

Knowledge-Based Network Management System for Movable and Deployable ICT Resource Unit

Yusuke Tanimura¹, Kazuto Sasai², Gen Kitagata², Tetsuo Kinoshita²

¹Graduate School of Information Sciences, Tohoku University, Sendai, Japan

²Research Institute of Electrical Communication, Tohoku University, Sendai, Japan

Email: tanimura@k.riec.tohoku.ac.jp, kazuto@riec.tohoku.ac.jp, minatsu@riec.tohoku.ac.jp, kino@riec.tohoku.ac.jp

How to cite this paper: Tanimura, Y., Sasai, K., Kitagata, G. and Kinoshita, T. (2017) Knowledge-Based Network Management System for Movable and Deployable ICT Resource Unit. *Journal of Computer and Communications*, 5, 135-151. <https://doi.org/10.4236/jcc.2017.57013>

Received: April 14, 2017

Accepted: May 20, 2017

Published: May 23, 2017

Copyright © 2017 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

When a disaster occurs, the demand for information and communication technology (ICT) services drastically increases. To meet such demands, a national project was undertaken in Japan to develop the Movable and Deployable ICT Resource Unit (MDRU). One challenge regarding the MDRU is securing operators to work the units in emergency situations. As ICT service users have diverse and frequently changing demands, strong technical skills and practical knowledge are required for the administration of MDRUs. In this paper, we propose a knowledge-based network management system to alleviate the burden on administrators. To deal with the structural changes to network systems that frequently occur with changes in ICT service demand, we introduce modularization techniques into our previous research. The proposed system can be easily reconfigured by join/disjoin modules corresponding to changes in the system configuration of the MDRU. The results of our experiments using the implemented experimental system confirm that the proposed system can be applied to MDRU operation and effectively supports administrators.

Keywords

Multi-Agent System, Knowledge-Based Network Management System, MDRU

1. Introduction

In recent years, with the advancement of virtualization technology and a reduction in the price of computing resources, it has become easier to build platforms to provide information and communication technology (ICT) services. Virtualized computing resources can be flexibly arranged in accordance with demand

[1] [2] [3]. Furthermore, with the advent of the concept of network functions virtualization, it is now possible to flexibly construct and reconfigure network infrastructure. With such a growing trend in virtualization technology, a novel method of constructing and operating ICT services has been proposed by a project led by the Ministry of Internal Affairs and Communications in Japan [4] [5]. In this way, ICT services can be quickly restored in disaster situations.

The need to rapidly restore ICT services without waiting for the recovery of affected equipment prompted Sakano *et al.* [4] to propose the Movable and Deployable ICT Resource Unit (MDRU). The MDRU accommodates servers and network equipment, and provides ICT services to users in disaster-affected areas. The MDRU is a portable data center, and can be deployed to disaster areas when a disaster occurs. Various demonstrations have been carried out showing the practical use of the MDRU method, and standardization activities are progressing [6] [7]. The basic ICT services provided by MDRUs include IP phones and message boards. By utilizing virtualized computing resources, MDRUs can flexibly scale ICT services according to demand. In addition, during MDRU operation, changes in demand for ICT services occur frequently because of restoration progress and secondary disasters. Administrators of MDRUs can flexibly deploy ICT services and respond to fluctuating demand by utilizing virtualized computing resources and network functions.

One research topic in the development of MDRUs concerned the burden on administrators. Advanced knowledge and skills are required to properly operate diverse ICT services created and based on highly virtualized computing resources and network functions. Thus, it is difficult to secure an adequate number of expert administrators in emergency situations. Therefore, an intelligent support system should be applied to MDRUs to ensure an efficient operation and reduce the burden on administrators.

Significant research and development have been spent on a network management system (NMS) as a tool to reduce the burden on network and system administrators. NMSs commonly used in actual network management are designed with a focus on the function of monitoring the state of network devices using monitoring protocols [8] [9]. These NMSs can detect abnormal changes in management targets and report to administrators by setting criteria that define normal states. However, certain tasks need to be performed by the administrators themselves, such as identifying detailed fault causes and necessary countermeasures.

From this perspective, various attempts are made to enhance the ability of the NMS and enlarge the scope of automation by applying knowledge derived by expert administrators. In this paper, we call such NMSs, which work autonomously using human expertise, Knowledge-based Network Management Systems (KNMSs). The Autonomous Network Management System (ANMS) is a popular approach to implement KNMSs [10]. ANMS is a concept based on the idea of Autonomous Computing [11], inspired by the autonomous nervous system. NMSs, based on the concept of ANMSs, have the ability to maintain a managed

system's normal state. A key point of an ANMS is a control loop called MAPE-K (Monitor-Analyze-Plan-Execute over a Knowledge) feedback loop. To realize its autonomic features, ANMSs utilize knowledge derived from the administrators and perform the control loop. In addition, various approaches based on artificial intelligence are conducted to apply human knowledge to network operations [12] [13] [14] [15].

To effectively use a KNMS administrators are required to maintain a knowledge base corresponding to changes in the configuration of the managed system. The knowledge base is a repository to accumulate experts' knowledge in the form of rules and policies. Because knowledge unsuitable for system configuration causes the malfunction and degradation of system performance, administrators must pay close attention to the consistency of the knowledge base. Conventional attempts to substantiate KNMSs are intended to be applied to static environments whose configurations do not frequently change, hence the burden of maintaining the knowledge base is not regarded as important. As mentioned above, when operating an MDRU in emergency situations, the system configuration frequently changes to respond to fluctuating demands for ICT services. Therefore, the operation of a MDRU with support from a KNMS is accompanied by a significant burden on those administrators involved in maintaining a knowledge base.

In this paper, we propose a KNMS that can flexibly add or revise knowledge according to changes in the configuration of the MDRU system. We extend our previous research [16] and develop the KNMS, which can be applied to MDRU operation.

