Agent-Based Active Information Resource and Its Applications

Tetsuo Kinoshita

Tohoku University, Katahira-2-1-1, Sendai 980-8577, Japan
kino@riec.tohoku.ac.jp

Abstract. A scheme of active information resource (AIR) provides a novel approach to actively support using the distributed information resources over the networked environment. With the AIR, passive information resources are extended into autonomous and active entities. The extended information resources can actively perform tasks to support use of them, so that the burden for users can be reduced. Moreover, multiple extended information resources can organize in a decentralized way to autonomously cooperate with each other, and this will enable a more flexible support for using them in the distributed environment. An agent-oriented design model for AIR is developed, and the agent-based AIR is applied and verified in two tasks of support for using distributed academic information resources and managing the operations of network system. The results of experiments on two prototype systems verified that the proposed approach has better performance than conventional approaches do.

Keywords: Active Information Resource, Information Retrieval, Network Management, Agent, Agent-based System.

1 Active Information Resource: Basic Concept

With the explosion of the amount of information available in digital form, as well as the rapid growth of networked environment, particularly the Internet, numerous electronic materials have been made available from various information sources and these materials are continuously generated, accumulated and updated as valuable information/knowledge sources in the distributed environment. Some characteristics of such information resources include: well organized metadata is available, meaningful relations between information resources exist, researchers' accumulated knowledge about information resources can be reused, etc. Due to the large amount of electronic information resources, the complex relations among them, and the distributed nature of them, it is becoming increasingly difficult for users to efficiently and effectively use these information resources. Therefore, an effective support for using information resources is required. Such a support approach should effectively use the characteristics of various information resources to provide mechanisms and functions such as managing, searching, showing and sharing. Furthermore, it should be adaptive in the distributed environment.

S. Kikuchi, S. Sachdeva, and S. Bhalla (Eds.): DNIS 2010, LNCS 5999, pp. 143–156, 2010.

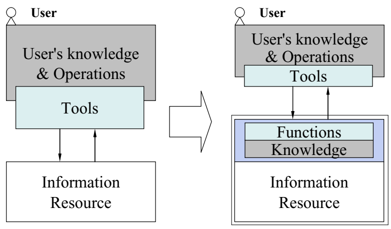


Fig. 1. Extending an Information Resource to Active Information Resource (AIR)

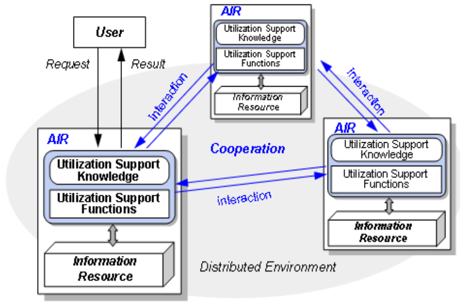


Fig. 2. Cooperation of AIRs

There are many conventional technologies can be applied to support the use of information resources. Typically, information resources in the conventional support system are modeled as passive entities, which can be accessed and manipulated by some functions required by the tasks of support [3]. In such circumstances, there are usually gaps among the characteristics of information resources, the manipulation of these resources, and the support for using them. Due to the gap, the characteristics are not effectively used, and the support is less effective in the distributed environment.

To realize an active support for using various information resources in the distributed environment, this research proposes a novel approach [1,2], as depicted in Fig.1. With this approach, passive information resources are extended into autonomous and active entities. Pertinent knowledge and functions are combined with information resources, and the characteristics of them can be effectively used. The extended information resources can actively perform tasks to support use of them, so that the burden for users can be reduced. Moreover, multiple extended information resources can organize in a decentralized way to autonomously cooperate with each other, and this will enable a more flexible support for using them in the distributed environment as shown in Fig.2. The extended information resource is called Active Information Resource (AIR). In order to design and implement the AIR over the networked environment, a design method for extending information resources into autonomous and active entities and a method how to use the realized AIRs to enable the active support are developed, aiming several application tasks such as searching/sharing useful academic information resources, managing the operation conditions of network system, using web services and so on.

2 Agent-Based Design of Active Information Resource

An AIR is an information resource that is enhanced and extended with information resource specific knowledge and functions for actively and flexibly facilitating use/reuse of it. An agent-oriented design method for AIR is developed and this design method consists of the basic model of AIR, and two agent-based design models of AIR.

Basic Model of AIR: There are 6 main parts in the proposed design model of AIR, which is shown in Fig.3: *Information Resource*, *Domain Knowledge-base*, *Knowledge about Information Contents*, *Information Extraction Unit*, *Information Processing Unit*, and *Contact & Cooperation Unit*. With the interaction between these 6 parts, an AIR can autonomously perform its tasks [5].

