In Proc. First ARL/ATIRP Annual Conference,
College Park, MD (Jan. 1997), pp. 251-256.

A SPREADSHEET-BASED SNMP SCRIPTING ENVIRONMENT FOR BATTLEFIELD NETWORK MANAGEMENT

Adarshpal S. Sethi
Pramod Kalyanasundaram
Christopher M. Sherwin
Dong Zhu

Department of Computer and Information Sciences
University of Delaware, Newark, DE 19716
{*sethi, kalyanas, sherwin, dzhu*}*@cis.udel.edu*

ABSTRACT

Effective management of battlefield networks requires a hierarchical network management architecture wherein managers can dyanmically delegate management tasks to intermediate managers. In this report we describe a novel paradigm called the spreadsheet paradigm which extends the traditional flat SNMP management model to a hierarchical architecture. We present a spreadsheet MIB, language, and event model that is located at a proxy node acting in the role of an intermediate manager and controllable by a manager. An important feature of the spreadsheet-based proxy is that it permits managers to configure dynamic MIB views based on application requirements that can be quite different from the underlying MIB structures at the agents. The spreadsheet-based proxy architecture presents an easy to use and comprehend abstraction and interface to extend a standard management framework.

Keywords: Network Management, Hierarchical Management, Proxy Agents, Spreadsheet Paradigm, SNMP.

INTRODUCTION

Management of battlefield networks poses unique challenges: rapid configuration and setup, quick adaptation to constantly changing conditions, critical response times, the need for survivable multi-level distributed management and control, and flexible routing and congestion control. Our research in network management at the University of Delaware has explored the use of proxy architectures for hierarchical management of a complex distributed system. Proxy architectures provide a flexible, dynamic management solution for the management of battlefield networks by permitting distribution of control and management functions over a hierarchical management structure. In proxy architectures, management authority can be delegated to Intermediate Managers (Proxy Agents) which can react quickly to changing battlefield scenarios.

To effectively manage large distributed systems and complex internetworks, a management model must go beyond the traditional flat model of a single manager communicating with a large number of agents. Hierarchical management and peer-to-peer communication among managers are attractive alternatives that need to be explored for distributed system management [1]. However, the most popular management framework, the SNMP framework (which includes both the SNMP and the SNMPv2 protocols) [2], [3], [4], only supports the flat management model. Although the SNMPv2 protocol includes an InformRequest PDU intended for manager-to-manager communication in a hierarchical environment, it cannot be effectively used because the SNMP framework lacks essential support for hierarchical management. Although the SNMP framework defines the concept of a *proxy agent* as an agent that acts on behalf of other agents, traditionally SNMP has used proxy agents only in a *pass-through* role, wherein a proxy might facilitate the implementation of administrative or security policies but otherwise passes the manager requests and agent responses through in an essentially transparent mode. The fact that a proxy can be used as an intermediate manager for hierarchical management is recognized by SNMPv2, but the framework provides no means for managers to delegate tasks to intermediate managers or for peer-to-peer communication between intermediate managers during the execution of these tasks.

Management by delegation is a well-known strategy [5], [6] for implementing hierarchical management, but so far the SNMP community has been unable to take advantage of it because the delegation primitives have not been integrated with the SNMP framework. Our research into new models for distributed system management has led us to propose a new paradigm – which we call the *spreadsheet paradigm* [7], [8] – that incorporates management by delegation concepts into the SNMP framework to facilitate the management of distributed systems. The spreadsheet paradigm allows a proxy acting as an intermediate manager to offload the routine management tasks of the manager and facilitates user configurability of management information and control in a *value-added* manner. This is achieved by providing a scripting MIB and language specially designed for management tasks in SNMP. It also permits managers to specify to their peers the tasks they would like to be executed on their behalf and for them to be notified on the occurrence of appropriate events.

The main objectives of the spreadsheet paradigm are: (a) to introduce a powerful intermediate manager that enhances (but preserves) the existing SNMP framework and provides value added functions in addition to pass-through capabilities; (b) to provide an environment that supports user configurability of management information. A user can define logical views that are different from the underlying Management Information Bases (MIB) and manipulate those logical views; (c) to support basic primitives, events and operations via a scripting MIB and language that allow a user to build fairly complex network management tasks. A user can build events using the basic primitives supported by the framework. These user-defined events are then available for generating more complex events. A user can thus set up a hierarchy of events based on the basic and user-defined events; and (d) to present the user with an abstraction and interface that is easy to comprehend and use.

