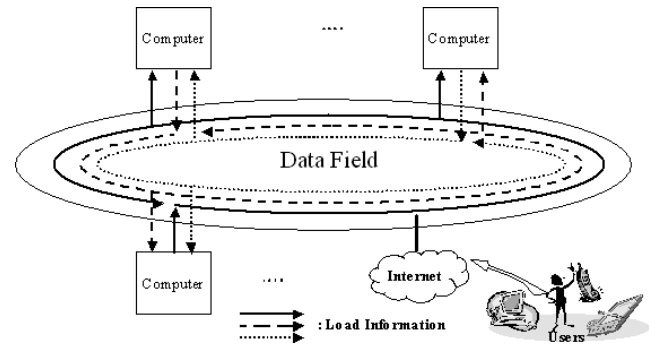


Fig. 4 DF architecture (3).

level, and the transactions change rapidly, and are unpredictable. Such a dynamic and heterogeneous environment has made it difficult for each service provider to manage computer systems independently. As a result, the data center comes forth to provide outsourcing of computers and management service to service providers. The task of a data center is to manage many computing resources for various service providers.

3.4.2 Requirements

A service provider must provide fair service, which means it must keep the same service level for all users with the same SLA (Service Level Agreement) without stopping its operation under evolving situations.

3.4.3 Architecture and Technologies

The homogeneous structure of DFs is constructed for the data center system so that the information system may provide fair service to users. In this system, the same service level, especially response time, is required for each computer. Therefore, each computer needs data to be able to grasp the situation of other computers. The characteristic of this system is that not only data but also information on the situation of each subsystem is broadcast into the DF. The subsystem exchanges its load information with the other subsystems and decides whether or not the subsystem should join the group to process requests for service (Fig. 4).

(1) Autonomous resource allocation

Autonomous resource allocation technology was proposed to provide and utilize fair service. To assure response satisfaction and avoid measurement delay, the load difference of each subsystem, which is the difference of necessary computing resource and actual deployed resource, is shared in the DF. Each subsystem works asynchronously and makes a decision autonomously according to the different sets of load differences. Autonomous load tracking measurement and control achieve quick response time by communicating load difference among subsystems and estimating total load using limited information gathered within a limited period [28], [29].

(2) Autonomous stabilization

As situations change frequently, measurement and decision would not be accurate. To raise the level of response satisfaction, it is more effective to track load change. However, frequent change of subsystems makes the system unstable. A tradeoff relationship exists between response satisfaction and stability. The system can be stable despite such errors since autonomous stabilization technology converges them according to feedback information [31], [32].

3.4.4 Application

In a conventional data center, it is possible to schedule computing resource allocation because load is predictable. Nowadays, however, many online applications on the internet, such as electronic ticket selling systems, have difficulty in predicting the users' requests in advance.

The size of data centers has been increasing due to growing number of customers. Thus, a more efficient utilization of computing resources is required. Moreover, the customers demand many different service levels, with many users at each service level. However, transactions cannot be predicted in the internet environment. Unpredictable peaks can arise within a short time. Autonomous load tracking technology is effective to distribute the load of the system among subsystems autonomously, and fair service can be achieved.

3.5 Unconscious Service

3.5.1 Background

By the beginning of the 21st century, services that are to be provided, should not only take user convenience in account, but also their quality of life as well. Service providers should offer appropriate service to users according to their situations. To realize this purpose, a heterogeneous system with heterogeneous requirement levels became necessary. This has made a change from homogeneous mass service to heterogeneous unconscious service, as shown in Table 1(c).

3.5.2 Requirements

In this system, a large number of the users utilize the system. The unconscious service, which means the users take for granted services unconsciously according to their own situations, is required for improving the quality of life.

3.5.3 Architecture and Technologies

Because there are different service contents, the process levels are also different. To meet different service process requirements, it is required to divide DF into heterogeneous levels and adapt different service processes (Fig. 5). In addition, it is difficult to implement the DF with high response and reliability at the same time. Therefore, different levels of heterogeneous timed DFs are constructed for different

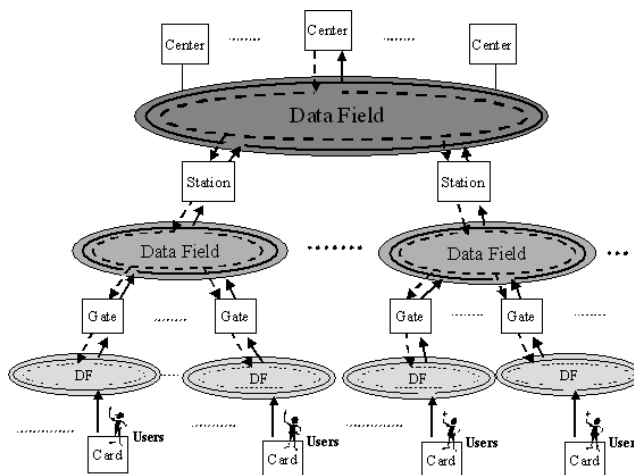


Fig. 5 DF architecture (4).

process levels such as high-response low-reliability and low-response high-reliability. To achieve high-performance and highly-reliable processes, different functions are distributed into heterogeneous DFs. Each DF autonomously executes the functions and coordinates with other DFs to continue its own operation even if data inconsistency occurs.

3.5.4 Application

This architecture and technologies have been applied in Suica, the ICCTS, introduced by JR East in November 2001, and can be regarded as the second infrastructure to combine transportation with e-commerce. The contactless IC card has made it possible to integrate smooth passenger flow (through real-time control of gate devices) with reliable information processing of fare calculation. The architecture and technologies of the Suica system are presented in the next chapter.

3.6 Customized Service

3.6.1 Background

Recently, advanced computer and communication technologies have made many services available to anyone, anytime, and anywhere. Conventional information systems, such as the internet, provide the same service to all users. However, users have a specific tendency to utilize information services with different preferences. Moreover, users' preferences diversify and change [53]. Therefore, the focus is evolving from provision of uniform services to offering personalized service according to preference, as shown in Table 1(d).

3.6.2 Requirements

The system is required to provide customized services, that is, services with various levels according to users' preferences. This is to cope with rapidly changing user demands, network status, and information content.

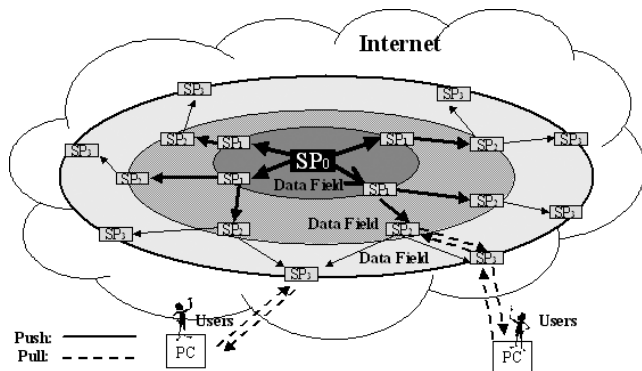


Fig. 6 DF architecture (5).