The main contributions of this paper can be summarized as follows:

- We modularize a KNMS for each ICT service to easily handle knowledge. We reduced the burden of knowledge management to make it possible to respond in a flexible manner to the changes in the configuration of MDRU systems.
- We introduce functions to improve the operability of the modularized KNMS. The experimental results show that the proposed system can support administrators even in fluctuating environments.

The remainder of this paper is organized as follows. In Section 2, our previous research is introduced, and the improved KNMS is proposed, and can be applied to an MDRU. In Section 3, the design of prototype system is illustrated, and then experiments are demonstrated in Section 4. Section 5 presents our conclusions.

2. A KNMS for MDRU Operation in Emergency Situations

2.1. AIR-NMS

Here, we describe our previous research, the Active Information Resource based Network Management System (AIR-NMS). An Active Information Resource (AIR) is a distributed information resource that is enhanced with the Knowledge of Utilization Support (KUS) and the Function of Utilization Support (FUS), which acts as an autonomous agent [17]. The KUS consists of meta-level know-

ledge, that is, knowledge for handling information resources and cooperation knowledge with other AIRs. The FUS consists of various functions to process its information and communicate with other AIRs. AIRs can cooperate with each other and actively and autonomously process complex information with these utilization supports. By using the concept of AIR, it is possible to deputize tasks necessary for the utilization of information resources. Thus, the burden on users can be reduced.

The AIR-NMS is a network management system based on the concept of AIR [16] [18]. AIR-NMS consists of Information AIRs (I-AIRs) and network management Knowledge AIRs (K-AIRs). An I-AIR manages the network status information acquired from network equipment as its information resource, including IP addresses, application settings, and server logs. A K-AIR manages the network management knowledge that human administrators have gained through experience. Within the concept of AIR-NMS, network information (I-AIR) and management knowledge (K-AIR) cooperate with each other and support management tasks to reduce the burden on administrators.

Figure 1 shows the concept of management support based on an AIR-NMS. As a practical application of AIR-NMS, studies have been conducted that focus on support for fault management. An I-AIR has KUS to monitor the managed object and cooperate with other AIRs, and FUS to retrieve, process, and store status information of the managed object. A K-AIR handles management knowledge about troubleshooting obtained from expert administrators or management manuals. A K-AIR has KUS to cooperate with other AIRs that have related

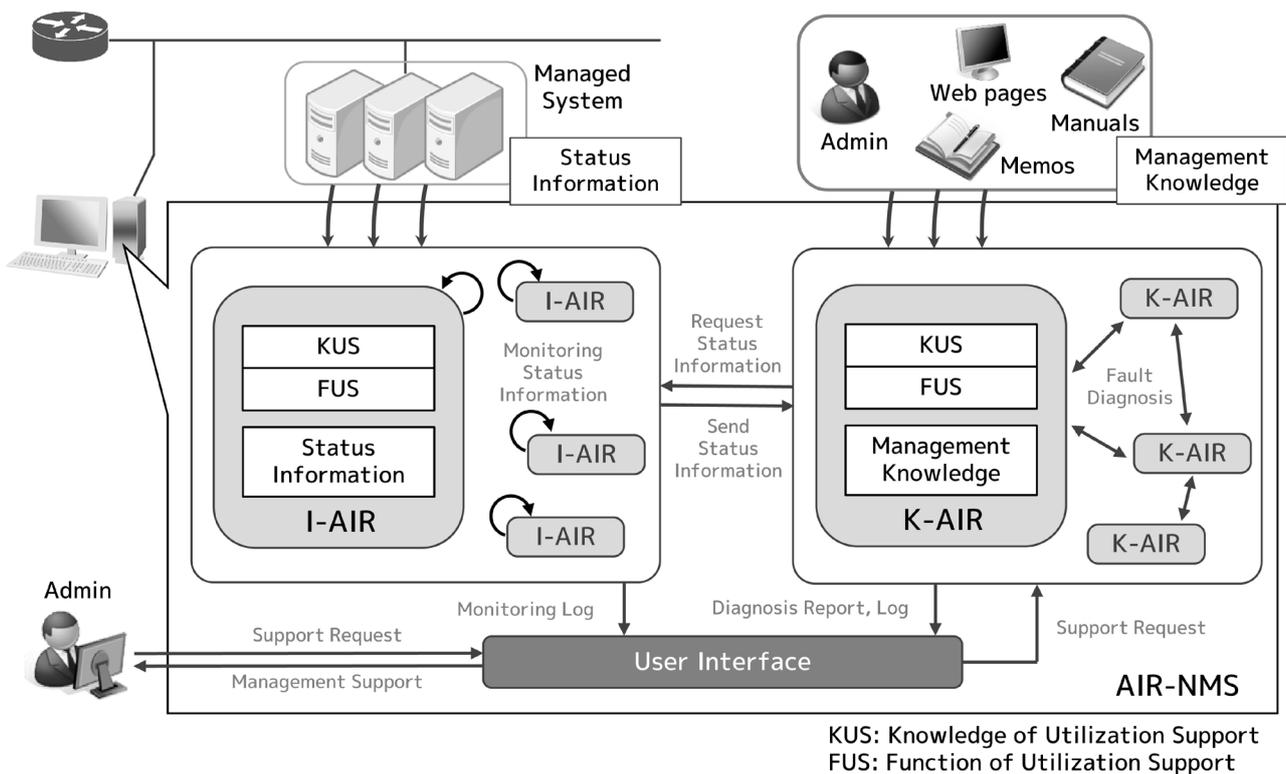


Figure 1. Conceptual model of AIR-NMS applied to fault management tasks.

management knowledge or status information required for fault management, and FUS to extract and analyze the management knowledge described in the text file. When an administrator asks the AIR-NMS to support troubleshooting, K-AIRs start the diagnosis. K-AIRs examine the cause of the fault and generate countermeasures to recover the identified cause. While this process occurs, K-AIRs inquire about the network status information of the I-AIRs, which is necessary for the diagnosis. I-AIRs retrieve the required information from the managed system and reply to the K-AIRs. Finally, K-AIRs identify the cause of the problem and present a countermeasure to the administrator. By executing the proposed countermeasure, the administrator can recover the fault.

