

# A Spreadsheet Paradigm for Network Management and Control

Adarshpal S. Sethi and Pramod Kalyanasundaram  
(email: sethi@cis.udel.edu, kalyanas@cis.udel.edu)

Department of Computer and Information Sciences  
University of Delaware, Newark, DE 19716

**Keywords:** Network Management, SNMP, Spreadsheet, Proxy Agent, Internet Management

## 1 INTRODUCTION

Network Management has gained momentum over the past few years and has become one of the fast paced research areas. The need for efficient management of networks of varying sizes and types has led to standardization efforts in the Internet and OSI communities. These efforts led to the Internet and OSI models of network management [IS191a, CFSD90]. These two models are somewhat similar at the architectural level but the differences are significant at the functional level. For instance, the Internet Management model supports a flat MIB [RM90] whereas the OSI model supports an object-oriented information framework [IS191b]. The OSI model for management objects is more powerful and expressive than its Internet counterpart. However, the Internet model scores in terms of its simplicity and pragmatic approach. The focus of this paper is to propose a scheme that can be built around the simple Internet management model that would enhance the power of the flat MIB structure without necessarily increasing the complexity to a level where implementation becomes a wish rather than reality. This scheme would provide the manager and management user with additional power to manage multiple MIBs located at different nodes.

In the context of the Internet model, a *proxy* is an agent that acts on behalf of one or more nodes. A proxy accepts management requests from the managers and forwards them on to appropriate agents, and similarly forwards responses and event notifications (or traps) from the agents to the managers. A proxy may be used when the agents do not speak the same protocol as the managers, but is more commonly used for administrative or security reasons.

This paper presents a new paradigm for network management and control that we call a *spreadsheet paradigm* as it is based on an abstraction of the well-known concept of a *spreadsheet*. The spreadsheet paradigm may be employed at any location in the network, including management stations, proxy nodes, or agents, but its maximum potential is best realized when implemented at a proxy. This paper concentrates on discussing the spreadsheet paradigm as located at a proxy, but also includes a brief description of other possible scenarios.

The major benefits of the spreadsheet paradigm are that it allows *dynamic* creation of relationships between objects belonging to different MIBs and permits distribution of control. This paradigm fits naturally into the existing Internet Management framework be-

cause 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 management application. Managers can dynamically decide what actions are to be performed and how they are to be executed at the proxy agent with a high degree of flexibility. Traditionally, a proxy agent has mostly provided a *pass-through* capability to the manager, but with the deployment of the spreadsheet paradigm, a proxy will be capable of *value-added* functionality. This behavior is achieved while continuing to function completely within the domain of the Internet Management framework and without any modifications to the SNMP (or SNMPv2 [CMRW93]) protocol.

The organization of this paper is as follows: Section 2 justifies the need for such a paradigm and the reasons for selecting the spreadsheet abstraction. Section 3 introduces and describes the details of the paradigm; Section 4 outlines future work and presents conclusions.

## 2 MOTIVATION

This section presents the need for a paradigm that can distribute control and create relationships between objects contained in different MIBs. It also justifies the selection of spreadsheet as a suitable paradigm.

*Why do we need such a paradigm?* Existing SNMP based network management is strictly limited to the services provided by SNMP. The SNMP framework is designed to operate on management information available at a particular node. It is the responsibility of the management application to receive the information from different agents and correlate information across nodes. Although event definitions are supported in SNMPv2, such events are fairly restrictive and primarily allow threshold type events. 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 existing services offered by the SNMP framework and provides a great degree of abstraction and flexibility to the management user.

In addition, for event definitions and object relationships to be available for use by a manager, they must be pre-defined in the MIBs supported by an agent. Some MIBs (such as the RMON MIB [Wal91]) 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.

*What are the advantages of such a paradigm?* The essential advantages of choosing such a paradigm include:

- The ability to relate objects across MIBs using relationships thereby enhancing the power of a flat MIB structure.
- The ability to dynamically specify and create relationships between arbitrary objects.
- Introduction of a powerful intermediate manager. When located at a proxy node, the spreadsheet can be effectively used as an intermediate manager to manage and control several agent nodes.