3.6.3 Architecture and Technologies

Service utilization varies in quantity and quality, and consequently the complete service offering is generally irrelevant to individual users. Usually, most users request a small part of available information, and few users request most of available information. As a result, the main feature of this architecture is that the service providers construct the DFs with different information service levels to satisfy the heterogeneous requirements of users (Fig. 6). However, the characteristic of each DF is homogeneous.

(1) FIF architecture

The “Faded Information Field” (FIF) was proposed to guarantee the heterogeneous requirements of providing and utilizing information service. Service providers trace information demand trend and send information to adjacent nodes. The nodes store the most accessed segment of information services, remove the less popular information in a recursive pruning process, and then send the remainder to other nodes. In this process, the DF is gradually constructed from large-data volume to small-data volume (Fig. 6). As a result, an area for multi-level distributed information services is created, called FIF. Users with different information requirements can be satisfied at different levels in the FIF. Consequently, the cost of service utilization (access time) and provision (update) are balanced by allocating the most accessed part of the information services closer to the majority of the users [33], [34].

(2) Autonomous agents

The SP generates Push-MAs and delegates to them the task of “information fading” Push-MA is responsible for allocating information based on the SP’s requirements for timeliness and reliability. Push-MAs autonomously coordinate with nodes for fading and selecting appropriate information according to the requirement of the node and importance level of the information. A user who requires information transmits a customized Pull-MA into the network. Pull-MA autonomously navigates through the distributed nodes in FIF to find more relevant information in a step-by-step fashion [35], [36].

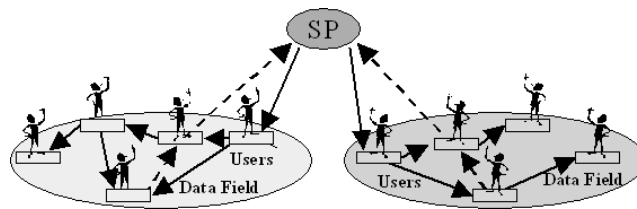


Fig. 7 DF architecture (6).

3.6.4 Application

This system is effective for applications in which the provision and utilization of large information volume are required. A simple prototype was developed based on the autonomous agents model to provide on demand streaming information with different qualities. It is effective to reduce the total storage volume and improve the response time. But still there are many areas in which the current model can be applied.

3.7 Community Service

3.7.1 Background

In the near future, users will require more mutual coordination in services. This requirement arises from unexpected situations and emergent events in the complex environment of modern society. This marks a shift from “service utilization” to “service creation” implemented by the input of users who share preferences and/or similar situations, as shown in Table 1(e).

3.7.2 Requirements

This requirement differs from the Internet requirement, which provides services to anyone, anytime, and anywhere. Under rapidly evolving situation, users with similar preference cooperate with each other not only sharing the services but also creating the services. Such services are characterized by “right me,” “right here,” and “right now” and are provided/utilized in accordance with the cooperation of users.

3.7.3 Architecture and Technologies

To meet these requirements, users with similar preferences organize a community. Each user can autonomously and actively form a local community with other users based on physical locations, time, and the kind of service. This communication field constructed by community users is called active DF, in which each user broadcasts information into it and shares information with other users (Fig. 7). Moreover, users create services through the cooperating with each other based on the shared information. The DF changes by time and place, and the members of the community also interchange.

By using autonomous construction technology, the size of the community is determined based on the required service level. Other technologies are proposed for autonomously reconstructing, integrating and dividing sub-communities according to different service level requests, and holding resources of community members under changing situations. However, community system still includes many research topics.

4. Suica: IC Card Ticket System

ICCTS is explained here as a typical example of unconscious service by ADS. It was developed by JR East and called “Suica.” Suica has been utilized not only for fare collection by JR East and the private railways, but also for e-commerce.

It is difficult to meet heterogeneous requirements, such as high response and reliability for fare collection and e-commerce, in the DF simultaneously. To meet the time constraints in different service process requirements, the DF is divided into different time levels. Therefore, the heterogeneous timed DFs are constructed in the Suica system.

4.1 System Structure

ICCTS consists of IC cards, terminals (Automatic Fare Collection Gates, Ticket Vending Machines, and Fare Adjusting Machines), station servers, and a center server. Each station has several terminals and a station server, all of which are connected to each other via LAN within one station. Station servers are connected to the center server via WAN.

Here, there are three DFs among the subsystems (Fig. 8): (1) DF1 between an IC card and a terminal, (2) DF2 among terminals and a station server within a station, and (3) DF3 among station servers and the center server throughout stations.

These three DFs have different transmission methods with the respective time ranges determined by the needs of the subsystems. The subsystems attached to DF1 have functions for payment, fare collection, and gate control. It is required for them to achieve real-time processing in order to avoid congestion. Thus, the total processing time in DF1

is designed to be under 0.2 second. DF2 and DF3 transmit the data through wired network hourly and daily. The station servers and the center server do not serve the hurrying passengers directly but deal with their accounts. Hence, the wired network and the longer data storage periods have been adopted to assure reliability of the data as information systems.

In addition, a “virtual data field” is applied to the system. This “virtual data field” is the field of the IC cards that are actually carried by the passengers. Here, an IC card is regarded as a method of data transmission, and the data are accessed when needed at the terminals.

These DFs have different characteristics and are named “Heterogeneous DFs,” and the total structure is named “Heterogeneous ADS Structure.” This structure is one of the properties in the ICCTS. The reason the heterogeneous DFs are introduced is to assure (1) high performance (real-time operation) and (2) fault-tolerance. Two technologies, which are necessary for the properties of real-time processing and transactions, are described below.

4.2 Autonomous Cooperative Processing Technology

Under the heterogeneous timed DFs architecture, each subsystem cooperates with other subsystems, and distributes processes autonomously based on the local information. Autonomous Cooperative Processing Technology was proposed to guarantee high-speed process of each subsystem by making cooperation among them according to the characteristic of each transaction.

4.2.1 Technology

Processing time at terminals in DF1 is required to be under 0.2 second. In this short time, they need to detect, authenticate, read, judge, write, and verify IC cards [40] in addition to fare calculation, which takes the most of the time of all processes.

There are two problems with fare calculation. The first problem is the highly complicated fare system in Japan, in which the fares are subject to distance in kilometers, while they are fixed at flat rates or based on simple zones in other countries. There are so many fare combinations equal to the number of combinations of stations. Moreover, some passengers hold “commuter passes” which allow unlimited rides within the passenger’s predetermined commuting zone. The fare calculations must take the passes into consideration if it is less expensive.

The second problem is that the passengers’ destinations are unpredictable. In a conventional magnetized ticket system, passengers have to check the fare table, buy their own tickets, and pay additional fare upon exiting if they did not pay enough for their original tickets. In the ICCTS, the passengers no longer need to buy tickets in advance — it is very convenient for them, but the terminals at the entrances have no way to know their destination. It takes a long time if fare calculation is done entirely at the exit; the terminals must

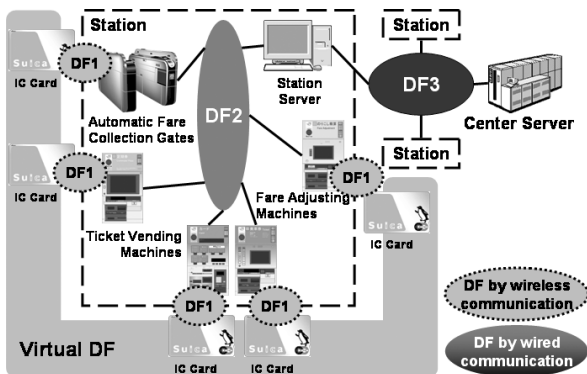


Fig. 8 IC card ticket system.