2.2. Expansion of Previous Research

Figure 2 shows a conceptual model of AIR-NMS, improved for MDRU operation. In this proposal, we modularize the AIR-NMS to flexibly add or revise knowledge according to changes in the configuration of the MDRU system. We consider that changes in the system configuration of the MDRU mainly occur when a new demand for ICT services emerges. Thus, we model a module for each ICT service. A module contains the K-AIRs and I-AIRs related to the ICT

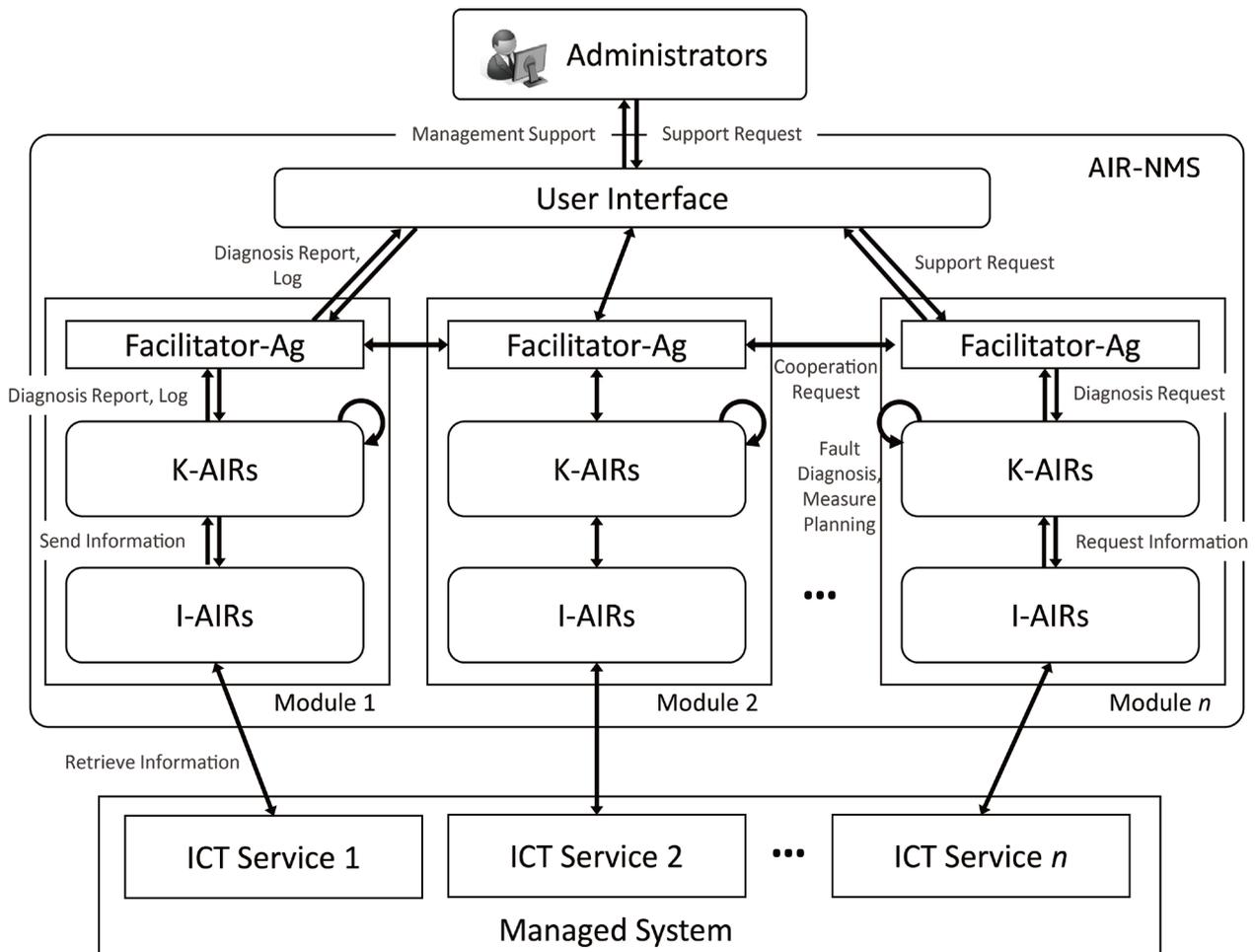


Figure 2. Conceptual model of AIR-NMS improved for MDRU operation.

service. When a change in system configuration occurs, the AIR-NMS follows the change by connecting/disconnecting a module. When an ICT service is added or becomes unnecessary, the AIR-NMS follows the change in system configuration by connecting or disconnecting the module related to the ICT service.

In the proposed system, K-AIRs are divided and managed for each ICT service, and the scope to consider the consistency of knowledge is limited within each module. Thus, it becomes possible to handle knowledge solely focused on an ICT service that is newly added or becomes unnecessary, independent of all other ICT services. Therefore, the burden caused by knowledge management is reduced, and administrators can easily add or revise knowledge to AIR-NMS in the operation of a MDRU where its system configuration frequently changes.

Furthermore, to improve the operability of the module, we introduce two functions: (F1) function to investigate the construction of an ICT service, and (F2) function to verify AIR cooperation between modules. Details of these functions are described as follows.

2.2.1. (F1) Function to Investigate the Construction of an ICT Service

(F1) is a function for I-AIRs to investigate the construction of an ICT service on behalf of human administrators. Information on the construction of an ICT service, namely, information on what kind of servers the ICT service is composed of, is indispensable for fault management. In MDRU operation, it is necessary for administrators to investigate and grasp the construction of the ICT service each time a new ICT service is added. Thus, to reduce the burden on administrators, we reinforce I-AIRs with (F1) as FUS. When a module is connected according to the addition of a new ICT service, I-AIRs in the module autonomously investigate the construction of the ICT service.

