http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 6 (6): 891-896, 2007
ISSN 1812-5638

Mobile Agent Based Expert System for Distributed Network Management

[1]Neeraj Nehra, [2]Naveen Kumar Gondhi, [3]Durgesh Pant and [4]R.B. Patel
[1,2]School of Computer Science and Engineering, Shri Mata Vaishno Devi University, Katra (J and K), India
[3]Department of Computer Science, Kumaon University, Nainital (Uttranchal), India
[4]Department of Computer Engineering, M.M. Engineering College, Mullana (Ambala), Haryana, India

Abstract: Network Management is a service that employs a variety of tools, applications and devices to assist human network managers in monitoring and maintaining networks. The existing Network Management solutions provide inadequate coverage of functional areas of heterogeneous network leading to degrading level of performance and reliability. However, these key issues should be properly tackled for efficient Network Management. This research proposes an Expert System for creation of an Intelligent Network Management Solution for Distributed Networks using Mobile Agents. In this study, we explore how to take advantage of the Expert System Technology and Mobile Agents to build High Performance Network Management Solution.

Key words: Network management, Mobile agent, Expert system

INTRODUCTION

Existing Network Management solutions focus primarily on fault and performance reporting, which needs to be analyzed by human network administrators to formulate a corrective action for the reported network faults. This process is compounded by the amount of events generated by the Network Management System (NMS) and Simple Network Management Protocol (SNMP) traps received from the various devices in the network, which threaten to exceed the cognitive capabilities of human network administrators/managers. The network administrator has to typically sift through thousands of alarms to figure out the observable symptoms of the network fault before devising a strategy for solving the network fault, sometimes having to solve similar problems multiple times.

Given the large scale of networks being managed today, organizations need to invest heavily in maintaining large IT infrastructure management teams or completely outsource the IT management business process of organizations which specialize in providing such services. The demand for expert network professionals in the IT industry worldwide is such that organizations struggle to recruit qualified and experienced candidates for such positions again having to invest heavily in training and re-training of the recruited candidates before they can effectively carry out their responsibilities.

This research proposes E-DNet, a self-learning expert-system which integrates with a Mobile Agent Platform and a Helpdesk Trouble-Ticketing system to build correlation rules between the reported network faults and the corresponding action initiated by the designated expert network administrator. Over a period of time, the system builds enough rules to be able to initiate corrective actions on its own, thereby alleviating the need to maintain huge IT teams within organizations which is both human and cost-intensive. Moreover, such a system serves as a tutoring tool to the new members of the IT team, by allowing them to learn from the actions taken by the experts in the team, reducing their learning curve and helping them in achieving higher levels of productivity in a relatively smaller time frame.

RELATED WORK

The application of Artificial Intelligence (AI) and expert systems to the domain of Network Management is not a new concept and has been addressed by various researchers (Roth and Waterman, 1983; Davis *et al*., 1977; Gurer *et al*., 1998). More recently there have been several research efforts in the diagnosis of physical systems and networks in particular by Jeffrey and Chess (2003) Marques (1988) and Hartely (1984). Mcauley and Morera (2003) describes a real time interactive expert system and the tasks of fault isolation by formulating plans for the diagnosis process. Andrew *et al*. (2003) came with the concept of Autonomic Computing Correlation for the Fault Management System. Many attempts have been made to automate diagnosis in the networks domain (Laffey *et al*., 1986; Cisco Event Correlation Guidelines, 2006; HP Open View Network Node Manager, 2006). Peacocke and Rabie (1987) describes the Knowledge based Maintenance in Networks. Christopher (1995) tied

Corresponding Author: Neeraj Nehra, School of Computer Science and Engineering, Shri Mata Vaishno Devi University, Katra (J and K), India

the concepts of AI and network monitoring and diagnosis. Tim (1995) Steinder (1994) present fault identification and fault localization techniques in Network Management. The concept of self healing of a system during fault diagnosis has been discussed by Sethi and Steinder (2002a, b) and Doyle and Sastry (1991) describes the applications of knowledge based system for communication Network Management. Monica and Jonathan (1991) and Goodman and Latin (1991) put forth the knowledge-based system for fault localization. Bush and Kalayanaraman (2006) explore the concept of automated knowledge acquisition. Jordaan and Paterok (1993) presented the event correlation guidelines for the heterogeneous networks and Sutter and Paul (1998) outlines the design of expert systems for real-time diagnosis of self-correcting networks. However, most of the research has focused on building intelligence and rules for the localization of the faults (Smyth *et al*., 1991) but very little research has focused on automated corrective action in response to the diagnosed faults.