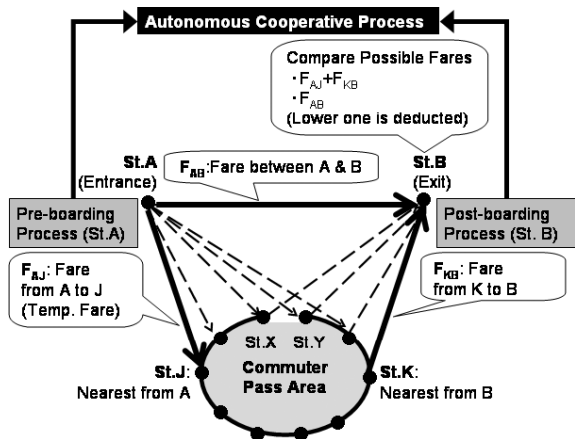


Fig. 9 Autonomous cooperative processing technology.

scan the complete fare list including the fares from the entrance station to the exit station.

Autonomous cooperative processing technology was proposed to resolve these two problems, using virtual DF [41]. In this technology, the fare calculations are divided into two steps (upon entrance and exit), and the necessary information is transmitted by cards carried by passengers who move within the virtual DF. The procedure is shown in Fig. 9. In this case, a passenger has a commuter pass, which is valid from station J to station K, and travels from station A to station B, both of which are outside of the valid commuter pass area. Within the commuter pass area, station J is the nearest station to station A, and station K is the nearest to station B. There are two possible fares: (1) the direct fare from station A to station B or (2) the sum of the fares from station A to station J and from station K to station B.

When the passenger enters the station A, the terminal determines that it is out of the valid commuter pass zone and selects station J, the nearest within the valid commuter pass area. Then, it writes on the IC card that the cardholder gets on at station A, and that the temporary fare to the nearest station J is F_{AJ} (pre-boarding process).

When the cardholder exits from station B, the terminal judges that it is also out of the valid commuter pass zone and selects station K, the nearest within the valid commuter pass area (post-boarding process). Then it calculates the fare between station K and station B (F_{KB}) and compares two possible fares: (1) the direct fare between station A and station B (F_{AB}) and (2) the sum of F_{AJ} and F_{KB} . The less expensive fare is deducted from the IC card (autonomous cooperative process).

4.2.2 Evaluation

The effectiveness of this technology is shown in Fig. 10 and Fig. 11. These are the results from simulations comparing the calculating time and the process time with autonomous cooperative processing technology to the ones without it. With the technology, the fare calculations are divided into two stages: upon entrance at A and upon exit at B. Without

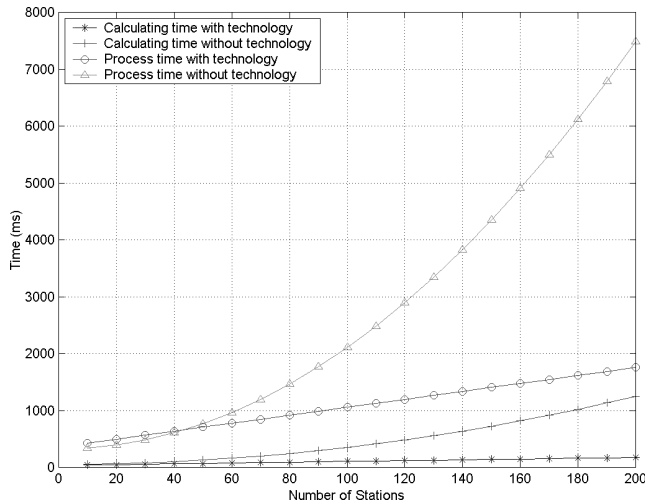


Fig. 10 Comparison by number of stations.

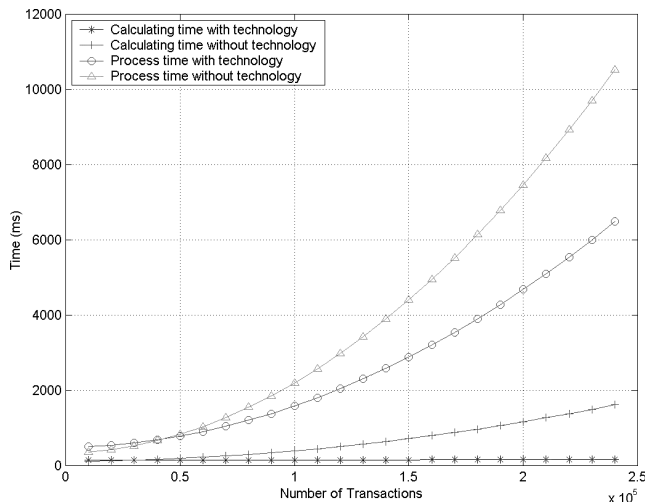


Fig. 11 Comparison by number of transactions.

it, all the fare calculations are done upon exit from station B. The “calculating time” is the simple calculating time, and “process time” is the total time for a passenger to go through a gate, including both “calculating time” and waiting time at gates. Each station is supposed to have 10 gates.

Figure 10 shows the times in relation to the number of stations. Autonomous cooperative processing technology is superior (spending less time in calculating and processing) when the number of stations exceeds 43. Figure 11 shows the times as the number of the transactions. The autonomous cooperative processing technology is superior when there are over 42,700 transactions per day.

According to these results, the higher the number of stations and transactions, the more effective the autonomous cooperative processing technology is.

The number of stations in the Suica system is 647, and the number of the transactions is approaching 20 million per day. Thus, autonomous cooperative processing technology is most effective in the situation that the number of stations

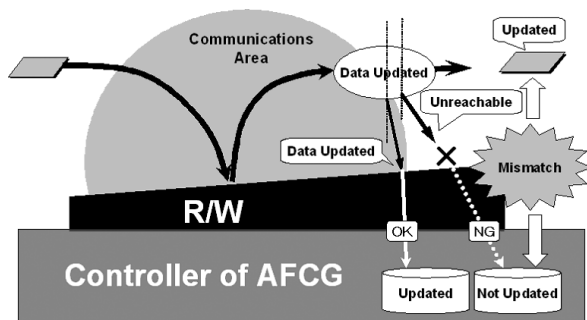


Fig. 12 Data processing at the R/W.

and transactions are high [41].

4.3 Autonomous Decentralized Data Consistency Technology (ADDCT)

The main objective of Suica is to assure the fluidity of passengers. To guarantee real-time process under high transaction, incomplete data is passed and coexists with complete data in DFs. ADDCT was proposed to make data consistency under the heterogeneous timed DFs architecture by cooperating with each other and determining residence time of data in each DF autonomously according to the data distribution.