Agent-Based Design Models of AIR: When an AIR is designed with the agent-oriented approach, the knowledge and functions of AIR should be mapped into knowledge and functions of an agent or multiple agents. Thus an AIR can be designed and implemented using one of the following design models:

- *Single-agent-based Design Model*: In this model, the knowledge and functions of an AIR is mapped into knowledge and functions of one agent. Generally, the amount of information is relatively small. Since the structure of such an AIR is relatively simple, its design, implementation and maintenance would be accordingly easy.
- *Multiagent-based Design Model*: For an AIR with fairly large quantity of information, complex functionality of information processing and corresponding knowledge about information use/reuse, it should be more suitable to deploy multiple agents. In this model, each agent is in charge of only a certain part of the AIR's functionality and corresponding knowledge. In such an AIR, the knowledge and ability of a certain agent could be relatively simple, which means the development and maintenance of each agent could be easy.

In the design and implementation of agent-based AIR, we use the Repository-based Multiagent Framework (ADIPS/DASH framework) [13] and the Interactive Design Support Environment for agent system (IDEA) [14,15].

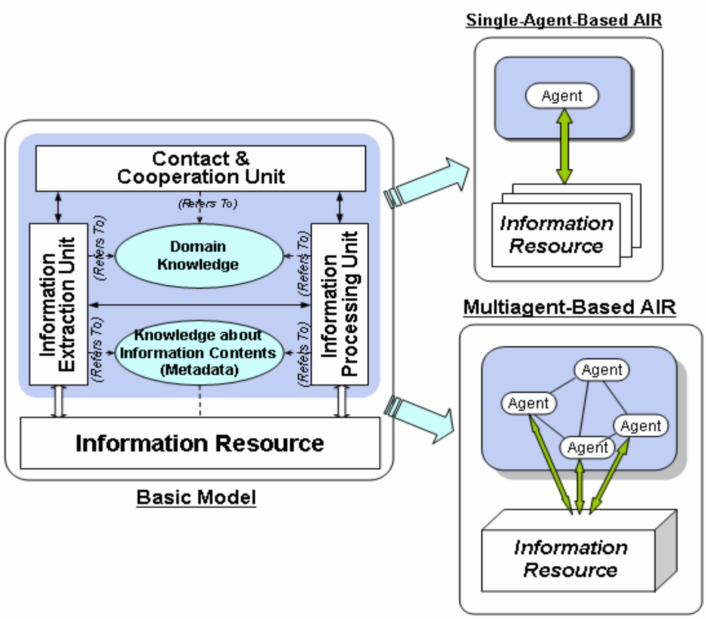


Fig. 3. Design Models of AIR

3 Knowledge Enhanced Search for Academic Information

3.1 AIR for Academic Information Resource

When a personal academic information collection is searched for a researcher's personal use, not only the literal contents, but also the characteristics of them should be considered [4]. However, when a conventional search method is used to find out such a collection, usually the relations among the academic information resources are not properly considered, and the researcher's personal knowledge about the academic information resources are often neglected. Consequently, the recall and efficiency of search could be low [6,7].

To deal with such problem, two types of knowledge about these information resources are introduced and utilized to enhance the search process based on the AIR:

- $\mathbf{K_R}$: Knowledge on Relations among the personally collected academic information resources
- $\mathbf{K_U}$: User's Knowledge about the personally collected academic information resources

The single-agent-based design model of AIR is used to realize the proposed knowledge enhanced search. Each piece of personally collected academic information resource is extended into an AIR called PerC-AIR (Personally Collected academic information resource AIR) [7].

3.2 Design of PerC-AIR

The PerC-AIRs are designed to automatically discover and update their K_R through cooperation. At the meantime, K_U can be automatically maintained by each PerC-AIR which keeps track of its user's operation on a piece of academic information resource.