SYSTEM MODEL/ARCHITECTURE

The Network to be monitored is the collection of network elements such as routers, switches, end-user devices. MAP (Mobile Agent Platform) (Puliafito *et al*., 2000), is a platform for the development and the management of mobile agents that gives all the primitives needed for their creation, execution, communication, migration, etc. It has been entirely developed in Java and this guarantees its total portability on the different hardware and software architectures.

One of purposes in the creation of the platform has been the complete integration and compatibility with the SNMP world. For this reason, in our system we have integrated and used classes which implement the SNMP stack. In this way, we have had the opportunity to interact with the different nodes of the network through standard Management Information Base (MIB) variables. At the same time the developed framework allows us to monitor non-standard quantities (not defined by a MIB) defined by the user. The Fig. 1 depicts the E-DNet schematic Architecture.

We have developed the following basic types of mobile agents for the management:

- Browser Agent
- Daemon Agent
- Messenger Agent
- Verifier Agent

Browser agent: The browser agent collects some MIB variables from a set of nodes specified by the user. After being started, the agent reaches the first node to be visited, opens an SNMP local communication session, builds a pdu-request containing the MIB variables to be searched, waits for the reply from the SNMP daemon and saves the data obtained in an internal structure. Then, if other nodes need to be visited, it reaches them and repeats the procedure mentioned above.

This agent realizes the functionalities of an extended MIB browser. In fact, in an ordinary MIB browser, after activating a connection with a specific node, we can view the contents of the MIB variables, by sending an SNMP request to the node in question for each variable. Conversely, through the browser agent, such requests are first given to the agent, which (by moving from a node to another) inter-acts with them locally and reports the global result to the initial station.

Daemon agent: The daemon agent monitors a health function defined by the user. For starting the agent, the function to be computed and the node of the network (where the computation has to be done) must be provided. Then this agent moves to the node in question, where it records the value of the function: if the value is higher than a fixed threshold (defined by the user), a notification message is sent to the server from which the agent has departed. We can implement (by means of this agent) a mechanism of generalized trap, where the events in which the NMS is notified can be freely and very flexibly set by the user.

Messenger agent: The two agents described before directly interact with the SNMP daemon present in the different nodes (through the Advent classes). Conversely, the messenger agent, during its migration through the nodes of the network, interacts with other agents for collecting specific information produced by them. During the configuration we need to select the agents to be contacted and the servers where they have to be searched and (if necessary) also the number of times the agent has to contact such agents. Thus, the messenger agent performs operations at a higher abstraction level than the mere retrieval of MIB variables. In fact, since daemon agents can perform the computation of any function on the different nodes of the network, the messenger allows us to collect such information, thus obtaining a general description about the state of the network.

Verifier agent: The verifier agent does not perform an actual Network Management action. Its task is that of collecting important information, which might be useful for further operations of Network Management, by MAP. It visits the nodes selected during the configuration and