4.3.1 Technology

The ICCTS uses wireless communications at DF1. It is very convenient for passengers, but it is prone to instability in communication due to improper card use. This section introduces the autonomous decentralized data consistency technology (ADDCT), which recovers missing data caused by such unstable communication [40]. This technology has two derivatives: Single-layered data consistency technology [42] and Multi-layered data consistency technology [43], [44].

In the ICCTS, an IC card can communicate with a Reader/Writer (R/W) on the gate while it is within the communication area. The time of staying within the communication area depends on the holder’s behavior. According to the statistics, the minimum required time is 0.2 second [40], [54], [55], the time period which the autonomous cooperative processing technology targets. However, some passengers do not handle the cards properly, and the card thus cannot be processed.

The way the ICCTS detects the end of process is shown in Fig. 12. Each R/W unit updates its data when it receives a “data-process completed” signal from the IC card. This last signal is transmitted near the border of communications area, so the process is not always completed successfully when the passengers handle their IC cards improperly. In this case, although the data in the IC card is updated, the R/W has not received any signal indicating that the update is complete. This is a problem called “data missing.” A conventional magnetized ticket system would shut the gate,

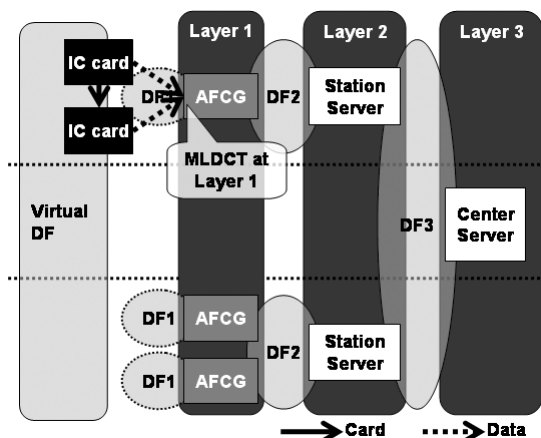


Fig. 13 ADDCT application running at Layer 1.

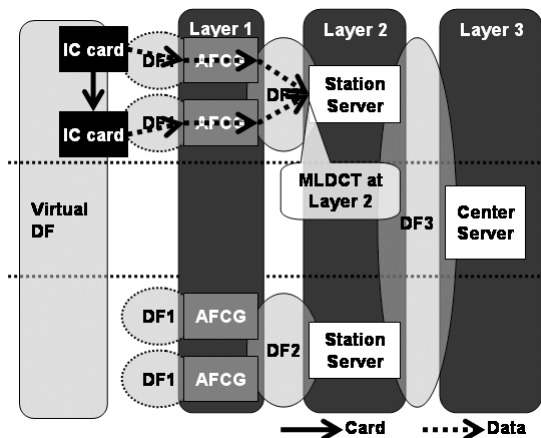


Fig. 14 ADDCT application running at Layer 2.

which completely governed the contact signals, thus “data missing” was practically unheard of. However, if ICCTS did so, serious congestion would occur which could lead to accidents. Opening gates even in case of failures has been a problem with ICCTS.

ADDCT is a technology that recovers the data from a failure, considering data consistency. Here, the data in the Rs/Ws are treated as “temporary data” even when the Rs/Ws cannot catch the “data-process completed” signal. If the next process is completed normally, the ADDCT checks the consistency of the data and revises the “temporary data” as “definite data.”

Samples of ADDCT are shown in Figs. 13 to 15. Subsystems are classified into several layers. For example, the ICCTS has three layers: (1) the gate at Layer 1, (2) the station server at Layer 2, and (3) the center server at Layer 3. Since the ADDCT application runs at each subsystem, “missing data” has only to meet the partner data to be recovered. If not, data will be broadcast to the lower layer, where the ADDCT application runs again.

These three figures are summarized in Fig. 16. Here, three data (Data 1, Data 2, and Data 3) are created first. Then, they are stored at Gate₁₁ in Layer 1 for time t_1 , and

then the ADDCT application runs for time t_{1p} . While being stored, Data 4 to match Data 1 catches up at Gate₁₁ at Layer 1. Hence, the ADDCT application adjusts the data, and the rest (Data 2 and Data 3) are broadcast to DF2. Data 5 from Gate₁₂ in Layer 1 joins them at Station Server₂₁ at Layer 2 through DF2 to match Data 2. Those three data are stored for the time t_2 , and then the ADDCT application runs for the time t_{2p} . Data 2 is adjusted, and the last one (Data 3) is again broadcast to DF3 where Data 6 comes from Station Server_{2x} at Layer 2.

As there are fewer layers, the number of transactions per node is greater; the processes at nodes are jammed, and the total staying time lengthens.

How the ADDCT application recovers “missing data” is shown in Figs. 17 to 19. This is an example of ADDCT application running at Layer 3. A passenger has an IC card

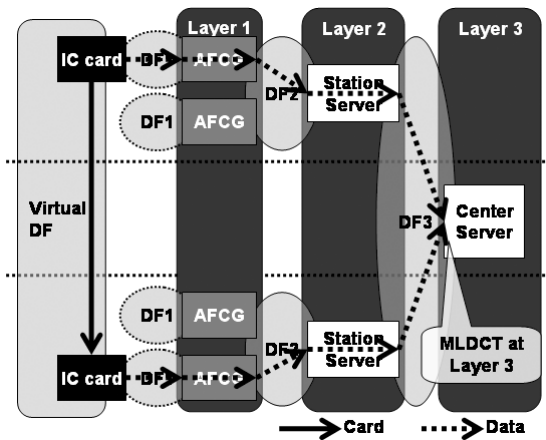
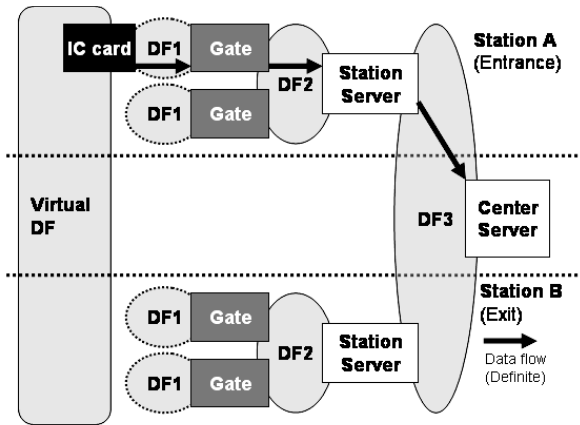


Fig. 15 ADDCT application running at Layer 3.

with the value of 1,000 yen and travels from station A to station B. The possible fare of 130 yen is written at a gate in station A. The gate broadcasts the data to the DFs as “definite” with a sequential number (#14 in this sample) when the process is successfully completed, and the R/W receives “data-process completed” signal (Fig. 17).

Suppose that an R/W at station B fails to receive the signal indicating the completion of data’s processing, although it has actually been processed in the passenger’s IC card itself. In this case, the gate autonomously broadcasts the unconfirmed data to the DFs as “temporary” (sequential number 15) (Fig. 18).