To realize this function, we embed experts' heuristics (concerning the investigation of the ICT service construction) to I-AIR as KUS. The following two types of heuristics are given to I-AIRs:

- 1) Knowledge about how to connect to servers: knowledge for using remote access tools such as Telnet and SSH; and
- 2) Knowledge about how to investigate the functions of servers: heuristics about a procedure to investigate the functions of servers, that is, knowledge about using commands to retrieve information such as running processes, the startup settings of background processes, port status, and the configuration of the package manager.

2.2.2. (F2) Function to Verify AIR Cooperation between Modules

In our proposal, we attempt to reduce administrators' burden of knowledge management by modularizing AIR-NMS with each ICT service. However, during actual MDRU operation, the same function can be shared among ICT services. An example of a function that is frequently shared is domain name systems or databases. When a problem occurs in a function shared by several ICT services, these ICT services are affected. In such cases, sufficient fault management support is difficult without sharing management knowledge and status in-

formation among several modules. Therefore, we introduce (F2) to achieve both the benefits of modularization and fault management support with AIR cooperation among modules.

To realize this function, we introduce a new agent to the AIR-NMS, the Facilitator-Agent. As shown in **Figure 2**, Facilitator-Agents are responsible for facilitating and mediating AIR cooperation among modules. When it is necessary for AIRs to cooperate among modules, they cooperate via the Facilitator-Agent in each module. To verify AIR cooperation, the Facilitator-Agent has the following functions:

1) Makes a list of AIRs in a module: a Facilitator-Agent makes a list of K-AIRs and I-AIRs in a module, namely, a list of meta-information about the knowledge and status information held in the module.

2) Examines AIR cooperation among modules: a Facilitator-Agent examines whether a cooperation request from the AIRs of other modules can be processed in its own module. The examination is executed according to the AIR list, and only when judged as acceptable. The cooperation request is forwarded to AIRs in its own module.

In the previous system, as all AIRs work in the same place, cooperation requests were broadcasted unduly. However, in the proposed system, AIRs are modularized and divided for each ICT service module. Moreover, Facilitator-Agents only transfer cooperation requests to modules that can accept requests. Therefore, the number of cooperation requests can be minimized in the proposed system. This means that the burden on AIR management, including knowledge management, can be reduced.

3. Design of the Prototype System

We design a prototype system to evaluate our proposal. The prototype system is designed to implement the repository-based multi-agent framework ADIPS/DASH [19] [20], following previous research [16].

3.1. K-AIR

We design three types of K-AIRs as in previous research [16], Ksc-AIR, Kcd-AIR, and Kcm-AIR. Each type of K-AIR handles the following management knowledge:

1) Ksc (symptom-cause): Cause assuming—assumes the conceivable causes from observed symptoms or detected faults.

2) Kcd (cause-diagnose): Cause diagnosing—diagnoses the exact causes of the faults and presents the diagnosis reports.

3) Kcm (cause-means): Measure planning—plans the countermeasures against the identified causes and presents them.

Figures 3-5 show the description examples of management knowledge. These knowledge examples concern fault management for SIP (IP phone) services and are written in XML.

The Ksc shown in **Figure 3** is the knowledge to derive possible causes described

```

<sc symptom="SIP client cannot make calls">
  <info>SIP-client_ID</info>
  <cause>SIP server process down</cause>
  <cause>SIP connection port not open</cause>
  <cause>Client ID not permitted to call</cause>
</sc>

```

Figure 3. Description example of Ksc.

```

<cd cause="SIP connection port not open">
  <dm>
    <p>get #SIP-port_num# cmd #grep ^port /etc/asterisk/sip.conf
    |awk -F "=" '{print $2}'# login #SIP-server_IP#</p>
    <p>get #SIP-port_state# cmd #grep val(SIP-port_num)
    /etc/sysconfig/iptables |awk -F " " '{print $NF}'#
    login #SIP-server_IP#</p>
    <p>true (#SIP-port_state# -ne ACCEPT)</p>
  </dm>
  <dr>
    Port number #SIP-port_num# is not open at server #SIP-server_IP#.
  </dr>
</cd>

```

Figure 4. Description example of Kcd.

```

<cm cause="SIP connection port not open">
  <m>
    Open UDP port #"^port="value"$"#.
    1. Login server #SIP-server_IP# as superuser.
    2. Open /etc/sysconfig/iptables and add a discription of below.
      -A INPUT -m state --state NEW -m udp -p udp --dport #"^port="value"$"#
    3. Reload iptables with "/etc/init.d/iptables restart".
  </m>
</cm>

```

Figure 5. Description example of Kcm.

within tag <cause> from the observed failure symptom “SIP client cannot make calls”. The Ksc-AIR who has this Ksc sends cooperation requests to Kcd-AIRs for each assumed cause.

The Kcd shown in **Figure 4** is the knowledge to diagnose and verify the cause “SIP connection port not open”. Within <dm>, the diagnostic method is listed as various steps that are to be executed sequentially. If a cause is diagnosed, the Kcd-AIR with this Kcd sends cooperation requests to Kcm-AIRs to obtain a countermeasure for the identified cause.

The Kcm shown in **Figure 5** is the knowledge to produce a countermeasure to the identified cause “SIP connection port not open”. The Kcm is described as a template for making a countermeasure. Kcm-AIRs fill gaps with information and make concrete countermeasure by cooperating with I-AIRs.

3.2. I-AIR

The basic design of I-AIR is inherited from previous research [16]. We mention here the newly added function, (F1) the function to invest the construction of an ICT service.

Regarding knowledge about how to connect to servers, we create knowledge for using SSH.