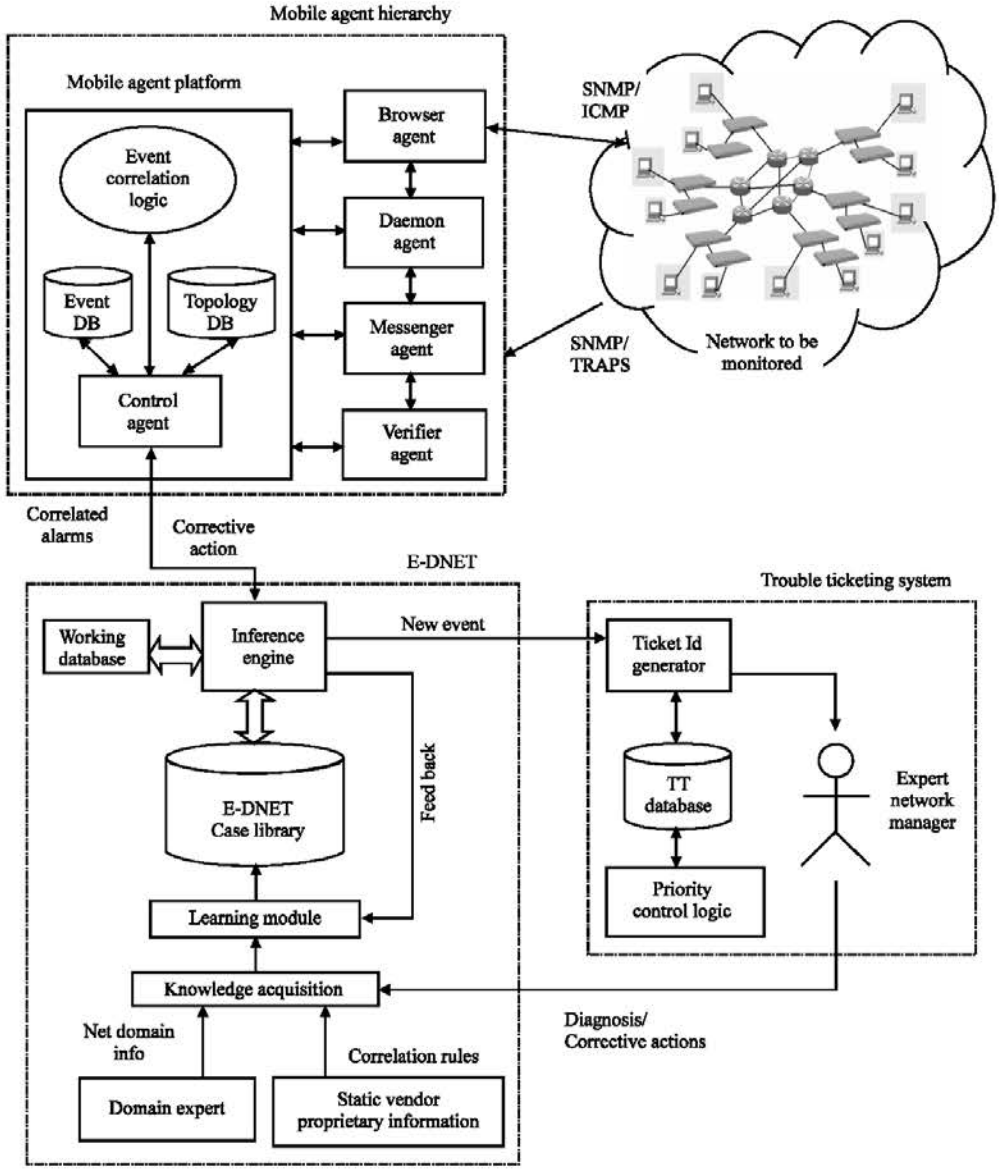

Fig. 1: E-DNet schematic architecture and relationship with other modules

computes a function whose purpose is the evaluation of the presence of a specific property of the system visited (for example, we might think of the verification of a software version, or the available space on disk not below a fixed threshold, or the verification of some log files, etc.). The verifier agent then reports the server, from which it departed, the list of the nodes that satisfy this property.

The mobile agent platform: All the coming events are captured by browser agent and are stored in event and topology databases. The Control Agent verifies integrity of databases and used them with Event Correlation Logic to correlate the coming events with the appropriate faults. When the fault is localized, these correlated alarms/events are sent to the Inference Engine of E-DNet.

E-DNet: Being the Core Component of Network Management Solution, the E-DNet captures the correlated alarms from MAP. The Inference Engine of E DNet uses E DNet Case Library as well as the Working Database for finding the associated corrective action which should be initiated to resolve the network fault and if the corrective action is automated it executes the necessary scripts in the form of corrective action.

