International Journal of Advanced Smart Convergence Vol. 3 No.1 1-6 (2014)
DOI 10.7236/IJASC.2014.3.1.1

IJASC 14-1-1

A Knowledge-based Network Management System Using Active Information Resources

Tetsuo Kinoshita, Gen Kitagata, Hideyuki Takahashi, Kazuto Sasai, KhamisiKalegele

Research Institute of Electrical Communication
Tohoku University, Japan
{kino, minatsu, hideyuki, kazuto,kalegs}@riec.tohoku.ac.jp

Abstract

An expert network administrator is not always stationed as disasters happen. In that case, it is desirable that a novice administrator is capable of taking part in network recovery operations as well. In this paper, aknowledge-based network management system in emergency situations is presented. We use the Active Information Resource based Network Management System (AIR-NMS) to relieve the human administrator from parts of her management tasks and present an interface that remotely can control this management system. The effectiveness of the system is demonstrated by experiments using a prototype system.

Keywords: *Active Information Resource (AIR), Network Management System, Knowledge-based Autonomous System, Disaster Recovery*

1. Introduction

Network systems have evolved fast and are now both sophisticated and complicated. Therefore, network administrators must have an advanced and broad knowledge in network management in order to operate and maintain their network. At the time of the Great East Japan Earthquake in 2011,network services like IP phone and e-mail were instantly discontinued and network administrators had to repair and restart their networks to get them up running again. However, expert administrators are not always stationed and large and complex networks are likely to have short-handed experts. Hence, it is desirable to make novice administrators also capable of taking part in network recovery operations. An interesting solution to this problem is to implement a network management system (NMS), where intelligent software agents [1] are applied. By automating some management tasks, NMSs can reduce the burden for network management.

Most traditional NMSs are able to gather network status information and detect faults automatically, but identifying the cause of a fault and recover it is one of the most difficult tasks for novice administrators, since they lack the expertise. In order to solve this problem of the traditional NMS, we have proposed an Active Information Resource (AIR) [4] based NMS, called AIR-NMS [7]. The AIR-NMS consists of two types of AIRs, I-AIR and K-AIR, where the former measures status information of various network equipment, and the latter controls network management heuristics of human administrators.

In this paper, we introduce a study on a knowledge based support method for autonomous service operations in emergency situations. A mobile network module called ICT unit, which is placed at a suffering area in an emergency situation and provides network services for users in the area, is introduced in this study. Using the ICT units, the network services of the damaged network are able to recover rapidly. To maintain stable operation of ICT units, an intelligent management function of ICT units takes important role. We

Manuscript received: Feb. 27, 2014 / revised: Apr. 11, 2014
Corresponding Author: kino@riec.tohoku.ac.jp
Tel: +81-22-217-5415, Fax: +81-22-217-5415
Tohoku University, Japan

realize this function based on the AIR-NMS concept to reduce the burden for administrators and to enable even novice administrators to operate complex network services. In Section 2, the concept of the AIR-NMS is introduced. In addition, problems of applying the existing AIR-NMS to ICT units are described. In Section 3, the knowledge-based support scheme using an improved AIR-NMS is explained. The experiments using a prototype system are demonstrated in Section 4. Finally, the conclusion is presented in Section 5.

2. Concept of Active Information Resource-based NMS

2.1 Active Information Resource

An Active Information Resource (AIR) is a distributed information resource [4], extended with support from Knowledge for Utilization Support (KUS) and Functions for Utilization Support (FUS). The KUS contains meta-level knowledge, i.e., knowledge for managing the information resource and knowledge of how to communicate with other AIRs. The active function of the AIR is supported by the FUS, which consists of various functions to process the status of the information resource and for communication between AIRs. The AIRs are specifically designed to be able to handle various distributed information in a network, thus being able to relieve the human administrator of a part of management tasks [5]. The AIRs will cooperate with each other and provide intelligent retrieval of information, actively and autonomously [6].