In addition, we create knowledge on how to investigate the functions of servers. In the prototype system, we also create knowledge for using a command to retrieve a startup setting of background processes. The way to retrieve the startup setting varies depending on the kind of operating system used. Thus, this knowledge includes the knowledge to identify the operating system of managed servers.

3.3. Facilitator-Agent

We describe the design of each function of the Facilitator-Agent introduced as (F2).

We design a protocol for a function to make a list of AIRs in a module to inquire about AIRs in the same module possessing meta-information about knowledge and status information.

To search for the Facilitator-Agent of other modules, Facilitator-Agents use a name server function included in the ADIPS/DASH framework. The Facilitator-Agent periodically looks up other modules and adapts to changes in the configuration of the MDRU. **Figure 6** shows an example of meta-information that can be acquired by this protocol, and the meta-information is associated with the AIR name of the information source. A cooperation request from the Facilitator-Agent of the other module is verified based on this list.

4. Experiments

4.1. Experimental Environment

We implemented a prototype system based on our proposal, an extended AIR-NMS. To implement AIRs, we used the repository-based multi-agent framework ADIPS/DASH [19] [20]. We also implemented a system based on a previous study [16] for evaluation purposes with a comparison of experimental results. We conducted experiments with troubleshooting tasks with two types of AIR-NMS, the proposed system and the previous system. The results were compared and evaluated for the number of manual input necessary for troubleshooting support, the output obtained as a result of troubleshooting support, and the number of AIR/Agent messages during troubleshooting support. In addition, we simulated the addition of knowledge to AIR-NMS, and evaluated the burden of knowledge management. **Figure 7** shows the construction of the managed system. In this experimental environment, the Web service depended on the database (DB) service, and the DB service and the Mail service depended on the Storage service.

The procedure used in this experiment is as follows:

- 1) In an MDRU, Web, DB, Storage and SIP services are in operation.

```
(SIP server process down,Ksc-AIR.201412111553015:w2:AIR-NMS-PC-1)
(SIP connection port not open,Kcd-AIR.201412111553020:w3:AIR-NMS-PC-1)
(Client ID not permitted to call,Kcd-AIR.201412111553023:w3:AIR-NMS-PC-1)
```

Figure 6. Example of meta-information associated with the AIR name.

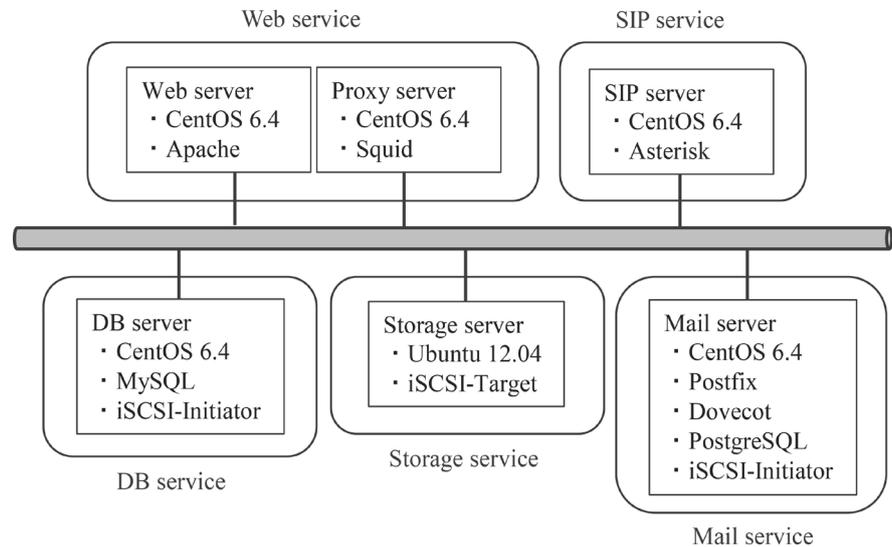


Figure 7. Experimental environment.

- 2) Administrators operate ICT services using AIR-NMS.
- 3) Add Mail service to the MDRU.
- 4) Administrators add knowledge regarding fault management of Mail service to AIR-NMS.
- 5) A fault occurs in a running ICT service.
- 6) Administrators notice the fault via contact from service users.
- 7) Administrators use the AIR-NMS to obtain a solution to the fault.
- 8) Administrators execute the countermeasure proposed by AIR-NMS and confirm whether the ICT service is restored.

Table 1 shows the number of various management knowledge prepared for the experiments. We prepared Kcd and Kcm as one-to-one correspondence. The management knowledge given to the proposed system and the previous system were the same. It must be noted that the consistency of management knowledge for Mail service, which was added during this experiment, was not considered. In the proposed system, management knowledge was modularized for each ICT service. However, in the previous system, all management knowledge was gathered at a single knowledge base.

We conducted experiments on troubleshooting for the following five types of failure causes:

- 1) DB server process down (failure symptom: Web page cannot be browsed);
- 2) HTTP server process down (failure symptom: Web page cannot be browsed);
- 3) Proxy server process down (failure symptom: Web page cannot be browsed);
- 4) SIP server process down (failure symptom: SIP client cannot make calls); and
- 5) POP/IMAP server process down (failure symptom: Email cannot be received).

Table 1. The number of management knowledge for each ICT service.

	# of Management Knowledge		
	Ksc	Kcd	Kcm
Web service	2	11	11
DB service	1	5	5
Storage service	1	4	4
SIP service	1	6	6
Mail service	3	13	13
Total	8	39	39

4.1.1. The Number of Manual Inputs Necessary for Troubleshooting Support

Table 2 shows the number of manual inputs necessary for the execution of troubleshooting support. As described in **Table 2**, in the proposed system the number of manual inputs decreased compared with the previous system. Information that is no longer necessary in the proposed system concerns the construction of ICT services. This is because the proposed system can retrieve such information by using (F1) autonomously.

Even in the proposed system, there are several items that must be input, such as *Web page URL* and *Web client IP address*. However, such information is obtained when the administrators are notified of the service failure by users. Therefore, the acquisition of this information does not represent an extra burden for administrators.