The major benefits of the spreadsheet paradigm are that it permits distribution of control between the manager and intermediate managers or between peer managers through the use of spreadsheet-based scripts and it allows *dynamic* creation of user-defined views of managed objects including relationships between objects belonging to different MIBs. This paradigm fits naturally into the existing Internet Management framework because a spreadsheet is in essence a two-dimensional table which is the fundamental structure supported by this framework. It relieves managers from some of the elementary lower-level chores of management thus allowing them to concentrate on higher level tasks required by a management application. Managers can dynamically decide what actions are to be performed and how they are to be executed at the proxy-agent/intermediate-manager with a high degree of flexibility.

The ability to selectively configure and view management information is a key feature that is missing in the SNMP framework. With our approach, the user can set up relevant views of management information suited to the task on hand. The user can thus organize management information based on the user's requirements, which can be quite different from the underlying MIB structure. Further, the user can perform operations on these user structured views of management information independent of the structure of the standard SNMP MIBs.

Existing models [3], [9] provide facilities for event definitions and object relationships. However, for these event definitions and object relationships to be available for use by a manager, they must be predefined in the MIBs supported by an agent. Some MIBs (such as the RMON MIB [10]) allow the manager to choose the objects that will enter into a relationship (e.g., choosing the interface whose statistics are to be observed), but the nature of the relationships is built into the MIB table structures and cannot be modified by the manager. To allow managers to perform more sophisticated control functions, there is a need for a paradigm that permits *dynamic* specification of object relationships and event definitions. A paradigm such as the one proposed here, allows a user to define more complex events and specify relationships between objects distributed across nodes. Such a paradigm enhances the power of the existing framework and allows dynamic configuration of information and control. The paradigm provides a distributed, robust environment for performing management tasks while seamlessly integrating with the existing framework.