2.2 AIR-NMS

The AIR-NMS is a network management system based on the concept of AIR and is shown in Figure 1 as a conceptual model. It consists of Information AIRs (I-AIRs) and Knowledge AIRs (K-AIRs) [7]. I-AIRs manage the network status information, such as IP address, host name, application information, logs, etc., for respective equipment in the network. K-AIRs manage the network management knowledge, which is learned through trial and error over the years. When an administrator asks the AIR-NMS to diagnose a network fault, the K-AIRs start the diagnosis. They examine the cause of fault, by inquiring network status information from the I-AIRs and generate counter measures for recovery. By following the proposed steps and execute the presented counter measure, an administrator can then repair the network.

2.3 Problems with the AIR-NMS in Disaster Situations

When it comes to disaster situations, there are in particular two problems with the AIR-NMS in its existing state.At first, an I-AIR has to be installed in the network equipment that is to be monitored. This is time consuming and also requires a certain amount of know-how, since the proper I-AIR to be installed depends on the target's OS distribution and version. Secondly, it is not trivial to prepare an I-AIR for every contingency in advance, especially in a dynamic environment where the network equipment ever so changes.

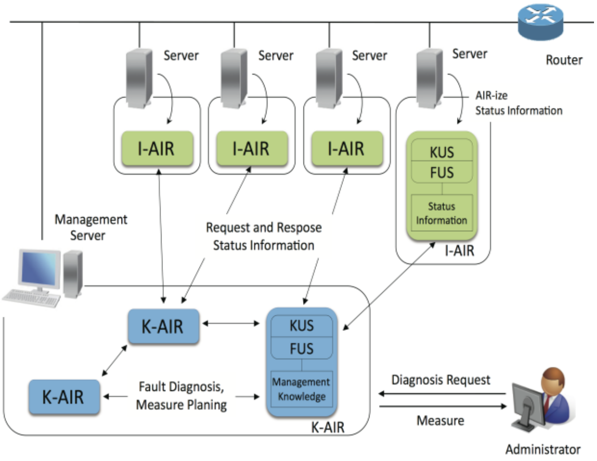


Figure 1. The conceptual model of the AIR-NMS.

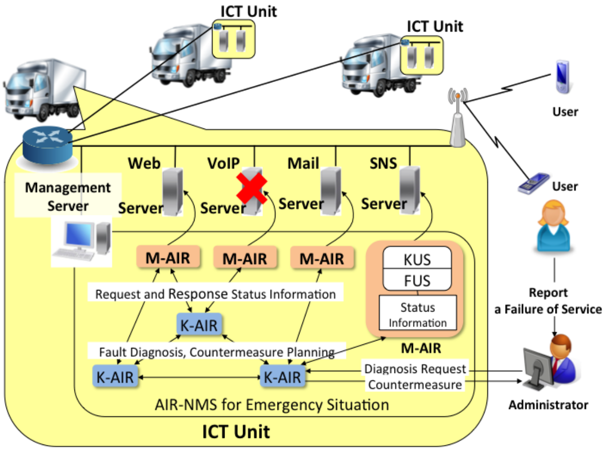


Figure 2. The conceptual model of the AIR-NMS with embedded M-AIRs instead of I-AIRs.