4.2. Execution Results of Troubleshooting Support

Following the input information to the user interface, troubleshooting support was executed for each failure symptom. **Figure 8** and **Figure 9** show the results (failure symptom: *Web page cannot be browsed*, failure cause: *DB server process down*) for the previous system and the proposed system.

4.2.1. Output from the Previous System

Figure 8 shows the previous system output countermeasures for the failure cause *DB server process down*. Specifically, procedures to restart a MySQL database server and a PostgreSQL database server were presented two times each. However, the database used in the Web service was a MySQL database server in the DB service. Therefore, the presented countermeasure, the procedure to restart a PostgreSQL database server, was not appropriate for the actual system.

From an analysis of operation logs of Kcd-AIRs, we found that two Kcd-AIRs that had Kcd to diagnose the cause *DB server process down* were active during troubleshooting support. This showed that collisions of knowledge and unexpected cooperation among AIRs occurred in the previous system, and caused incorrect and redundant outputs. There were two Kcds for *DB server process down* in the single knowledge base of the previous system: one was for a MySQL database server in the DB service and the other was for a PostgreSQL database

Table 2. The number of manual inputs necessary for troubleshooting support.

Symptom	Required Information	Previous Sys.	Proposed Sys.
Web page cannot be browsed	Web server IP address	✓	
	DB server IP address	✓	
	Storage server IP address	✓	
	Web page URL	✓	✓
	Web client IP address	✓	✓
SIP client cannot make calls	SIP server IP address	✓	
	Caller client IP address	✓	✓
	Caller client ID	✓	✓
	Receiver client IP address	✓	✓
	Receiver client ID	✓	✓
Email cannot be received	Mail server IP address	✓	

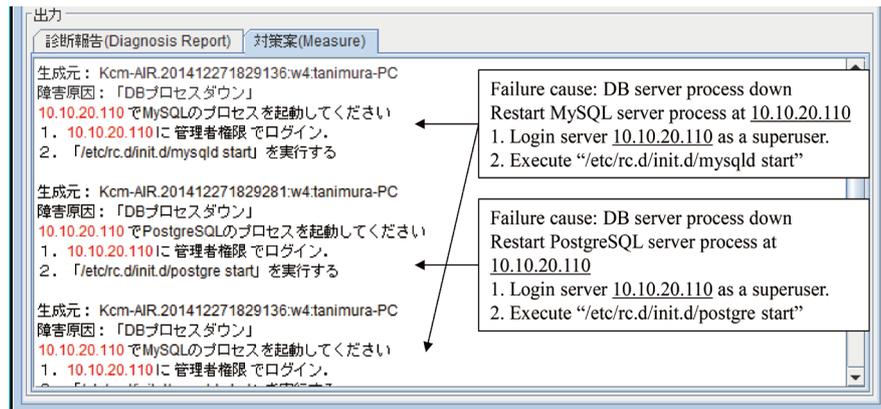


Figure 8. Output countermeasure (previous system).

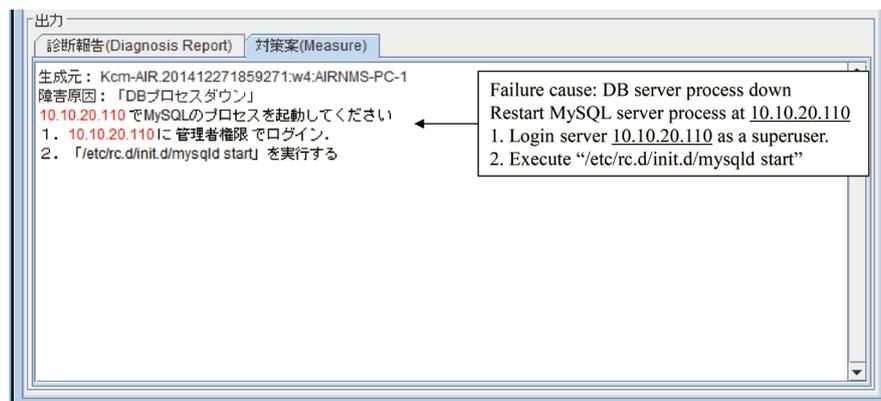


Figure 9. Output countermeasure (proposed system).

server in the Mail service. In this case, administrators must modify the Kcdfs to ensure they do not react to each other. This process makes the knowledge description complex and redundant to avoid collisions of knowledge.

4.2.2. Output from the Proposed System

Figure 9 shows the output from the proposed system, the countermeasure for

the failure cause *DB server process down*. We confirm that the failed Web service was restored by executing the proposed countermeasure. In this experiment, the countermeasure was presented only once, thus, producing a different result than the previous system.

From an analysis of operation logs of AIRs in the Web service and DB service modules, we found the troubleshooting support was conducted with AIR cooperation between the two modules. In detail, a cooperation request generated in the Web service module was taken over by the DB service module, performed by the Facilitator-Agents in each module. Although the K-AIRs in the Web service module did not identify the failure cause in their own module, they cooperated with K-AIRs in the DB service module by using (F2), and identified the correct cause.

In this experiment, the AIRs related to the Mail service were inactive, which is in contrast to the previous system. We considered that a conflict of management knowledge did not occur because of the modularization of AIR-NMS.

4.3. The Number of AIR/Agent Messages during Troubleshooting Support

AIRs and agents in an AIR-NMS based on the ADIPS/DASH framework cooperate with each other by exchanging messages. Thus, we compared the efficiency of AIR/agent cooperation by measuring the number of messages generated during troubleshooting support.

Figure 10 shows the number of AIR/agent messages during troubleshooting support in each experiment. In every experiment, the number of messages in the proposed system was greatly reduced compared with that in the previous system. This result occurred because the proposed system was modularized and the broadcast domain of cooperation requests was restricted in each module.