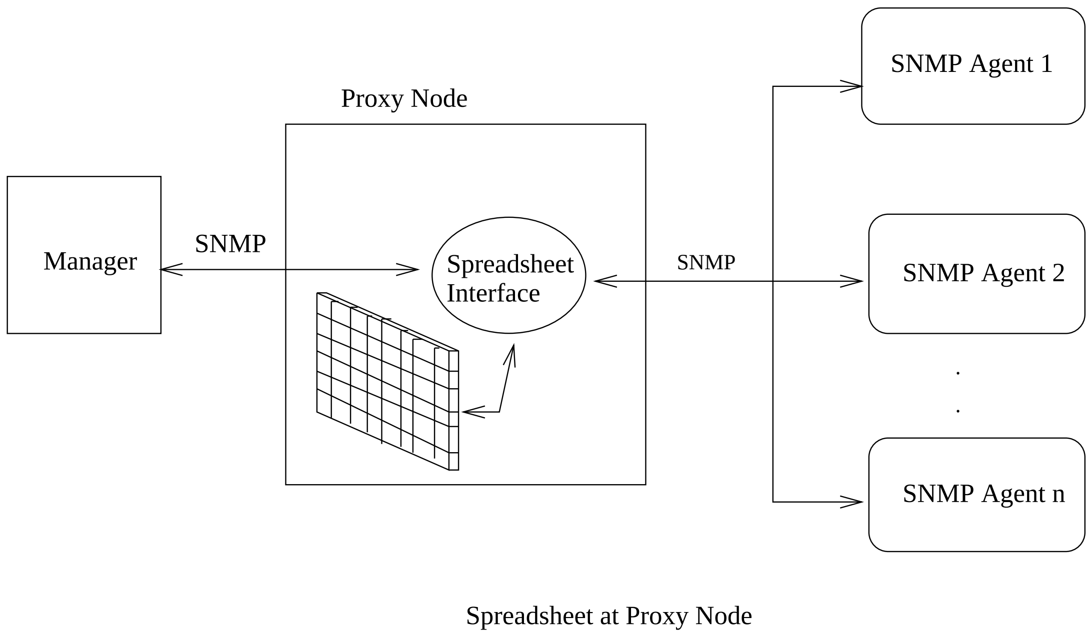

Fig. 1. A proxy agent with a spreadsheet interface

THE SPREADSHEET PARADIGM

An essential component of the spreadsheet paradigm is an abstraction of a spreadsheet. A spreadsheet is composed of *cells* arranged in a two-dimensional matrix consisting of *rows* and *columns*. The number of rows and columns in a spreadsheet is arbitrary and is only limited by memory constraints in the implementation or platform. A cell is the fundamental unit of operation in the spreadsheet paradigm. A cell contains a *control information part* and a *data part*. The control part can serve as a repository for event specifications, relationships between management objects or references to other cells. The control part dictates the rules for collection of information or relationships between objects. The data part contains the data collected as a result of executing the control information specification. For instance, the control part may specify that the cell should contain the result of summing two or more counters in different nodes (or different MIBs). The data part contains the result of such a summation. A user of the spreadsheet is given the flexibility of creating, deleting and modifying cells. In addition, a user can view and clear data in the cells. Several cells can be combined together to form a complex event which can be stored in another cell. This structure facilitates the construction of hierarchical events.

For example, a user could create a cell which contains the following as its control part: “Is the interface between Node A and Node B of the network down?”. The data part of the cell would be “true” if the interface is down and “false” otherwise. Another cell could represent the same condition but between Nodes C and D. Now if the management user wants to check the interface condition for all four nodes, such a condition can be checked by combining the first cell and the second cell and storing this condition in a third cell i.e., the condition for the third cell would be “Are Cell1 and Cell2 true?”. Thus cells which have already been defined could be used to form more complex conditions.

Figure 1 shows the spreadsheet abstraction located at a node that functions as a proxy (or an intermediate manager). The proxy accepts SNMP requests from the manager and enters them into the spreadsheet that is maintained at the proxy. The control information entered into the spreadsheet contains scripts written in the Spreadsheet Language (SSL), a specially designed scripting language for use with the spreadsheet paradigm. The scripts are interpreted locally at the proxy which may result in SNMP requests to be forwarded to one or more agents. The agents' responses are processed as specified by the scripts in the spreadsheet cells. As a result of this processing, updates may occur in the data contained in one or more cells. An operation initiated by a single cell may result in a data update in the cell which initiated the operation as well as in cells that are dependent on the cell that initiated the operation. The spreadsheet abstraction can be supported using the standard SNMPv2 protocol operations.

The two essential features supported by this paradigm are: a) specification of relationships between objects *across MIBs* and b) flexible, hierarchical event building. Relationship specification allows a user to specify a logical, temporal or other such condition to relate two or more objects belonging to different MIBs. Hierarchical event building allows a user to specify simple events (in cells) and use those events to build subsequent events. To achieve the spreadsheet perception, the paradigm must allow the creation, deletion, reading and modification of the control and data part of the cells.

The spreadsheet paradigm is intended to and will provide the user with the power and flexibility to specify criteria to selectively view management information spanning several nodes in a network. In addition, it will allow the user to dynamically define relationships and specify control information for collection of data, monitoring network resources and scheduling routine management information processing. Further, the paradigm allows a user to tailor network management to suit individual networks. By virtue of the fact that this paradigm allows customization, management users can add specifics related to management objects at the spreadsheet and allow the proxy to handle such object dependencies.

SPREADSHEET OPERATION

Figure 2 shows the basic components contained in an implementation of the spreadsheet entity. The primary function of the spreadsheet MIB (*ssmib*) is to implement the spreadsheet abstraction using SNMP tables. User operations on cells map to operations on tables that are part of this MIB. The proxy function module coordinates the activities of the various components at the proxy node. When the proxy receives an SNMP request from the manager, the proxy function module performs the necessary operations on the spreadsheet MIB to implement the cell abstraction. Once the request has been carried out, the proxy function module responds to the manager that requested the operation. If a control value is entered into a cell, the objects that need to be polled are forwarded to the polling subsystem. Once the polling entries are set up, the proxy requests them on an as needed basis, for spreadsheet processing. The proxy module interacts with the event model to perform event based processing of the spreadsheet. When the script in a cell needs to be executed, the proxy function module interacts with the interpreter to process the cell control information.