Data at Center Server

Seq. #	Status	Date & Time	Used	Left
				1000
14	Definite	2008/04/01 10:00	-130	870

Fig. 17 ADDCT application at Layer 3 (1): complete process at R/W.

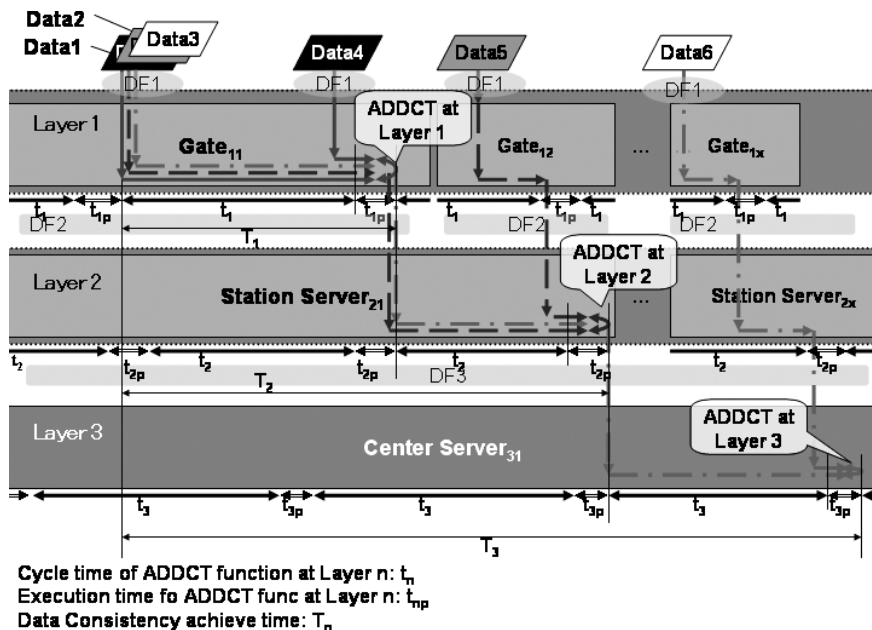


Fig. 16 ADDCT model.

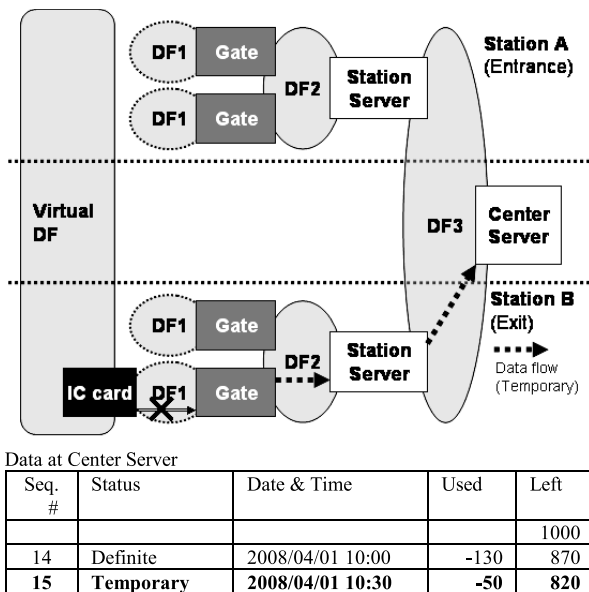


Fig. 18 ADDCT application at Layer 3 (2): incomplete process at R/W.

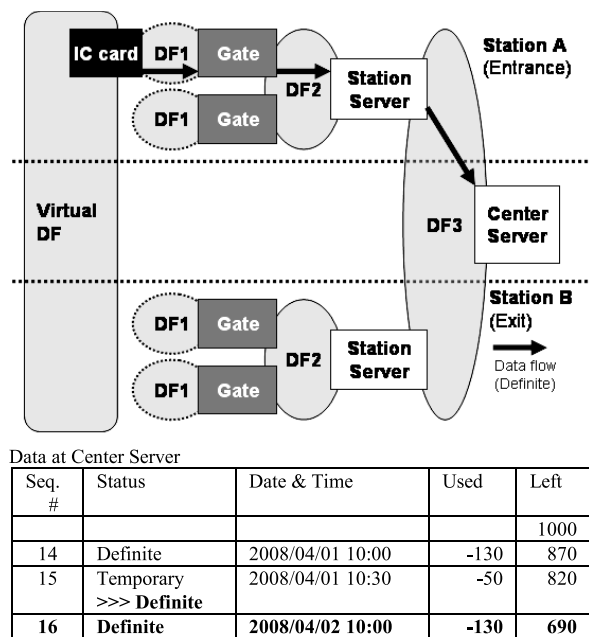


Fig. 19 ADDCT application at Layer 3 (3): another complete process at R/W and data recovery.

If the passenger uses the same card and completes the processes upon entrance at station C, the data numbered 16 is “definite.” The center server checks those sequences and changes its status from “temporary” to “definite” if the “temporary” record is surrounded by “definite” data without inconsistency (Fig. 19).

With ADDCT, IC cards can escape from being voided even if passengers have caused “data missing.” The data are recovered before being blacklisted. That is, this technology assures reliability of data with integration of autonomous processes at the terminals and ones at the center server.

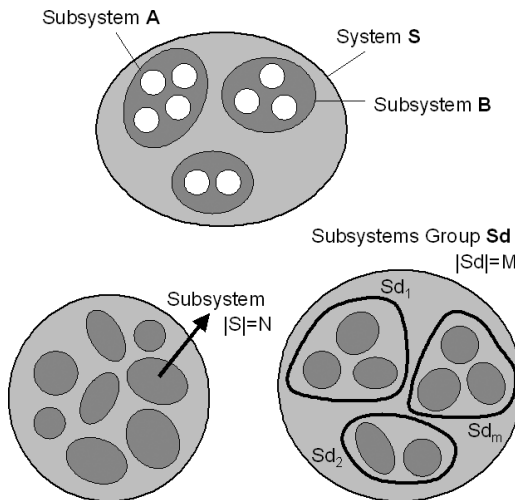


Fig. 20 Function reliability evaluation model.

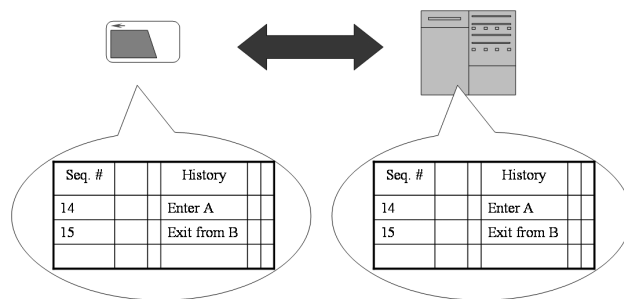


Fig. 21 Consistency of data.

4.3.2 Evaluation

(1) Function reliability

ADS guarantees smooth operation even with partial failure. Therefore, function reliability is more a suitable criteria by which to evaluate the architecture since it evaluates the functioning portion of subsystems with the ability to cooperate and integrate with one another, while the conventional method of reliability evaluation judges whether total function is achieved or not [56]–[58].