The AIR-NMS is a multi-agent-based KNMS, hence the management of knowledge consistency in the AIR-NMS is equivalent to directing messages between AIRs/agents.

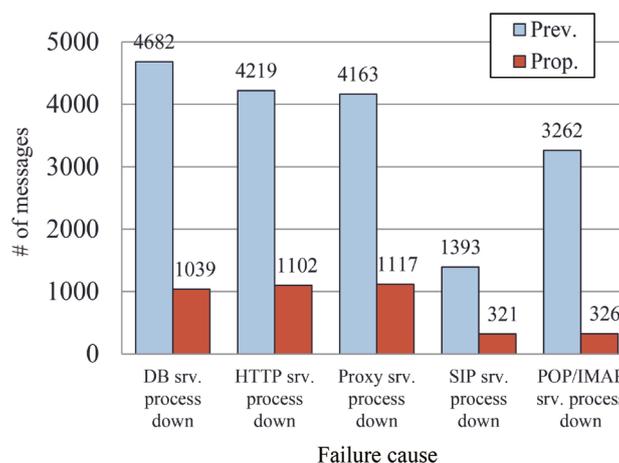


Figure 10. The number of AIR/agent messages during troubleshooting support.

Therefore, in the proposed system, the burden of knowledge management is reduced because the number of messages that must be controlled by administrators is less than that of the previous system.

4.4. Simulation of Knowledge Management

To compare the burden of knowledge management, which occurred when ICT services were added to a MDRU, we conducted simulation experiments. In this simulation, we expressed the amount of the burden on administrators as the number of knowledge units that must have their consistency confirmed by the administrators.

When a k th ($k = 1, 2, \dots$) ICT service is added to an MDRU, the burden on administrators using the previous system N_k or the proposed system N'_k are:

$$N_k = \sum_i^k n_{sc_i} + \sum_i^k n_{cd_i} + \sum_i^k n_{cm_i} \tag{1}$$

$$N'_k = \sum_i^k n_{sc_i} + n_{cd_k} + n_{cm_k} \tag{2}$$

where n_{sc_i} , n_{cd_i} , and n_{cm_i} denote the number of Ksc, Kcd and Kcm, respectively, for the i th ($i = 1, 2, \dots, k$) ICT service.

In the previous system, when new knowledge is added to the knowledge base, administrators must confirm the consistency of Ksc, Kcd, and Kcm between all modules. Thus, the burden on administrators increases accumulatively. However, in the proposed system, although administrators must confirm the consistency of Ksc between all modules considering the sharing of functions between ICT services, the confirmation task for Kcd and Kcm is limited within the added k th module.

Figure 11 shows the estimated burden on administrators, based on experiments using the implemented system mentioned above. We set values of n_{sc_i} , n_{cd_i} , and n_{cm_i} , from the average number of knowledge units used in the experiment (see Table 1), namely, $n_{sc_i} = 2$, $n_{cd_i} = 8$ and $n_{cm_i} = 8$. Then, Equations

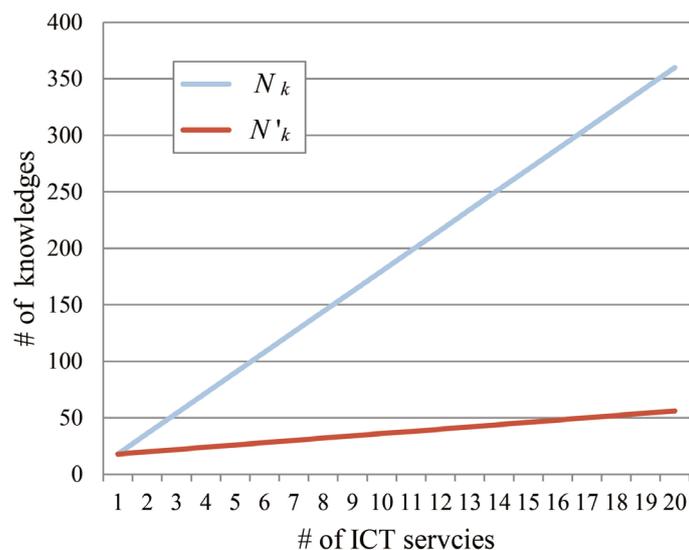


Figure 11. Comparison of burden of knowledge management.

(1) and (2) are rewritten as follows:

$$N_k = 18k \quad (3)$$

$$N'_k = 2k + 16 \quad (4)$$

Figure 11 shows that under the proposed system, the burden on administrators was greatly reduced compared with the previous system, even when the number of ICT services was increased.

4.5. Evaluation

The experimental results show that the burden on administrators, accompanied by the knowledge management of a KNMS, was reduced in the proposed system because of modularization. In addition, we showed that the proposed system was able to conduct troubleshooting support using (F1), without any information about the construction of the ICT service that was to be input by administrators. Furthermore, the issues surrounding the cooperation of AIRs among modules were solved using (F2). Consequently, we consider that our proposal solved the problem of KNMS, which concerns knowledge management with the fluctuating system configuration of a MDRU. Thus, our proposal can be applied to MDRU operation.

5. Conclusions

In this paper, as a means of management support for a MDRU operation, we proposed a flexible KNMS that was able to add or revise knowledge according to changes in the configuration of an MDRU system. We extended our previous research and devised the KNMS, which can be applied to MDRU operation. By modularizing AIR-NMS for each ICT service, we reduced the burden of knowledge management to make it possible to respond in a flexible manner to the changes in the configuration of an MDRU system. The experimental results confirmed the effectiveness of the modularization and that the proposed system has the ability to support MDRU administrators in emergency situations.

In proposed system, administrators easily maintain the knowledge base because the Facilitator-Agent has the function to verify knowledge consistency between modules. However, this does not mean that it becomes completely unnecessary to check dependencies between ICT services. Therefore, as a continuation of this research, we plan to enhance the ability of AIRs to autonomously investigate the dependence between ICT services.