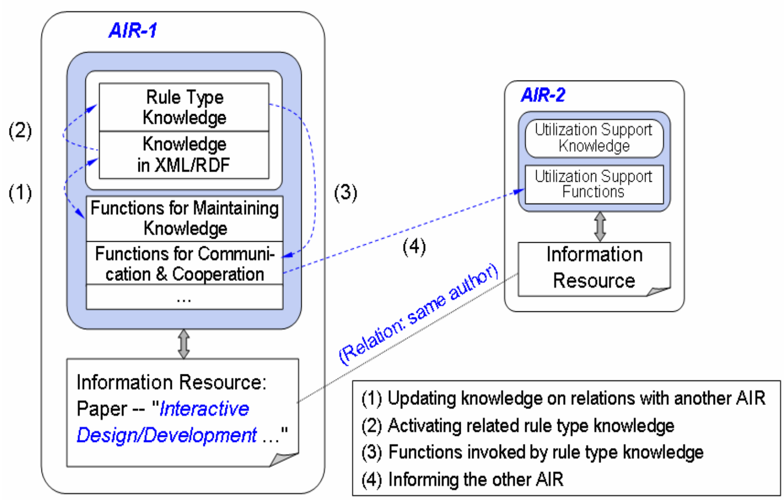


Fig. 4. Architecture of PerC-AIR

Moreover, a knowledge enhanced search in distributed PerC-AIRs is performed through autonomous cooperation between them. The knowledge and functions of PerC-AIR are designed and implemented by using the XML/RDF format representation and the rule-based agent behavior knowledge representation of ADIPS/DASH framework.

Basic knowledge about the information contents

```
...
<!--  AIR-K_B  -->
<dc:type>paper</dc:type>
<dc:title>Interactive Design/Development Methodology of Agent System</dc:title>
<dc:subject>
  interactive, repository-based agent framework, design/development methodology
</dc:subject>
...
```

Knowledge on relations with other AIRs

```
<!--  AIR-K_R  -->
<air:relatedAIRs>
  <rdf:Seq>
    <rdf:_1 dc:identifier="AIR-Paper0005"
            air:relationType="same-author"
            air:relationDescp="T. Kinoshita"/>
    ...
```

User's knowledge about the information contents

```
<!--  AIR-K_U  -->
<air:userComments>
  <rdf:Seq>
    <rdf:_1 air:commentType="user-marked-relevant"
            air:commentDescp="IDEA"/>
    ...
```

```
// Inform related PerC-AIR about the newly found relation.
(rule inform-relation
  (Relation :airID ?airID :informed NO) = ?relation
  (RelatedAIR :airID ?airID) = ?reltdAIR
  (SelfInfo) = ?selfInfo
  (Status) = ?status
  -->
  (send :performative information-about-relation
        :to       ?reltdAIR:airAgentID
        :arrival  ?reltdAIR:airWP
        :content  (Relation
                    :airID          ?selfInfo:airID
                    :airAgentID     ?status:name
                    :airWP          ?status:environment
                    :relationType   ?relation:relationType
                    :relationDescp  ?relation:relationDescp))
  (modify ?relation:informed YES)
)
```

This rule enables an AIR to inform another AIR when some relations between them are found.

Fig. 5. Example of knowledge description of PerC-AIR

3.3 Knowledge Enhanced Search by PerC-AIRs

In a K_R enhanced search, relevance spreads from relevant PerC-AIRs to non-relevant ones, and relevance is accumulated in non-relevant PerC-AIRs. The influence of a relevant PerC-AIR on a related one is determined by the relevance value of the relevant PerC-AIR, the strength of their relation, and the depth of iteration. On the other hand, in a K_U enhanced search, the researcher's evaluation or opinion on each item in the collection is reused. Relevant or non-relevant items can be efficiently determined when there is proper K_U of PerC-AIR indicating so.

Fig.6 depicts a process of the combination of K_R and K_U enhanced search, in which basically the protocol and algorithm for the K_R enhanced search are used, except that K_U is used in each step of the K_R enhanced search. The proposed method finds more relevant PerC-AIRs than the conventional search method.