A system model for evaluating function reliability is shown in Fig. 20. It shows a system consisting of several subsystems (or subsystem groups), and each subsystem has certain functions. Each unit is valued by amount of function.

This section takes “function reliability” into consideration for evaluation of ICCTS. The basic functions of the ICCTS are to check for invalid cards and calculate fares. Since these functions are related to data found in the cards, each subsystem has “consistent data.” The achievement of functionality in terms of ICCTS is defined as the “consistency of data,” which means how much data the servers and the cards have in common (Fig. 21).

Accordingly, ICCTS is modeled as seen in Fig. 22. Each pellet shape seen within the subsystem represents one piece of consistent data. In the actual ICCTS, servers save

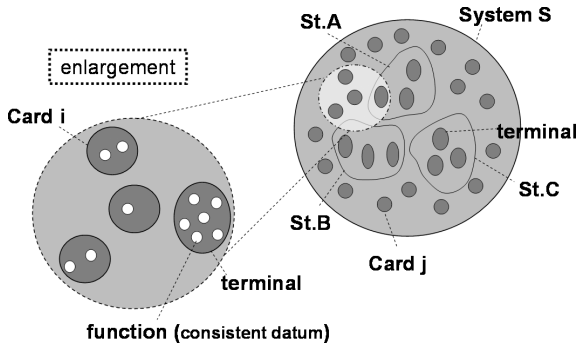


Fig. 22 Model for evaluating function reliability.

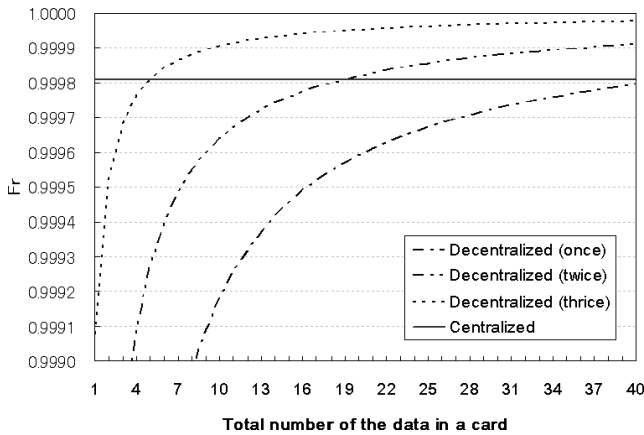


Fig. 23 Function reliability with/without ADDCT.

the data simply as backup and use them to facilitate greater reliability. This simulation focuses on modeling cards and terminals since each terminal checks invalid cards and calculates fares.

(2) Evaluation Results

The effectiveness of the ADDCT in enhancing data reliability is shown in Fig. 23, according to simulations based on actual transactions. The amount of data which a card is able to store is shown along the X axis, and the average function reliability is plotted along the Y axis. Function reliability in the ICCTS is evaluated in terms of data consistency, which defines stable operation of a system in terms of how well it guarantees the provision of reliable data. In the simulation, four cases are examined: one system each which runs the ADDCT application once a day, twice a day, and three times a day; and a centralized system without ADDCT.

ADDCT can make up for as much missing data as there is available number of record space in the card. The more records a card is able to store, the more “missing data” is accepted. The results from the simulations prove that if there is sufficient data, the ADS structure is more reliable with ADDCT. However, it was found that there is no apparent merit in increasing the number of records in one card to more than 20. Hence, it is very important to determine the appropriate memory size. In the Suica system, each IC card is capable of keeping 20 records at a time, and the ADDCT application

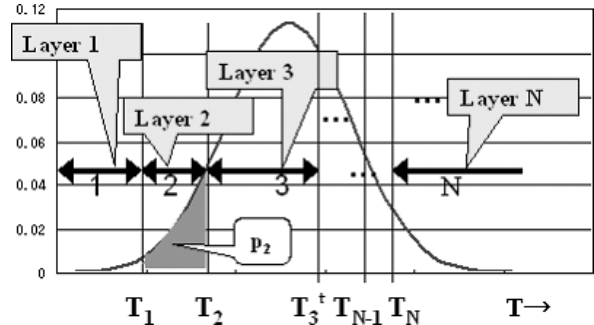


Fig. 24 Distribution rate of passengers' traveling times.

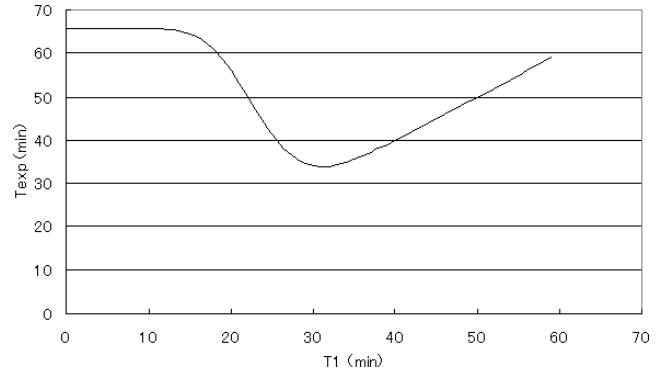


Fig. 25 T1 vs. Texp in 2-layered models.

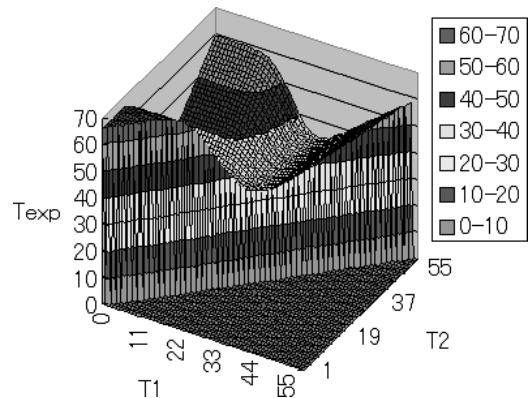


Fig. 26 T1 vs. Texp in 3-layered models.

runs three times a day. According to Fig. 23, the system on the condition scores the highest function reliability among all systems. This means that the effectiveness of the ADDCT is proven both practically and theoretically [42].

Then, what if the ADDCT application runs in each subsystem at each layer? To evaluate this, a sample distribution of the passengers' traveling times is prepared as seen in Fig. 24. The data in such distribution is divided into groups based on T ($0 < T < T_1$, $T_1 < T < T_2$, and so on), and each group is assigned to a layer where it is recovered. The simulations are intended to find the most effective configuration to deal with the transactions: the number of layers and the time each layer holds the data.