The Knowledge Acquisition Process (KAP) initiated by E-DNet is used to build the Diagnosis/corrective

actions as part of E-DNet Case Library. The Domain Expertise and the Vendor specific information are sources of input for KAP Human Network Expert advice in from of corrective actions are also incorporated by KAP. The Learning Module in between E DNet Case Library and KAP constantly refines the E DNet case library by building the new corrective actions to be taken for the new event.

The trouble ticketing system: In case of a New event, it is piled up to the Trouble Ticketing System. All new events are assigned Ticket Id by the Ticket Id Generator Module and are stored in Trouble Ticketing Database. The events are categorized on basis of Priority by Priority Control Logic as some events require immediate attention than others. All such events are compiled by experts/ Network Administrator who initiate sequence of corrective actions for a particular event This Human Network Expert advice in from of corrective actions is piled in Knowledge Database by KAP.

IMPLEMENTATION

A prototype implementation of E-DNet is available and has been tested for a variety of scenarios on MAP (Puliafito *et al*., 2002), while a proof-of-concept Helpdesk/Trouble Ticketing System was built for testing purposes. Figure 2 depicts the logical working of E-DNet while troubleshooting network conditions to generate the corrective actions to be taken. Only critical Alarms from MAP were configured to be forwarded to the helpdesk utility, which assigned a unique trouble-ticket ID to the incoming critical event (internally designated as a call in helpdesk parlance) and assigned an expert network administrator from a pre-configured list to the call.

The helpdesk utility also classified the calls into various severity levels by the use of Priority Control Logic. For instance, network infrastructure related calls (routers, switches, or critical servers) are assigned the highest priority whereas individual host related calls are assigned the lowest priority. The helpdesk utility then provides an easy-to-use interface to the Network Administrator to record the steps taken in the diagnosis and resolution of the reported fault serving as an input to E-DNet.

One of the features introduced in the latest MAP version is the compliance with the MASIF standard (Milojicic *et al*., 1999). This is a standard for the interoperability of several agent systems and is introduced by OMG. In fact, during the last few years, several agent systems have been created; such systems are different in architecture and implementation. Thanks

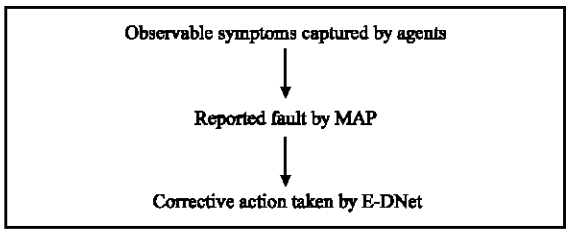

Fig. 2: Sequence of events

to the improved research in the area of mobile agents, new basic requirements have appeared which any agent platform should have. However, even if such functionalities are present in many agent systems, the differences in architecture and implementation make their interaction very difficult (or even impossible), thus preventing the agent technology to be used in some applications.

The MASIF standard has been introduced in order to solve this problem and help the interaction among different systems, by defining a set of basic interfaces with which the systems must comply, in order to communicate and interact. The interfaces specified by MASIF are the following ones:

- **MAF agent system:** It concerns with the management, the transfer and the execution of agents.
- **MAFF inder:** This provides the methods needed for locating and identifying agent and agent system, by considering a central component for their registration.

The compliance with the MASIF standard enables our agent system to interact with the Common Object Request Broker Architecture (CORBA) world; this way, our system can re-use and access legacy services present in the network. The flexibility provided to the proposed management system is considerable, since the possibility of integrating management mechanisms based on CORBA (Zahavi and Mowbray, 1995) is added to the peculiar aspects of the mobile agent.

E- DNet basically works on three inputs:

- Correlated alarms from the MAP which provide the relationship between various event patterns and help E-DNet in learning about the various symptoms of a reported network fault.
- The expert action taken by the Network Administrator in diagnosing and resolving the critical network problem.

- Static pre-configured rules describing the relationship between device specific faults and recommended corrective action by the device vendor (wherever such information is available).