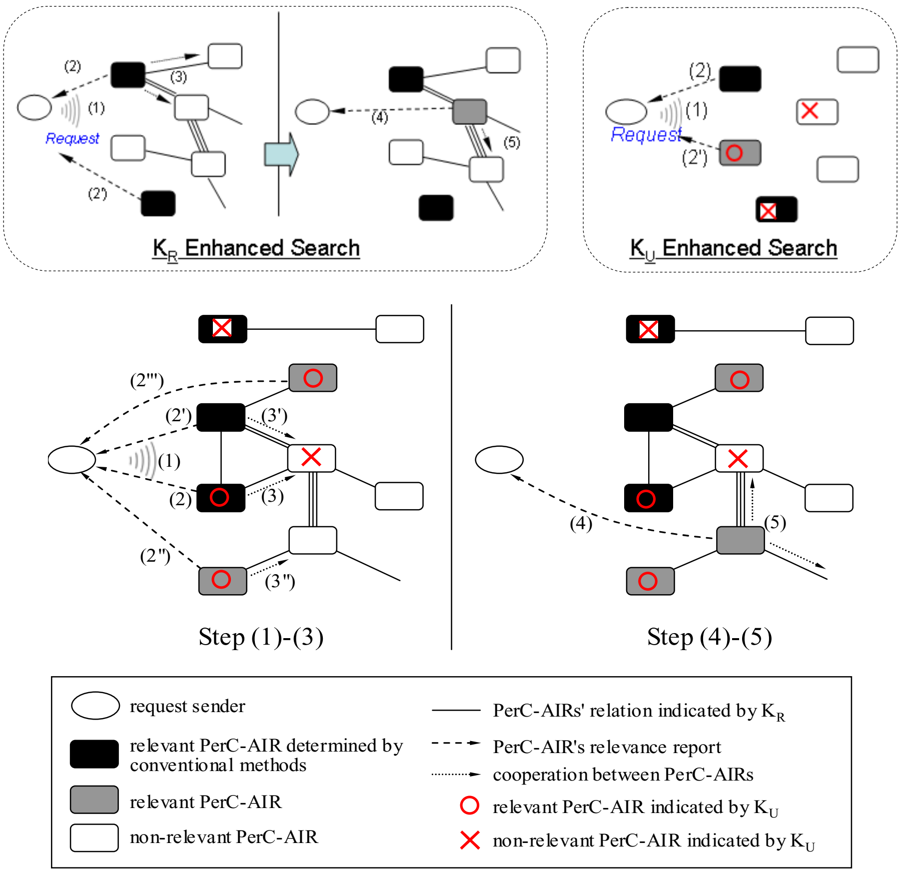
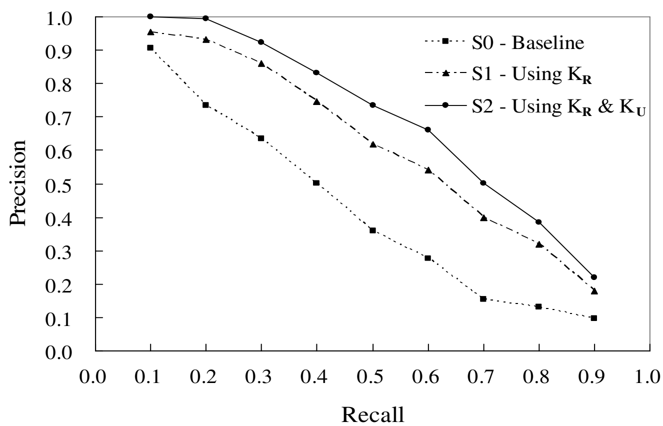


Fig. 6. Combining K_R and K_U Enhanced Search

To verify the proposed search method, some experiments were performed on a prototype system which consists of 110 distributed PerC-AIRs. As demonstrated by the average results of multiple searches shown in Fig.7, the proposed search method has better performance compared to the conventional method. Particularly, the use of a combination of both K_R and K_U can lead to a better performance and a higher efficiency. It can be concluded that the proposed knowledge enhanced search method can be used to search personal academic information collections effectively.

Since the information resources in a researcher's personal academic information collection are potentially valuable to some other researchers, making use of shared personal academic information collections could be an efficient way for researchers to obtain academic information resources [4)]. However, in a conventional mechanism for sharing such collections, there is usually a lack of efficient functions for discovery of valuable collections, as well as an effective search method for them.

Using the AIR-based academic information collection, the Active Mechanism for Sharing Distributed Personal Academic Information Collections can also be realized based on the following two main functions: (f1) Autonomous Discovery of Valuable Collections, and (f2) Knowledge Enhanced Search in Shared Collections, in which the Col-AIRs (Collection AIRs) are introduced to automatically gather information about the collection from all PerC-AIRs in the collection, and the proposed mechanism can support academic researchers to efficiently share the academic information resources in the distributed environment.