Its effectiveness is shown in Fig. 25 and Fig. 26. Fig-

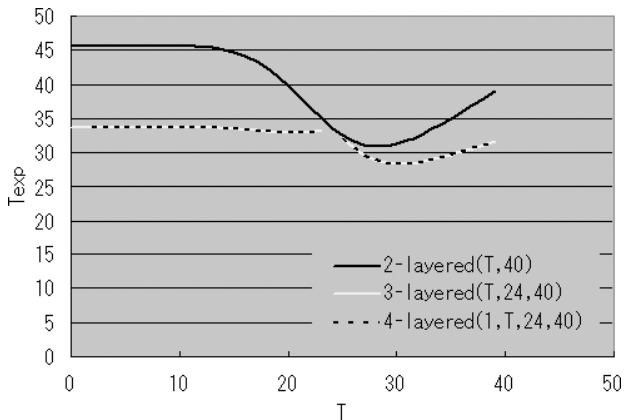


Fig. 27 Texp in 2-, 3-, and 4-layered simulations.

ure 25 shows the result from a 2-layered model where T_1 ranges from 1 minute to 59 minutes while T_2 is fixed at 60 minutes. T_{exp} , the expectation of recovery time, indicates the minimum value of 33.82 when $T_1 = 31$. This means, in the most effective configuration of 2-layered structure, the ADDCT application at Layer 1 should be set to run 31 minutes after the flow starts.

Figure 26 shows the result from the 3-layered model, where T_1 and T_2 ranges from 1 minute to 59 minutes while T_3 is fixed at 60 minutes. T_{exp} indicates a minimum value of 29.72 when $T_1 = 26$ and $T_2 = 34$. According to these results, in the most effective configuration of 3-layered structure, the ADDCT application at Layer 1 should be set to run 26 minutes after the flow starts at Layer 1, and the ADDCT application at Layer 2 should be set to run 34 minutes after. In addition, T_{exp} is less in a 3-layered model than in a 2-layered model. That is, the 3-layered model is more efficient in this input flow.

T_{exp} depends on the number of the layers (indicated as n) and the timing of the application (indicated as T_n). The way to design the most efficient system is to find n and T_n to minimize T_{exp} . To compare the results from the simulations more easily, Fig. 27 shows the most appropriate parameters in each layered simulation. The minimum values of T_{exp} are 30.96 in the 2-layered model, 28.43 in the 3-layered, and 28.44 in the 4-layered. The 3-layered model performs better than the 2-layered or the 4-layered. These results reveal two facts; more layers work more efficiently but too many layers become useless because the nodes cannot gather enough diffused data to match up [43], [44].

5. Conclusion

In this paper, the advancement of ADS technologies was surveyed in accordance with the changing and growing requirements and their corresponding system structures in the last 30 years. The Suica system operated by JR East is one of the most advanced implementations of ADS, and it has been utilized not only for information transaction processing in fare calculation and e-commerce but also for controlling AFCG under 0.2 second.

These research activities of ADS extend to computer, communications, and control technologies, and their integration. It is expected that ADS created in Japan will play an active role in collaboration of academia, industry, and government around the globe.

References

- [1] "Special Issue on Large-scale Systems and Decentralized Control," *IEEE Trans. Autom. Control*, vol.AC-23, no.2, pp.105–371, 1978.
- [2] W.D. Barritt, "Centralized control system for appliances," *IEEE Trans. Ind. Appl.*, vol.24, no.2, pp.328–331, 1988.
- [3] B. Ciciani, D.M. Dias, B.R. Iyer, and P.S. Yu, "A hybrid distributed centralized system structure for transaction processing," *IEEE Trans. Softw. Eng.*, vol.16, no.8, pp.791–806, 1990.
- [4] M. Jeng and H.J. Siegel, "A distributed management scheme for partitionable parallel computers," *IEEE Trans. Parallel Distrib. Syst.*, vol.2, no.1, pp.120–126, 1990.
- [5] "Special Issue on Distributed Computing Systems," *Computer*, vol.24, no.8, 1991.
- [6] P. Jalote, *Fault Tolerance in Distributed Systems*, Prentice Hall, Englewood Cliffs, NJ, 1994.
- [7] R. Adler, "Distributed coordination models for client/server computing," *Computer*, vol.28, no.4, pp.14–22, 1995.
- [8] G. Coulouris, J. Dollimore, and T. Kindberg, *Distributed Systems, Concepts and Design*, Addison-Wesley, Wokingham, 2001.
- [9] K. Mori, S. Miyamoto, and H. Ihara, "Proposition of autonomous decentralized system concept," *IEEJ Trans. EIS*, vol.104, no.12, pp.303–340, 1984.
- [10] K. Mori, "Autonomous decentralized software structure and its application," *IEEE Proc. FJCC86*, pp.1056–1063, 1986.
- [11] K. Mori, *Introduction of Autonomous Decentralized System*, Morikita Pub., 2006.
- [12] S. Shenker, "An architecture for the future Internet," *IEEE Proc. Hot Interconnects II Symposium Record*, pp.261–267, 1994.
- [13] H. Hultzsch, "Network evolution: Convergence of decentralised information technology and telecommunications," *IEEE Proc. ISADS97*, pp.3–8, 1997.
- [14] J. Kaiser, "Uncertain, dynamic, out of control? Coordination and cooperation in next generation systems," *IEEE Proc. ISADS03*, pp.165–168, 2003.
- [15] P. Bellavista, A. Corradi, and C. Stefanelli, "The ubiquitous provisioning of Internet services to portable devices," *IEEE Pervasive Computing*, vol.1, no.3, pp.81–87, 2002.
- [16] S. Yau and G.H. Oh, "An object-oriented approach to software development for autonomous decentralized systems," *IEEE Proc. ISADS93*, pp.37–43, 1993.
- [17] K.H. Kim and C. Subbaraman, "Interconnection schemes for RTO.k objects in loosely coupled real-time distributed computer systems," *IEEE Proc. COMPSAC97*, pp.121–128, 1997.
- [18] A.W. Scheide, "Robotics," *IEEE Trans. Ind. Appl.*, vol.IA-19, no.6, pp.901–902, 1983.
- [19] <http://www.odva.org/default.aspx>
- [20] <http://www.ieiej.or.jp/standard/index.html>
- [21] <http://www.omg.org/docs/orbos/99-11-05.ppt>
- [22] I. Kaji, Y. Tan, and K. Mori, "Autonomous data synchronization in heterogeneous systems to assure the transaction," *IEEE Proc. HASE1999*, pp.169–178, 1999.
- [23] I. Kaji and K. Mori, "A gateway filtering technique to maximize the transactions in heterogeneous systems," *IEICE Trans. Commun.*, vol.E84-B, no.10, pp.2759–2767, Oct. 2001.
- [24] T. Kobayashi, O. Iba, H. Ebine, and S. Aoyagi, "Advanced train administration and communication system based on ADS technologies," *IEEE Proc. ISADS1999*, pp.388–391, 1999.
- [25] K. Kera, K. Bekki, K. Fujiwara, F. Kitahara, and K. Kamijo, "Assurance system technologies based on autonomous decentralized