3. Design of a Prototype System

3.1 Extend the AIR/NMS with embedded M-AIRs

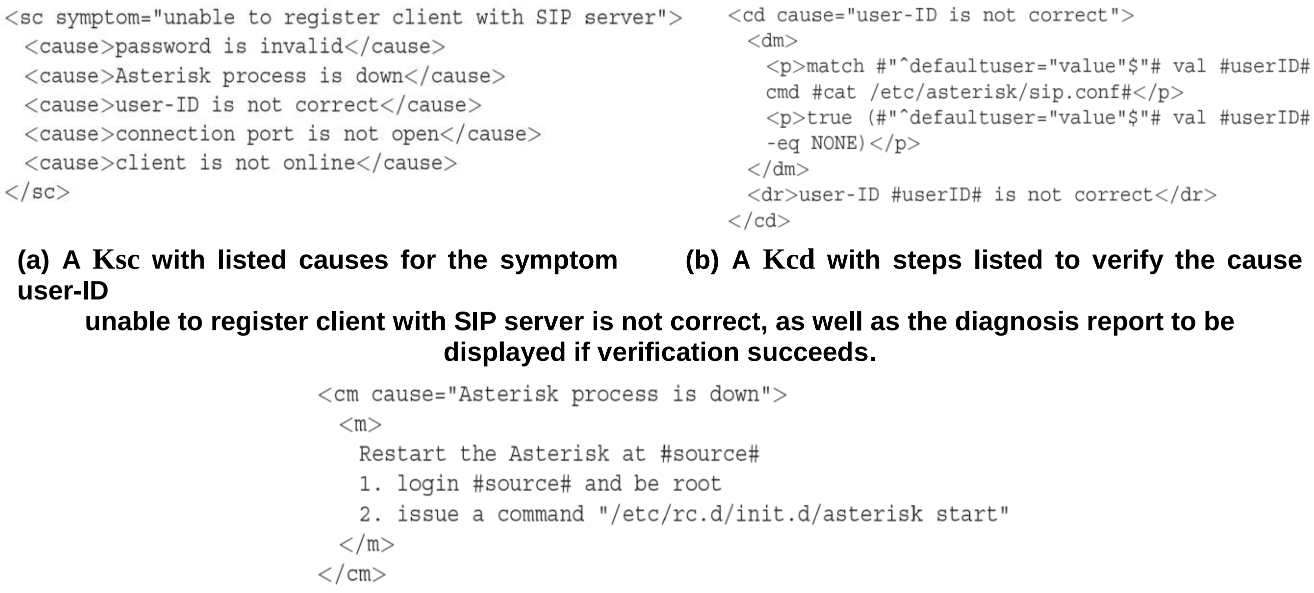
By extending the I-AIR to a Measurement AIR (M-AIR), new functions and knowledge are added to its FUS and KUS. The M-AIR needs fewer requirements and can login remotely to the managed network equipment and retrieve status information from an unknown environment. Figure 2 shows the conceptual model with I-AIRs replaced with M-AIRs.

To save time and the trouble from installing I-AIRs in managed network equipment, they are instead remotely monitored by M-AIRs on the management server. In a similar way to how a human administrator would go about searching for required information in the unknown environment, these human heuristics are implemented as knowledge in the M-AIR's KUS. This knowledge will enable an M-AIR to flexibly explore the unknown environment for status information. If the M-AIR fails to get information initially, an event is fired and caught according to the failure. For example, the M-AIR is looking for a file at a certain path but cannot find it, the rule responding to "No such file or directory" catches the event and instead tries to retrieve the information with a modified method (file search command).

3.2 K-AIRs Composition

In Figure 3, examples of the representation of management knowledge are presented. K-AIRs are classified into three types, Ksc-AIR, Kcd-AIR and Kcm-AIR [8].

- Ksc-AIR: Cause assuming - assumes the conceivable causes from observed symptoms or detected faults.
- Kcd-AIR: Cause diagnosing - diagnoses the exact causes of the faults and presents a diagnosis reports.
- Kcm-AIR: Measure planning - plans the measure against the identified causes and presents them.



```
<sc symptom="unable to register client with SIP server">
  <cause>password is invalid</cause>
  <cause>Asterisk process is down</cause>
  <cause>user-ID is not correct</cause>
  <cause>connection port is not open</cause>
  <cause>client is not online</cause>
</sc>
```

```
<cd cause="user-ID is not correct">
  <dm>
    <p>match #"^defaultuser="value"$"# val #userID#
    cmd #cat /etc/asterisk/sip.conf#</p>
    <p>true (#"^defaultuser="value"$"# val #userID#
    -eq NONE)</p>
  </dm>
  <dr>user-ID #userID# is not correct</dr>
</cd>
```

(a) A Ksc with listed causes for the symptom **(b) A Kcd with steps listed to verify the cause user-ID**

unable to register client with SIP server is not correct, as well as the diagnosis report to be displayed if verification succeeds.

```
<cm cause="Asterisk process is down">
  <m>
    Restart the Asterisk at #source#
    1. login #source# and be root
    2. issue a command "/etc/rc.d/init.d/asterisk start"
  </m>
</cm>
```

(c) A Kcm with the recovery process for the cause Asterisk process is down.