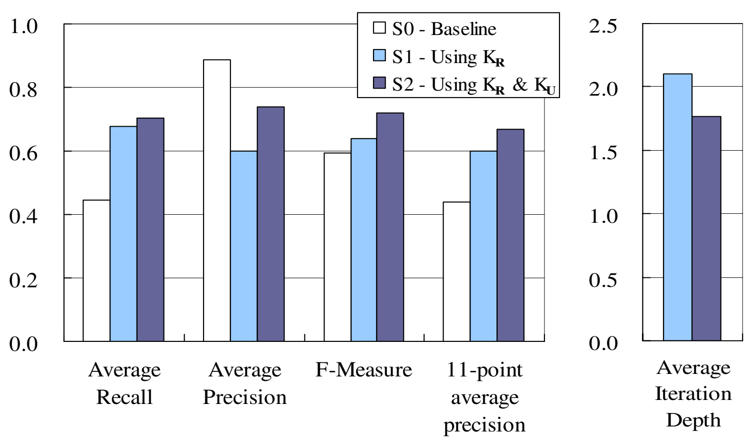


Fig. 7. Performance of Knowledge Enhanced Search

4 Autonomous Monitoring of Network Management Information

4.1 AIR-Based Network Management System (AIR-NMS)

The management activities performed by the network managers/administrators are becoming more demanding and data-intensive because of the rapid growth of modern networks, and automation of network management activities has become necessary. A typical approach to network management is centralized, static, polling-based management that involves high-capacity computing resources at the centralized platform including commercially available management tools [8]. However, in view of the dynamic nature of evolving networks, future network management solutions need to be flexible, adaptable, and intelligent without increasing the burden on network resources. The rapid of network systems has posed the issues of flexibility, scalability, and interoperability for the centralized paradigm. Even though failures in large communication networks are unavoidable, quick detection and identification of the causes of failure can fortify these systems, making them more robust, with more reliable operations, thereby ultimately increasing the level of confidence in the services they provide [9].

Motivated by these considerations, the AIR-based network management system (AIR-NMS) is intended to provide an intelligent, adaptive and autonomous network management support paradigm for various network systems [10,11].

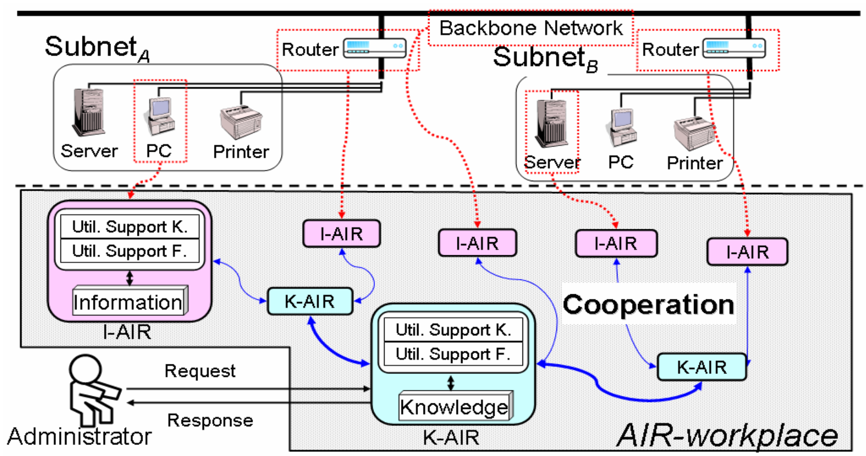


Fig. 8. Configuration of AIR-NMS

The AIR-NMS consists of two types of AIRs; I-AIR (AIR with status information of network) and K-AIR (AIR with knowledge of network management task). I-AIRs manage the status information, which is classifiable into two types: static information and dynamic information. For instance, the relationship between IP addresses and Mac addresses, host names, domain names, IP-routing, etc., are included as static network information, and the dynamic information includes number of packet traffic, RMON-MIB, SNMPv2-MIB, logs of network services, and so on. I-AIRs are responsible to monitor the operational conditions of network, detect the important conditions to be alarmed, inspect/notify the conditions in response to requests of administrator.

On the other hand, K-AIRs manage heuristics or expertise of expert administrators which can utilized as the generic knowledge of network management tasks. K-AIRs and I-AIRs interact with each other to deal with the given/detected problem of the management task. This paper focus on the network monitoring task of AIR-NMS based on capabilities of I-AIRs [12].

4.2 Design of Agent-Based I-AIR

Conventionally, administrators collect status information through periodical polling, aggregate them, and decide the operational conditions of the network system using his expertise/heuristics. The administrator's task can be disaggregated into three sub-tasks, such as detection, recognition, and specification of the failure (abnormal status). In each sub-task, much experience as a network manager are required, therefore, a beginner cannot be employed as an administrator. The I-AIR is introduced to partially support such empirical tasks of administrator; the distributed and effective monitoring of network system, detection of network failure, processing of collected information according to failure, improvement of reliability of detection, recognition, and specification of failure through cooperation among AIRs.

In I-AIR, two information resource types, plain-text format and RDF/XML format, are utilized to represent and manage the status information. For instance, the log-information is acquired through the Syslog (a standard logging solution on UNIX and Linux systems) in plain-text format and the I-AIR extracts a diverse type of log-information and converts it to RDF/XML format specifications.