Acknowledgements

This work was partially supported by a Grant-in-Aid for Scientific Research (C) No.16K00292, from Japan Society for the Promotion of Science (JSPS).

References

- [1] Ahmad, R.W., Gani, A., Hamid, S.H.A., Shiraz, M., Yousafzai, A. and Xia, F. (2015) A Survey on Virtual Machine Migration and Server Consolidation Frameworks for

Cloud Data Centers. *Journal of Network and Computer Applications*, **52**, 11-25.

- [2] Dabbagh, M., Hamdaoui, B., Member, S. and Guizani, M. (2015) Energy-Efficient Resource Allocation and Provisioning Framework for Cloud Data Centers. *IEEE Transactions on Network and Service Management*, **12**, 377-391. <https://doi.org/10.1109/TNSM.2015.2436408>
- [3] Beloglazov, A. and Buyya, R. (2013) Managing Overloaded Hosts for Dynamic Consolidation of Virtual Machines in Cloud Data Centers under Quality of Service Constraints. *IEEE Transactions on Parallel and Distributed Systems*, **24**, 1366-1379. <https://doi.org/10.1109/TPDS.2012.240>
- [4] Sakano, T., Fadlullah, Z., Ngo, T., Nishiyama, H., Nakazawa, M., Adachi, F., Kato, N., Takahara, A. and Kurihara, S. (2013) Disaster-Resilient Networking: A New Vision Based on Movable and Deployable Resource Units. *IEEE Network*, **27**, 40-46. <https://doi.org/10.1109/MNET.2013.6574664>
- [5] Movable and Deployable ICT Resource UNIT (2017). <http://www.mdru.org>
- [6] Kobayashi, M. (2014) Experience of Infrastructure Damage Caused by the Great East Japan Earthquake and Countermeasures against Future Disasters. *IEEE Communications Magazine*, **52**, 23-29. <https://doi.org/10.1109/MCOM.2014.6766080>
- [7] ITU-T (2014) ITU-T Focus Group Technical Report: Requirements on the Improvement of Network Resilience and Recovery with Movable and Deployable ICT Resource Units, FG-DR & NRR Version 1.0.
- [8] Nagios (2017) <https://www.nagios.org/>
- [9] Zabbix (2017) <http://www.zabbix.com/>
- [10] Movahedi, Z., Ayari, M., Langar, R. and Pujolle, G. (2012) A Survey of Autonomic Network Architectures and Evaluation Criteria. *IEEE Communications Surveys & Tutorials*, **14**, 464-490. <https://doi.org/10.1109/SURV.2011.042711.00078>
- [11] Kephart, J. and Chess, D. (2003) The Vision of Autonomic Computing. *Computer*, **36**, 41-50. <https://doi.org/10.1109/MC.2003.1160055>
- [12] Ross, M.J., Covo, A.A. and Hart, C.D. (1988) An AI-Based Network Management System. *7th Annual International Phoenix Conference on Computers and Communications*, Scottsdale, 16-18 March 1988, 458-461. <https://doi.org/10.1109/PCCC.1988.10121>
- [13] Cronk, R.N., Callahan, P.H. and Bernstein, L. (1988) Rule-Based Expert Systems for Network Management and Operations: An Introduction. *IEEE Network*, **2**, 7-21. <https://doi.org/10.1109/65.17975>
- [14] Zaman, F., Hogan, G., Der Meer, S., Keeney, J., Robitzsch, S. and Muntean, G.M. (2015) A Recommender System Architecture for Predictive Telecom Network Management. *IEEE Communications Magazine*, **53**, 286-293. <https://doi.org/10.1109/MCOM.2015.7010547>
- [15] Montani, S. and Anglano, C. (2006) Case-Based Reasoning for Autonomous Service Failure Diagnosis and Remediation in Software Systems. *Advances in Case-Based Reasoning*, 489-503. https://doi.org/10.1007/11805816_36
- [16] Sasai, K., Sveholm, J., Kitagata, G. and Kinoshita, T. (2011) A Practical Design and Implementation of Active Information Resource Based Network Management System. *International Journal of Energy, Information and Communications*, **2**, 67-86. http://www.sersc.org/journals/IJEIC/vol2_Is4/5.pdf
- [17] Kinoshita, T. (2010) Agent-Based Active Information Resource and Its Applications. In: Kikuchi, S., Sachdeva, S. and Bhalla, S., Eds., *Databases in Networked Information Systems*, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-12038-1_10

- [18] Tanimura, Y., Sveholm, J., Sasai, K., Kitagata, G. and Kinoshita, T. (2013) A Knowledge-Based Support Method for Autonomous Service Operations after Disasters. *IEEE/ACIS 12th International Conference on Computer and Information Science*, Niigata, 16-20 June 2013, 229-233. https://doi.org/10.1007/978-3-642-12038-1_10
- [19] Uchiya, T., Hara, H., Sugawara, K. and Kinoshita, T. (2011) Repository-Based Multiagent Framework for Developing Agent Systems. In: Wang, Y., Ed., *Transdisciplinary Advancements in Cognitive Mechanisms and Human Information Processing*, IGI Global, Hershey, 60-79.
- [20] IDEA (2017) Interactive Design Environment for Agent System. <http://www.k.riec.tohoku.ac.jp/s/idea/index.html>



Scientific Research Publishing

Submit or recommend next manuscript to SCIRP and we will provide best service for you:

Accepting pre-submission inquiries through Email, Facebook, LinkedIn, Twitter, etc.

A wide selection of journals (inclusive of 9 subjects, more than 200 journals)

Providing 24-hour high-quality service

User-friendly online submission system

Fair and swift peer-review system

Efficient typesetting and proofreading procedure

Display of the result of downloads and visits, as well as the number of cited articles

Maximum dissemination of your research work

Submit your manuscript at: <http://papersubmission.scirp.org/>

Or contact jcc@scirp.org