The polling subsystem plays a vital role in implementing the spreadsheet abstraction. When a user sets up information in a cell, there is a need to constantly update the value(s) contained in the cell. This can be achieved by polling the managed objects referenced in the cell. In order to perform such polling, the proxy maintains polling tables that facilitate the polling process that automatically updates the cell values. This allows a user to view the spreadsheets that are set up at the proxy and expect the cells to contain values that are current (within a certain time granularity). The polling subsystem allows polling entries to be set up based on a cell id. Thus the proxy function can request the polling subsystem to set up polling entries associated with a cell and retrieve all the values that are part of a cell.

The spreadsheet paradigm supports a scripting language called SSL (Spreadsheet Language) that can be used by a user to set up sheets of control. The spreadsheet language is interpreted by an interpreter and scripts that are set up in the various cells can be executed under the control of this interpreter. The interpreter performs the functions of syntax checking, run time error checking, detection and reporting.

A language that targets a network management environment must be able to support features that facilitate the specification of network management tasks coupled with user flexibility and expressive power [11]. The SSL has been specially designed for this purpose. The SSL supports arithmetic operators that allow a user to perform expression evaluation. It further allows a user to specify numeric (dot notation) or symbolic form of managed objects. It allows a user to specify a fully qualified managed object i.e., the OID and the host on which the managed object resides. Since the spreadsheet paradigm sup-

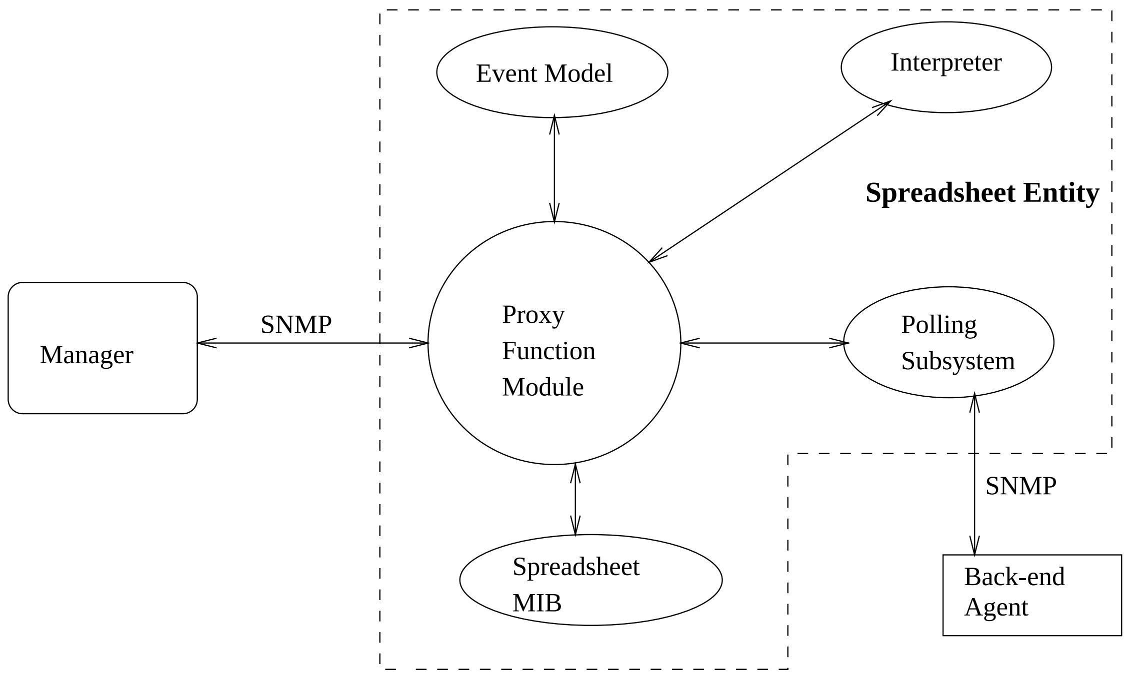

Fig. 2. Spreadsheet Entity

ports access to managed objects on different nodes, it is important for a user to be able to specify the actual managed object qualified with a host name. The language is married to the spreadsheet paradigm and offers facilities to use and manipulate cells. The language supports cell assignment, and cell local variables for temporary storage. Further, the language provides the standard constructs for iteration (*forall*, *while*), condition (*if…else*) etc. This allows a user to setup procedural scripts in the cells. A cell can contain a procedure that can be executed from another cell.