On the other hand, I-AIRs hold knowledge about information resources together with the functionality to handle collected information. Essential components of knowledge represented in an I-AIR are as follows:

- *I-AIR Identification Knowledge (ID)*: The ID includes an identification number, task number of I-AIR, etc.
- *Knowledge about Information Resource (IR)*: The IR includes a type, an update-time, a format type, etc.
- *Knowledge about Failure Inspection (FI)*: The FI includes two types of knowledge to inspect the failure: text information to be detected in logs, and a threshold of packets, etc.
- *Knowledge about Periodic Investigation Process Control Method (CM)*: The CM includes the polling time and other conditions for updating of the information resource.
- *Knowledge about Cooperation Protocol (CP)*: The CP includes protocol sequences for cooperation with other AIRs.

The knowledge contained in an I-AIR as ID, IR, and CP is required mainly to operate on the information resource and facilitate communication and cooperation among

I-AIRs. The preeminent characteristic of I-AIR is its autonomous monitoring mechanism, which is supported via FI and CM for the inspection and investigation of obstacles that hinder the normal network operation. Table 1 illustrates the I-AIRs developed in the prototype system and Fig.5 shows an example of describing the knowledge of I-AIR (No.15) based on Object-Attribute-Value description format of ADIPS/DASH agent.

Table 1. Example of implemented I-AIRs

I-AIR No.	Function	I-AIR No.	Function
1	Network Disconnection detector	11	DNS server process checker
2	NIC configuration failure detector	12	SMTP server process checker
3	SPAM mail detector	13	POP server process checker
4	MSBlaster attack detector	14	DNS connection checker
5	Mail send/receive error detector	15	Network route to host checker
6	TCP/IP stack failure checker	16	Kernel information checker
7	NIC configuration failure checker	17	Lease IP address checker
8	HUB failure checker	18	Mail server error checker
9	Router failure checker	19	Number of SPAM mail
10	Communication failure checker		

```
I-AIR-No.15
(ID  :air id          "i-air@w1:pcB1.example.com"
     :workplace id    "w1:pcB1.example.com"
     :task id         "0123456789"  )

(IR  :info type       " ping_result"
     :path            " stdout"
     :format type     "text"
     :time            "2007/01/01/00:11:22"  )

(FI  :failure name    "NIC problem"
     :check name      "Ping_NIC_pcB "
     :check string    ("100% packet loss"
                       "Destination net unreachable"
                      "TTL expired in transit"
                       "Ping request could not find"
                      "unknown host")
     :check info       (CI :exist "yes")  )

(CM  :method name     "ping"
     :arguments       ("-c"
                        "4"
                        "172.17.1.2")
     :trigger info    (TI :interval "60000")  )

(CP  :protocol        "Inform - failure Protocol"
                       "Report Protocol"  )
```

Fig. 9. Example of I-AIR (No.15)

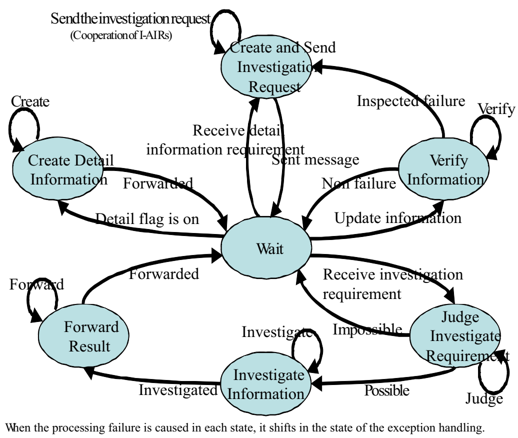


Fig. 10. State transition diagram of I-AIR

Moreover, applying the Single-Agent-based Design Model of AIR, the functionality of I-AIR is designed and implemented based on the state-transition depicted in Fig.10.

4.3 Experiment of Failure Detection by I-AIR

To evaluate the capabilities of I-AIR, an experimental AIR-NMS is set up as shown in Fig.11. The network system comprises a 100BASE-TX Ethernet with a firewall configured as a Network Address Translation (NAT) firewall, a router, and various personal computers (PCs) arranged in four subnetworks. Subnetwork A is configured as a Demilitarized Zone range 172.16.0.0/24. The server (sevA1) DNS and Mail application settings are configured. The other three subnetworks (B, C, D) have IP-addresses in the order given as 172.17.1.0/24, 172.17.2.0/24, and 172.17.3.0/24.

Moreover, the network management console for managing the whole setup resides in pcB1 of subnetwork B. In subnetwork C, there is a desktop-type PC system (pcC1) with a fixed IP address from the DNS server, and a notebook computer (pine) which acquires the IP-addresses through the DHCP. Each node (PC, router, firewall etc.) shows the corresponding AIR workplace where the I-AIRs operate actively. For each node, about 15 AIRs were implemented. This implies that nearly 140 I-AIRs were incorporated within the experimental setup. A Linux operating system was used in each PC.