Figure 3. Representation examples of management knowledge

A Ksc-AIR (symptom-cause) deals with management knowledge Ksc of possible causes to an observed symptom. Figure 3(a) shows relevant causes listed for the symptom *unable to register client with SIP server*. The FUS of the Ksc-AIR provides with functions to send each cause as a message to other Ksc-AIRs to further break the cause down into other causes and to Kcd-AIRs where the assumed cause can be handled.

A Kcd-AIR (cause-diagnose) deals with management knowledge Kcd to diagnose and verify a cause assumed by a Ksc-AIR. Figure 3(b) shows the Kcd setup for the cause *user-ID is not correct*. Within tag <dm>, the *diagnose method* to verify the cause is listed in steps, that are to be executed sequentially. If verification succeeds, the Kcm-AIRs are messaged and the administrator is notified with a diagnosis report.

Finally, Kcm-AIR (cause-means) deals with management knowledge Kcm to recommend a counter measure to the detected cause. Figure 3(c) shows the recovery process for the cause *Asterisk process is down*. The FUS of Kcm-AIR provides with functions to plan the counter measure and to present it to the administrator.

4. Experiments

A prototype system of an AIR-NMS with M-AIRs was implemented, with the repository-based multi-agent framework ADIPS/DASH [9]. An administrator was set to operate an IP phone service in emergency, with the experimental network environment shown in Figure 4 displaying two servers controlled by one management server. Server A runs CentOS 6.3 with Asterisk 1.8 for SIP service, Server B runs FreeBSD 9.0-RELEASE with Asterisk 1.8 and the management server runs Windows 7. The K-AIRs included knowledge of how to operate an Asterisk IP phone service on CentOS, but not on FreeBSD. Therefore, Server B is unknown territory for the prototype system.

Two experiments were conducted. The first experiment was to test the M-AIR's function of remote login, by doing fault resolution on Server A. The second experiment was to test the M-AIR's capability of handle an unknown environment, Server B.

4.1 Remote Login Experiment

Figure 4 shows experimental network environment. The experiment procedure is described as follows:

1) The experimenter simulates a failure in the experimental network.
2) A user reports the failure to the administrator (step (a)).
3) The experimenter asks the AIR-NMS to diagnose the failure (step (b)).
4) Kcd-AIRs try to identify the cause of the failure (step (c) and (d)).
5) M-AIRs retrieve status information from the servers via remote login (step (e)).
6) A Kcm-AIR generates a countermeasure and reports it to the administrator (step (f) and (g)).
7) The administrator executes the countermeasure on the actual server (step (h)).
8) The experimenter checks whether the presented identified cause and countermeasure are proper or not.