The spreadsheet supports two modes of operation: synchronous or request/response mode and asynchronous or event mode. In the synchronous mode, the manager requests some operation to be performed using one of the standard SNMP protocol operations and the proxy responds after processing the request. In the asynchronous mode, the user sets up events and actions associated with such events. These events are constantly monitored by the proxy. On occurrence of any of the events being watched, the proxy carries out the associated actions which may include notifying the manager. This mode of operation allows the manager to successfully delegate some of its routine tasks to the proxy. In order to support asynchronous event processing, the spreadsheet entity uses an underlying event model that permits event and temporal criteria specification.

The spreadsheet paradigm defines certain basic events (poll event). The event model allows a user of the spreadsheet paradigm to set up events based on cells and the basic events supported by the paradigm. The user can also use events that have already been defined in conjunction with the SSL, to construct more complex events. This hierarchical event building can be performed recursively until (in some cases) a complete management task can be modeled as a set of events. The event model uses ordering of events based on the local clock. This ordering obviates the need for clock synchronization. Also, the events are defined at the proxy and hence any ordering other than the one based on the local clock at the proxy would be inconsistent. The spreadsheet paradigm event model also supports features that allow the backward propagation of events to ensure that all the cells that depend on a particular event are processed and the spreadsheets updated.

CONCLUSIONS

In conclusion, we have presented a new paradigm for network management which will allow the SNMP management framework to be extended to hierarchical management environments such as for battlefield networks. We described the proxy architecture that supports the spreadsheet paradigm and outlined some of the features of the language and event model that form an integral part of this paradigm.

A prototype implementation of the spreadsheet paradigm based in a proxy agent is currently in progress. The implementation includes a graphical user interface that can be used by a manager to construct, load and execute scripts in the proxy. These ideas have also been submitted to the IETF Network Management Working Group on Distributed Management *disman* with a view to influence the emerging standards in this area.

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government.

REFERENCES

[1] A.S. Sethi, Y. Raynaud, and F. Faure-Vincent, editors. *Integrated Network Management IV*. Chapman and Hall, London, 1995.

[2] J. D. Case, M. S. Fedor, M. L. Schoffstall, and C. Davin. *Simple Network Management Protocol (RFC 1157)*, May 1990.

[3] J. Case, K. McCloghrie, M. Rose, and S. Waldbusser. *Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2) (RFC 1905)*, January 1996.

[4] M. T. Rose and K. McCloghrie. *Structure and Identification of Management Information for TCP/IP based internets (RFC 1155)*, May 1990.

[5] Y. Yemini, G. Goldszmidt, and S. Yemini. Network Management by Delegation. In I. Krishnan and W. Zimmer, editors, *Integrated Network Management II*, pages 95–107. North Holland, Amsterdam, 1991.

[6] K. Arai and Y. Yemini. MIB view language (MVL) for SNMP. In A.S. Sethi, Y. Raynaud, and F. Faure-Vincent, editors, *Integrated Network Management IV*, pages 454–465. Chapman and Hall, London, 1995.

[7] P. Kalyanasundaram, A.S. Sethi, and C. Sherwin. Design of A Spreadsheet Paradigm for Network Management. In *Proceedings of the 7th IFIP/IEEE Workshop on Distributed Systems: Operations and Management*, L'Aquila, Italy, October 1996.

[8] P. Kalyanasundaram, A.S. Sethi, C. Sherwin, and D. Zhu. A Spreadsheet-based Scripting Environment for SNMP. In *Proceedings of the 5th IFIP/IEEE International Symposium on Integrated Network Management (to appear)*, San Diego, CA, May 1997.

[9] M. Hasan. An Active Temporal Model for Network Management Databases. In A.S. Sethi, Y. Raynaud, and F. Faure-Vincent, editors, *Integrated Network Management IV*, pages 524–535. Chapman and Hall, London, 1995.

[10] S. Waldbusser. *Remote Network Monitoring Management Information Base (RFC 1757)*, February 1995.

[11] D. Holden. Predictive Languages for Management. In *Integrated Network Management I*, pages 585–596. North Holland, Amsterdam, May 1989.