The network administrator performs the management task according to the conventional manual method, as well as with the I-AIRs based proposed system.

He also measures the performance of the proposed approach adopted for the automation of network functions. In the experiment, the time and the number of procedures executed to correct the obstacle were measured after a network obstacle was reported to a subject. In this paper, the results of detecting specific-failure for multiple causes are demonstrated in below.

In the experiment, two kind of experimental methods have been designed, and for each method, five persons having expertise of managing computer communication systems have been employed:

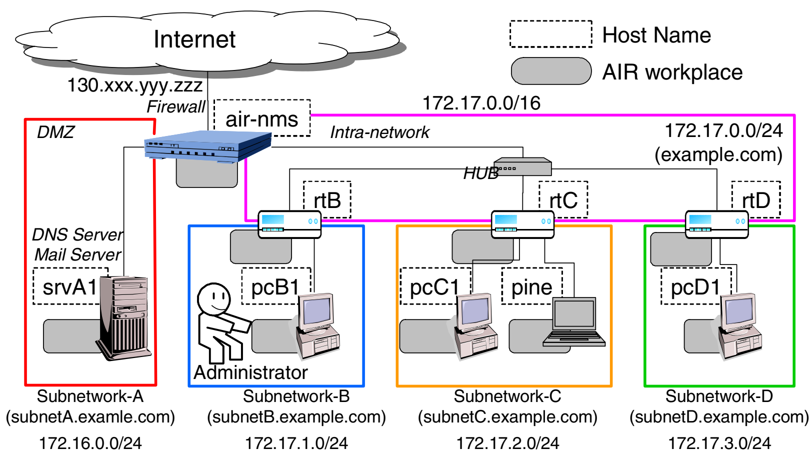


Fig. 11. Construction of experimental AIR-NMS

(i) Monitoring the network with the OS-default network management tools: Several failures obstructing the normal operation of network system are generated and accordingly it is required to restore the network services manually with the client management tools. Also, the time elapsed between the notification of failure to its remedy is measured.

(ii) Monitoring the network utilizing the I-AIRs: The obstacles are detected by the communication / cooperation mechanism of I-AIRs which are then reported to the I-AIR interface, then it is required to rectify the occurring failures. In this case also the time is measured from the point when the obstacle information is presented on the interface to the absolute restoration.

Hence, after some network obstacle has been reported and corrected, the time is measured as well as the number of procedures executed to restore the network to its normal operation.

Table 2. Assumed failure causes: Mail Sending / Receiving Error

Problem	Causes
Cable problem	a. Cable was disconnected.
Port problem	b. The 25th port was closed.
	c. The 110th port was closed.
DNS Server problem	d. DNS Server process was downed.
	e. Configration was not available.
Mail Server problem	f. Mail Server process was downed.

Table 3. Experimental results among individual administrators

	F			G			H			I			J		
		Time	Step		Time	Step		Time	Step		Time	Step		Time	Step
no I-AIR	d	158	9	b	566	8	e	929	23	f	235	5	a	655	19
	e	743	24	d	871	12	b	339	9	c	615	9	f	182	5
I-AIR	a	51	1	f	104	2	c	82	3	a	40	1	b	86	2
	f	85	4	c	106	2	d	52	3	e	74	2	e	128	6
I-AIR / no I-AIR (%)		15.1	15.2		14.6	20.0		10.6	18.8		13.4	21.4		25.6	33.3

Management experience: F. 1year, G. 2year, H. 2year, I. 3year, J. 7year

Table 4. Experimental results among individual failures

	a		b		c		d		e		f	
	Time	Step	Time	Step	Time	Step	Time	Step	Time	Step	Time	Step
no I-AIR	655	19	566	8	615	9	158	9	743	24	235	5
	-	-	339	9	-	-	871	12	929	23	182	5
I-AIR	51	1	86	2	106	2	52	3	74	2	85	4
	40	1	-	-	82	3	-	-	128	6	104	2
I-AIR / no I-AIR (%)	6.9	5.3	19.0	23.5	15.3	27.8	10.1	28.6	12.1	17.0	45.3	60.0

Table 2 depicts the failure situation "Mail Sending / Receiving Error" with some possible causes underlying the occurrence of this anomaly. The task of the subject is to determine the cause of this error. These causes do not occur necessarily in any fixed pattern. The checks to detect these causes are performed randomly. However, using I-AIRs is advantageous because every check is done only once during the course of the fault-localizing process. The failure cause is detected and the main cause behind the failure is reported to the network operator actively.