The SNMPTRAP utility was utilized to generate specific events and event patterns for testing purposes, which were correlated by MAP and forwarded to the helpdesk utility. The Static pre-configured rules for the E-DNet system were adapted from the Cisco Event Correlation Guidelines, which is a detailed document published by Cisco Inc. describing detailed event correlation scenarios for its devices and the causes for such events.

E-DNet incorporates the logic for decoding the event stream from MAP. For this purpose, it makes use of the event APIs to enable it to parse the incoming events and extract useful information from it such as event name, event OID (object identifier), event source, event severity etc. This information includes the SNMP community strings needed by E-Net to execute any SNMP-SET operations on network devices as part of the automated corrective actions:

The E-DNet employs heuristic techniques for finding correct responsive corrective action for faults which includes Forward Chaining and Backward Chaining. In case of Forward Chaining, the Hypothesis is taken first and the rules are verified to prove the Hypothesis whereas the process reverses in case of Backward Chaining. Given a symptom (or a set of symptoms), the heuristics initially associate the symptom(s) with a particular corrective action and assert the strategy (and/or tests) to adopt in diagnosis. The System Event Correlation Strategy is framed on the following patterns

If < Symptom_Patterns >
THEN< Diagonsis-Class>

Here, the <Symptom Pattern> depicts Heuristic Rule specifying various network conditions and <Diagonsis-Class> is Hypothesis showing the Corrective Action to be taken to troubleshoot the Network. The Inference Engine evaluates the Hypothesis to be TRUE if the corresponding Rules evaluates to TRUE. The following examples depict the specific network conditions as SNMP Traps and System Log Messages and the Corrective Actions executed by E- DNet.

Example # 1: Consider an environment with Cisco router 7500 and NMS station gets the following System Log (Syslog) message:

Error: SNMPTRAP (Chassis Alarm)

Problem description: The operating Temperature of the networking device exceeds the threshold limit.

Corrective action: If String (Message (Chassis Alarm))

THEN Shut_Router (R)

The above stated Corrective Action is applied to shut down the networking device Router (R)

Example # 2: Consider Cisco 7500 router, a Cisco 5000 Catalyst Switch in a LAN Environment and the following syslog messages are logged by NMS:

Error: SNMPTRAP (E0_Router, bulk_traffic)

Problem description: The Interface E0 of Router experiencing bulk traffic i.e., No. of packets entering the interface exceeded threshold limit.

Corrective action: If String (Message ((Snmptrap_Bulktraffic))

Then Router(R, Increase_Queuelength) and Switch (S, inc_time_out, inc_retry)

The correlation rule stated above is applied to try and alleviate the congestion conditions by increasing the Router Queue length, Timeout and Retry values for Packets sent over the Router and Switch Interface.

CONCLUSIONS AND FUTURE WORK

This research establishes the feasibility of E-DNet, which is an attempt towards building a complete Network Management solution, one which not only has the capability of detecting and reporting network faults, but also diagnosing and executing corrective actions to resolve them, thereby completing the loop. The potential of E-DNet is immense and the impact of such a system on improving the overall Return on Investment (ROI) figures for businesses can be huge. It can also be viewed as a starting step in building completely autonomous systems for Network Management which is self-diagnostic and self-healing in nature.

Future work shall involve establishing contact with experts in the Network Management domain to enable the creation of as many rules as possible for the E-DNet system. Further, intelligence needs to be built into the E-DNet system to enable it to track the results of the automated corrective action initiated by it, so that it can

continuously evaluate the effectiveness of its own actions and improve on it. Moreover, similar problems can be handled in different ways by two designated experts. E-Net needs to be intelligent enough to decide on which of the two approaches can be suitable adapted as the corrective action. The authors are also evaluating whether the event information that E-DNet has access to, can be analyzed over a period of time to build temporal and spatial relationships between events which can then be examined by E-DNet to predict network faults and prevent them from occurrence or minimizing their impact. Such a solution can be based on the historical analysis of network faults.

REFERENCES

Andrew, M., R. Sterritt1 and D. Bustard, 2003. Autonomic Computing Correlation for Fault Management System Evolution. IEEE Press, pp: 240-247.