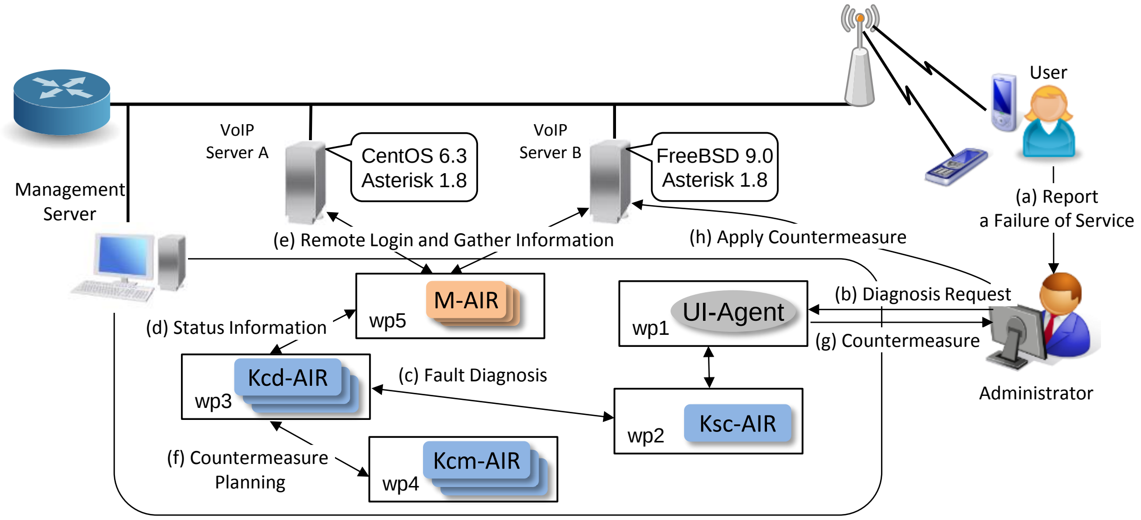


Figure 4. The experimental network environment: Two different servers are controlled by the management server.

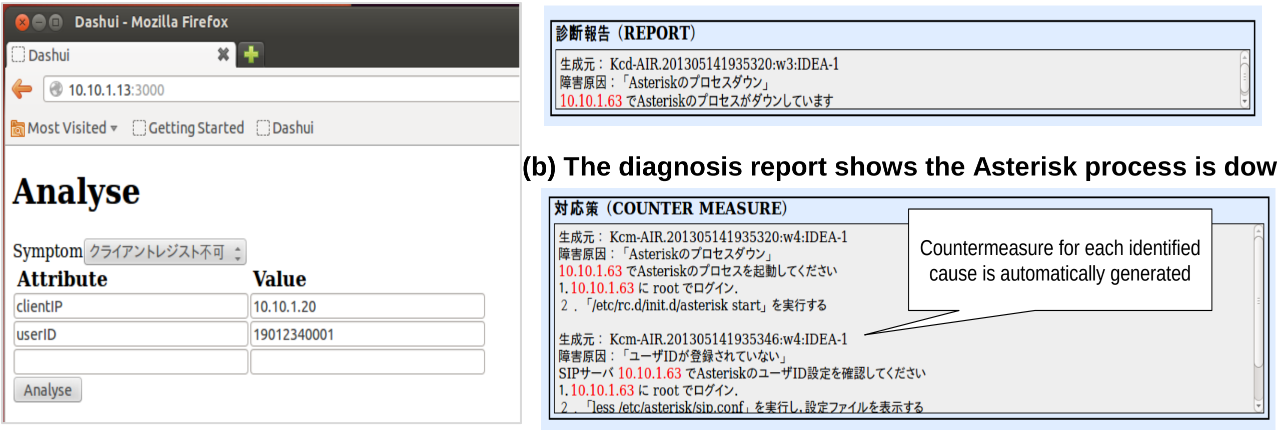
The experimenter started off by killing the SIP process running on Server A, rendering the SIP clients unable to register with the server. To diagnose the error, the experimenter then filled in the blanks of the

AIR-NMS interface with required information, such as symptom name, client's IP and user ID (Figure 5(a)). As a result, the identified cause *Asterisk process is down* and a counter measure were presented (Figure 5(b) and 5(c)), the same as was arranged.

The above result shows that the AIR-NMS is fully functional by using the M-AIR's remote login function, instead of having to install I-AIRs in the managed servers.

4.2 Handling Unknown Environment Experiment

This time, the error is set to occur in Server B. Since the directory structure is different between CentOS and FreeBSD, it is likely that the knowledge the existing AIR-NMS possesses about CentOS is not applicable to FreeBSD. Figure 5 shows the screen shot of the execution log. The M-AIR failed to retrieve status information from Server B, with the event No such file or directory being raised. As a counter measure to find the missing file sip. conf, the M-AIR issued a file-search command. The M-AIR therefore modified the initial command cat /etc/asterisk/sip. conf to cat /usr/local/etc/asterisk/sip. conf and succeeded in acquiring the status information from server B. Thus, M-AIR can acquire information from an unknown environment with the knowledge about a human administrator's way of dealing with a similar situation.