- system for large scale transport operation control systems," *IEICE Trans. Commun.*, vol.E83-B, no.5, pp.1085-1093, May 2000.
- [26] T. Aizono, K. Kawano, H. Wataya, and K. Mori, "Autonomous decentralized software structure for integration of information and control systems," *IEEE Proc. COMPSAC1997*, pp.324-331, 1997.
- [27] F. Kitahara, K. Kamijiou, Y. Kakurai, K. Bekki, K. Kare, and K. Kawano, "Phased-in construction method of ATOS," *IEEE Proc. ISADS1999*, pp.415-424, 1999.
- [28] M. Matsumoto, A. Hosokawa, S. Kitamura, D. Watanabe, and A. Kawabata, "Development of the autonomous decentralized train control system," *IEICE Trans. Commun.*, vol.E84-D, no.10, pp.1333-1340, Oct. 2001.
- [29] T. Masuishi, H. Kuriyama, Y. Ooki, and K. Mori, "Autonomous decentralized resource allocation for tracking dynamic load change," *IEEE Proc. ISADS2005*, pp.277-283, 2005.
- [30] T. Masuishi, K. Shibata, Y. Ooki, and K. Mori, "Autonomous decentralized load tracking techniques and evaluation," *IEEE Proc. 2nd International Symposium on Dependable, Autonomic and Secure Computing (DASC 2006)*, pp.69-76, 2006.
- [31] T. Masuishi, K. Shibata, Y. Ooki, and K. Mori, "Techniques to improve tracking ability in autonomous decentralized load tracking system," 4th International Conference on Computing, Communications and Control Technologies, 2006.
- [32] T. Masuishi, K. Shibata, Y. Oki, and K. Mori, "Autonomous decentralized load tracking systems and evaluation criteria for response and stability," *IEEE Proc. ISADS2007*, pp.255-262, 2007.
- [33] K. Mori, "Autonomous fading and navigation for information allocation and search under evolving service system," *Proc. IEEE Conf. on APSITT*, pp.326-330, 1999.
- [34] H.F. Ahmad and K. Mori, "Autonomous information service system: Basic concepts for evaluation," *IEICE Trans. Fundamentals*, vol.E83-A, no.11, pp.2228-2235, Nov. 2000.
- [35] H. Arfaoui and K. Mori, "Autonomous navigation architecture for load balancing user demands in distributed information systems," *IEICE Trans. Commun.*, vol.E84-B, no.10, pp.1085-1093, Oct. 2001.
- [36] X.D. Lu, H. Arfaoui, and K. Mori, "Autonomous information fading and provision to achieve high response time in distributed information systems," *IEEJ Trans. EIS*, vol.125, no.4, pp.645-652, 2005.
- [37] IBM, "Autonomic computing: IBM's perspective on the state of information technology," <http://www-1.ibm.com/industries/government/doc/content/resource/thought/278606109.html>
- [38] J. Kephart and D. Chess, "The vision of autonomic computing," *Computer*, vol.36, no.1, pp.41-50, 2003.
- [39] Y. Shirakawa and A. Shiibashi, "JR East contact-less IC card automatic fare collection system "Suica"," *IEICE Trans. Inf. & Syst.*, vol.E86-D, no.10, pp.2070-2076, Oct. 2003.
- [40] A. Shiibashi, "Autonomous decentralized high-speed processing technology and the application in an integrated IC card fixed-line and wireless system," *IEICE Trans. Inf. & Syst.*, vol.E88-D, no.12, pp.2699-2707, Dec. 2005.
- [41] A. Shiibashi, N. Mizoguchi, and K. Mori, "High-speed processing in wired-and-wireless integrated autonomous decentralized system and its application to IC card ticket system," *Innovations in Systems and Software Engineering*, vol.3, no.1, pp.53-60, March 2007.
- [42] A. Shiibashi, T. Kuroda, M. Yamana, and K. Mori, "Research of reliability technology in heterogeneous autonomous decentralized assurance systems," *ISADS*, pp.207-214, Sedona, US., March 2007.
- [43] A. Shiibashi, M. Yamana, and K. Mori, "Multi-layered data consistency technology in IC card ticket system," *Assurance Symposium*, pp.1-8, Yokohama, Japan, March 2007.
- [44] A. Shiibashi, Y. Maruyama, M. Yamana, and K. Mori, "Multi-layered data consistency technology in IC card ticket system," *ADSN*, p.58, Toronto, Canada, June 2007.
- [45] K. Mori, "Autonomous decentralized systems: Concepts, data field architecture and future trends," *IEEE Proc. ISADS93*, pp.28-34, March 1993.
- [46] J. Dennis, "Data flow supercomputer," *Computer*, vol.13, no.11, pp.48-56, 1980.
- [47] K. Mori and H. Ihara, "Autonomous decentralized loop network," *COMPCON Spring*, pp.192-195, 1982.
- [48] H. Ihara and K. Mori, "Autonomous decentralized computer control system," *Computer*, vol.17, no.8, pp.57-66, 1984.
- [49] K. Mori, H. Ihara, Y. Suzuki, K. Kawano, M. Koizumi, M. Orimo, K. Nakai, and H. Nakanishi, "Autonomous decentralized software structure and its application," *IEEE Proc. FJCC86*, pp.1056-1063, 1986.
- [50] I-Ling Yen, R. Paul, and K. Mori, "Toward integrated methods for high-assurance systems," *Computer*, vol.31, no.4, pp.32-34, 1998.
- [51] K. Mori, "Trend of autonomous decentralized systems," *IEEE Proc. FTDCS04*, pp.213-216 May 2004.
- [52] D. Oppenheimer and D.A. Patterson, "Architecture and dependability of large-scale Internet services," *IEEE Internet Comput.*, vol.6, no.5, pp.41-49, 2002.
- [53] C.W. Bachman, "A personal chronicle: Creating better information systems, with some guiding principles," *IEEE Trans. Knowl. Data Eng.*, vol.1, no.1, pp.17-32, 1989.
- [54] Y. Naka, "Study on complicated passenger flow in a railway station," *Railway Technical Research Report*, no.1079, 1978.
- [55] A. Imai, "Examination of parameter of size of automatic fare collection gate," *Omron Technics*, vol.12, no.1, pp.25-40, 1972.
- [56] K. Mori, S. Miyamoto, and H. Ihara, "On evaluation of function and reliability of distributed control system," *Society of Instrument and Control Engineers Trans.*, vol.20, no.4, pp.314-321, 1984.
- [57] K. Kera, K. Bekki, and K. Mori, "Step-by-step system construction technique with assurance technology — Evaluation measure for step-by-step system construction," *22nd International Conference on Distributed Computing Systems Workshops*, pp.101-106, Vienna, Austria, July 2002.
- [58] M. Matsumoto and K. Mori, "Assurance evaluation technology for an autonomous decentralized ATC system," *IEICE Trans. Inf. & Syst. (Japanese Edition)*, vol.J86-D-I, no.1, pp.14-22, Jan. 2003.



Kinji Mori received 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 in System Development Lab.; Hitachi, Ltd. In 1997 he joined Tokyo Institute of Technology; Tokyo, Japan as a professor. His research interests include distributed computing, fault tolerant computing, and mobile agent. He proposed Autonomous Decentralized Systems (ADS) in 1977 and since then he has been involved in research and development of ADS. He is a Fellow of IEEE and a member of IEEEJ, IPSJ and SICE, Japan.



Akio Shiibashi graduated from Saitama University and joined Japanese National Railways (JNR) in 1976. He was transferred to JR East, one of the successors of JNR which was divided and privatized in 1987. He has worked for research and development of IC card ticket system since 1994. He received Ph.D. from Tokyo Institute of Technology.