Experimental results computed by each manager while resolving the mail sending / receiving anomaly were compiled into Table 3. Additionally, the results corresponding to each failure cause were accumulated into Table 4. The results demonstrate that the network management overhead regarding the time taken to resolve a certain fault, along with the number of steps necessary to locate the cause of failure, were reduced to 20% on average.

The foundation of autonomous network monitoring is the use of I-AIRs, which, through active mutual interaction and with the functional network system, can resolve various network-failure situations quite efficiently. A part of I-AIR knowledge is modified dynamically on frequent basis, according to the operational characteristics of the network. The experimental results demonstrated a marked reduction in the administrator workload, through the use of the network monitoring and fault detection functions of I-AIR, as compared to the conventional network management methods.

5 Summary

The concept of active information resource (AIR) is presented and two applications, (i) knowledge enhanced search of academic information collection using PerC-AIRs and (ii) autonomous monitoring of network management information using I-AIRs, are also demonstrated in this paper. The agent-based computing technologies such as the repository-based multiagent framework and the interactive design environment for agent system can successfully be applied in the design and implementation of various AIRs and their agent-based applications.

References

1. Kinoshita, T.: A Method for Utilizing Distributed Information Resources Effectively: Design of Active Information Resource. IEICE Technical Report, AI99-54, pp.13–19 (1999) (in Japanese)
2. Li, B., Abe, T., Sugawara, K., Kinoshita, T.: Active Information Resource: Design Concept and Example. In: Proc. 17th Int. Conf. Advanced Information Networking and Applications (AINA 2003), pp. 274–277 (2003)
3. Chervenak, A., Foster, I., Kesselman, C., Salisbury, C., Tuecke, S.: The data grid: Towards an architecture for the distributed management and analysis of large scientific datasets. Journal of Network and Computer Applications 23(3), 187–200 (2000)
4. Bruce, H.: Personal, Anticipated Information Need. Information Research 10(3) (2005)
5. Bailey, C., Clarke, M.: Managing Knowledge for Personal and Organizational Benefit. Journal of Knowledge Management 5(1), 58–67 (2001)

6. Li, B., Abe, T., Kinoshita, T.: Design of Agent-based Active Information Resource. In: Proc. 1st Int. Conf. Agent-Based Technologies and Systems (ATS 2003), Univ. Calgary, pp. 233–244 (2003)
7. Li, B., Kinoshita, T.: Active Support for Using Academic Information Resource in Distributed Environment. Int. J. Computer Science and Network Security 7(6), 69–73 (2007)
8. Consens, M., Hasan, M.: Supporting network management through declaratively specified data visualizations. In: IEEE/IFIP 3rd International Symposium on Integrated Network Management, pp. 725–738 (1993)
9. Bouloutas, A., Calo, S., Finkel, A.: Alarm correlation and fault identification in communication networks. IEEE Transactions on Communications 42(2,3,4), 523–534 (1994)
10. Konno, S., Iwaya, Y., Abe, T., Kinoshita, T.: Design of Network Management Support System based on Active Information Resource. In: Proc. 18^{th} Int. Conf. Advanced Information Networking and Applications (AINA 2004), vol. 1, pp. 102–106. IEEE, Los Alamitos (2004)
11. Konno, S., Sameera, A., Iwaya, Y., Kinoshita, T.: Effectiveness of Autonomous Network Monitoring Based on Intelligent-Agent-Mediated Status Information. In: Okuno, H.G., Ali, M. (eds.) IEA/AIE 2007. LNCS (LNAI), vol. 4570, pp. 1078–1087. Springer, Heidelberg (2007)
12. Abar, S., Konno, S., Kinoshita, T.: Autonomous Network Monitoring System based on Agent-mediated Network Information. International Journal of Computer Science and Network Security 8(2), 326–333 (2008)
13. Kinoshita, T., Sugawara, K.: ADIPS Framework for Flexible Distributed Systems. In: Ishida, T. (ed.) PRIMA 1998. LNCS (LNAI), vol. 1599, pp. 18–32. Springer, Heidelberg (1999)
14. Uchiya, T., Maemura, T., Xiaolu, L., Kinoshita, T.: Design and Implementation of Interactive Design Environment of Agent System. In: Okuno, H.G., Ali, M. (eds.) IEA/AIE 2007. LNCS (LNAI), vol. 4570, pp. 1088–1097. Springer, Heidelberg (2007)
15. IDEA/DASH Online, `http://www.ka.riec.tohoku.ac.jp/idea/index.html`