Analyse

Symptom クライアントレジスト不可

Attribute	Value
clientIP	10.10.1.20
userID	19012340001

診断報告（REPORT）

```
生成元： Kcd-AIR.201305141935320:w3:IDEA-1
障害原因：「Asteriskのプロセスダウン」
10.10.1.63 でAsteriskのプロセスがダウンしています
```

(b) The diagnosis report shows the Asterisk process is down.

対応策（COUNTER MEASURE）

```
生成元： Kcm-AIR.201305141935320:w4:IDEA-1
障害原因：「Asteriskのプロセスダウン」
10.10.1.63 でAsteriskのプロセスを起動してください
1. 10.10.1.63 に root でログイン.
２．「/etc/rc.d/init.d/asterisk start」を実行する

生成元： Kcm-AIR.201305141935346:w4:IDEA-1
障害原因：「ユーザIDが登録されていない」
SIPサーバ 10.10.1.63 でAsteriskのユーザID設定を確認してください
1. 10.10.1.63 に root でログイン.
２．「less /etc/asterisk/sip.conf」を実行し，設定ファイルを表示する
```

Countermeasure for each identified cause is automatically generated

(a) Interface of the NMS analysis. The experimenter chooses the symptom (client cannot register) and fills in the IP address and the user ID of the client

(c) Suggested counter measure states: login as root and execute the command /etc/rc.d/init.d/asterisk/start.

Figure 5. Screen shots of the prototype system

5. Conclusion

We have proposed a method which can reduce the burden on administrators and greatly help novice administrators to operate a complex network. This is especially true, when running the network in case of an emergency. We reinforced the AIR-NMS with M-AIRs to be able to flexibly acquire remote status information by remote login like a human administrator, and to be able to restart network services rapidly. The capability of the proposed method was shown through experiments using a prototype system. As future work, we plan extend K-AIR's knowledge for error handling and M-AIR's knowledge for handling unknown environments.

Acknowledgement

A part of this work is supported by supplementary budget for 2012 of Ministry of Internal Affairs and Communications of Japan, Research and Development of Movable ICT Unit enabling Emergent Transport to Disaster Area and Multiply Connected Operation.

References

[1] H. S. Nwana, “Software agents: An overview”, Knowledge Engineering Review, 11(3), pp.205-244, 1996

[2] N. Samaan, A. Karmouchm “Towards Autonomic Network Management: an Analysis of Current and Future Research Directions”, IEEE Communications Surveys & Tutorials, 11(3), pp.22-36, 2009

[3] J. Keeney, D. Lewis and D. O‘Sullivan, “Ontological semantics for distributing contextual knowledge in highly distributed autonomic systems”, Journal of Network and System Management, Special Issue on Autonomic Pervasive and Context-aware Systems, 15(1), pp.75-86, 2007

[4] B. Li, T. Abe, K. Sugawara and T. Kinoshita, “Active Information Resource: Design Concept and Example”, 17th International Conference on Advanced Information Networking and Applications (AINA), pp.274- 277, 2003

[5] S. Konno, Y. Iwaya, T. Abe, T. Kinoshita, “Design of Network Management Support System based on Active Information Resource”, 18th International Conference on Advanced Information Networking and Applications (AINA), pp.102-106, 2004

[6] B. Li and T. Kinoshita, “Active Support for Using Academic Information Resource in Distributed Environment”, International Journal of Computer Science and Network Security, 7(6), pp.69-73, 2007

[7] K. Sasai, J. Sveholm, G. Kitagata and T. Kinoshita, “A Practical Design and Implementation of Active Information Resource based Network Management System”, International Journal of Energy, Information and Communications, 2(4), pp.67-86, 2011

[8] Y. Takahashi, D. Misugi, A. Sakatoku, A. Satoh, A. Takahashi, K. Sasai, G. Kitagata, T. Abe and T. Kinoshita, ”Knowledge Oriented Network Fault Resolution Method Based on Active Information Resource,” IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, pp.361-364, 2010

[9] T. Kinoshita and K. Sugawara, “ADIPS Framework for Flexible Distributed Systems”, Multiagent Platforms, ser. Lecture Notes in Computer Science, T. Ishida, Ed. Springer Berlin/Heidelberg, vol. 1599, pp.18-32, 1999