Bush, S.F. and S. Kalyanaraman, 2006. Management of Active and programmable Networks. J. Network Syst. Manage., 14: 1-5.

Christopher, L., 1995. Experience and trends in AI for network monitoring and diagnosis. Proceeding of IJCAI Workshop on AI in Distributed Information Networks, pp: 57-60.

Cisco Event Correlation Guidelines, 2006. Website www.Cisco.com/doc/eventcorrelate_event.pdf.

Davis, R., B. Buchanan and E. Shortliffe, 1977. Production Rules as a representation for a knowledge-based consultation program. Artif. Intell., 8:15-45.

Doyle, R.J. and A.R.K. Sastry, 1991. Application of knowledge-based network management techniques for packet radio networks. In: Proceeding of MILCOM '91, VA (USA), pp: 414-419.

Goodman, R.M. and H. Latin, 1991. Automated knowledge acquisition from network management databases. In: Integrated Network Manage., II: 541-549.

Gurer, D., K. Irfan and O. Richard, 1998. An Artificial Intelligence Approach to Network Fault Management IEEE Press.

Hartely, R.T., 1984. CRIB: Computer fault finding through knowledge engineering. IEEE Press, pp: 76-83.

HP Open View Network Node Manager, 2006. Advanced Edition. Website :http://www.openview.hp.com/products/nnm.

Jeffrey, O. and D.M. Chess, 2003. The vision of autonomic computing. Computer, 36: 41-52.

Jordaan, J.F. and M.E. Paterok, 1993. Event Correlation in Heterogeneous Networks Using the OSI Management Framework. Integrated Network Manage., III: 683-695.

Laffey, T.J., W.A. Perkins and T.A. Nauyen, 1986. Reasoning about Fault Diagnosis with LES. IEEE Press, pp: 13-20.

Monica, F. and G. Jonathan, 1991. A knowledge-based system for fault localisation. In: Integrated Network Manage., II: 519-530.

Marques, T.E., 1998. A symptom-driven expert system for isolating and correcting network faults. IEEE Commun., 26: 6-13.

Mcauley, A. and R. Morera, 2003. Fault localization and self-healing with dynamic domain configuration. IEEE Press, pp: 977-981.

Milojicic, D., M. Breugst and S. Covaci *et al.*, 1999. MASIF: The OMG mobile agent system interoperability facility. In: 2nd International Workshop on Mobile Agents, (MA'98), Stuttgart, Germany, pp: 628-641.

Peacocke, D. and S. Rabie, 1987. Knowledge based Maintenance in Networks, GLOBECOM'1987, USA., pp: 46-50.

Puliafito, A., O. Tomarchio and L. Vita, 2002. Design and implementation of a mobile agent platform. J. Syst. Architecture, 46: 145-162.

Roth, H. and D.A. Waterman, 1983. Building Expert Systems, Addison-Wesley Publishing Company.

Sethi, A.S. and M. Steinder, 2002a. Intelligent real time network management. In: Proceeding 23rd Army Science Conference, Orlando, Florida, USA., pp: 242-248.

Sethi, A.S. and M. Steinder, 2002b. Fault localization and self-healing mechamsms for fcs networks. In: Proceeding 23rd Army Science Conference, Orlando, FL, pp: 340-349.

Smyth, P., S. Joseph, G. Oliver and G. Rodney, 1991. Combining knowledge-based techniques and simulation with applications to communications network management. Integrated Network Manage., II: 505-515.

Steinder, M., 1994. A survey of fault localization techniques in computer networks. Watson Research Center, Hawthorne, NY, USA.

Sutter, M.T. and E.Z. Paul, 1988. Designing Expert System for real time diagnosis of self correcting networks. IEEE Networks, pp: 43-51.

Tim, O., 1995. Fault Identification in Network Management: A review and a new approach Technical Report, Experimental Knowledge Laboratory, University of Massachusetts, MA, pp: 1-21.

Zahavi, R. and T.J. Mowbray, 1995. The Essential CORBA: Systems Integration Using Distributed Objects, John Wiley and Sons, New York.