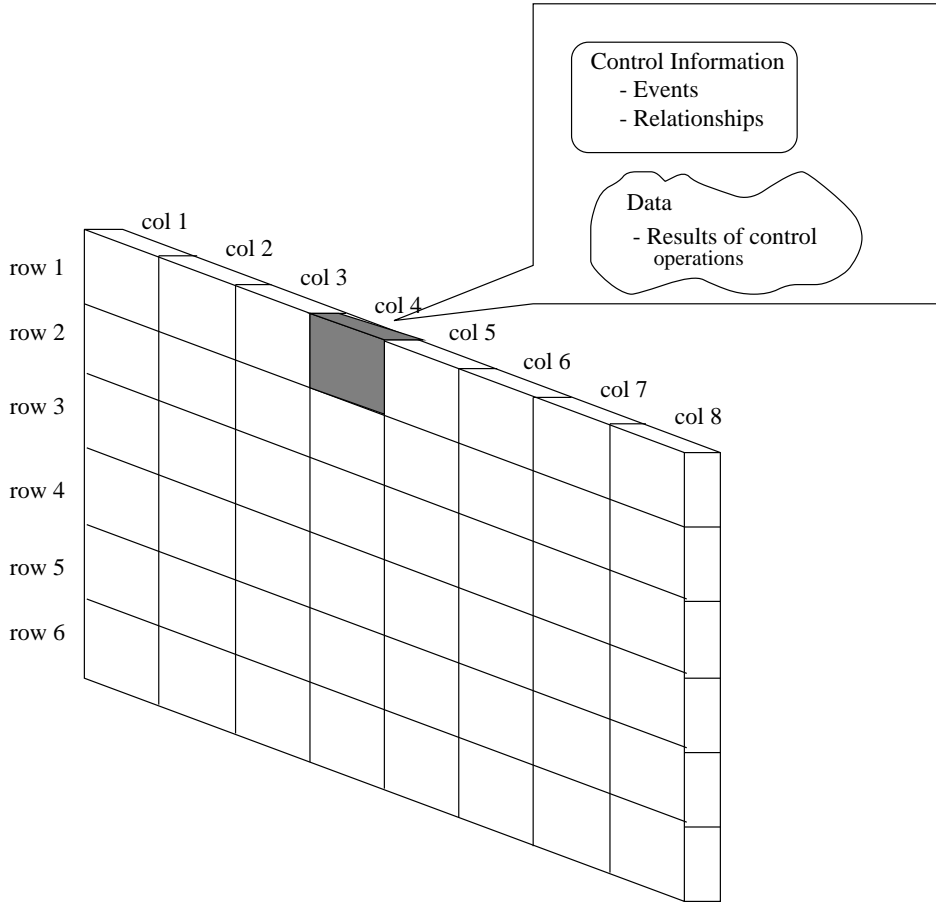


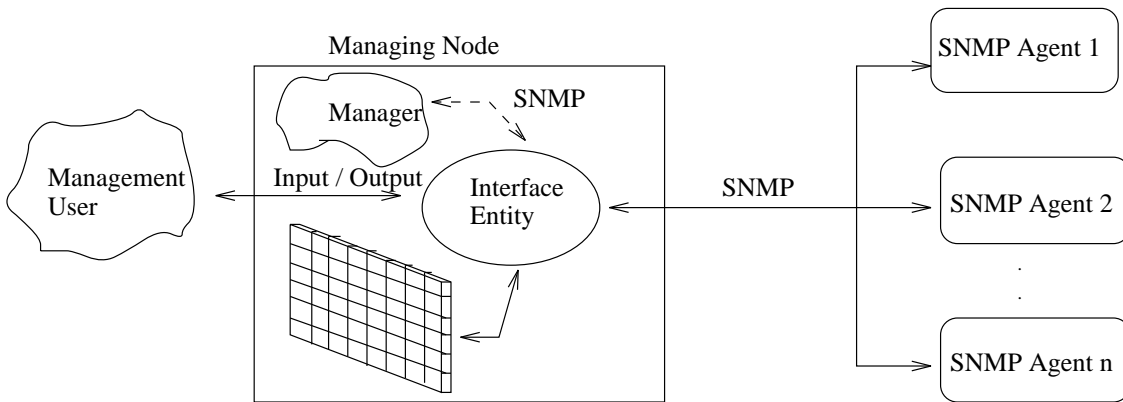
Figure 1: Cell Structure

- Flexibility in specifying events. Events can be specified using logical, relational, temporal, conditional and iterative event specifiers. These events can be used as building blocks for constructing more complex events.
- Extension of the capabilities of the existing Internet Management Architecture without requiring new management protocol or information model.
- Distribution of control and management functions.

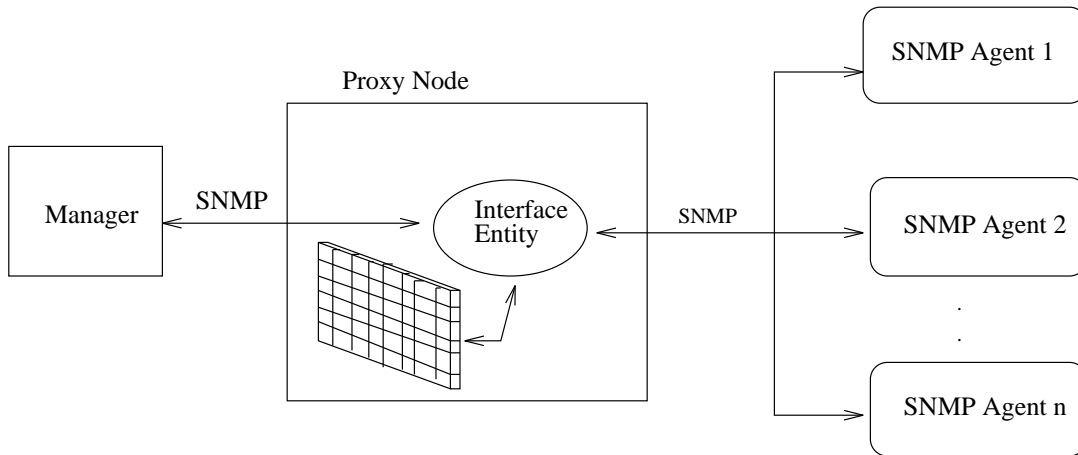
Using this paradigm, a management user can selectively monitor MIB objects and event reports generated by different agents. In addition, a management user can define logical views of interest and can manipulate and control only those logical views. Further, the user can relate objects across MIBs and customize these relationships to suit the local management goals and policies.

### 3 THE SPREADSHEET PARADIGM

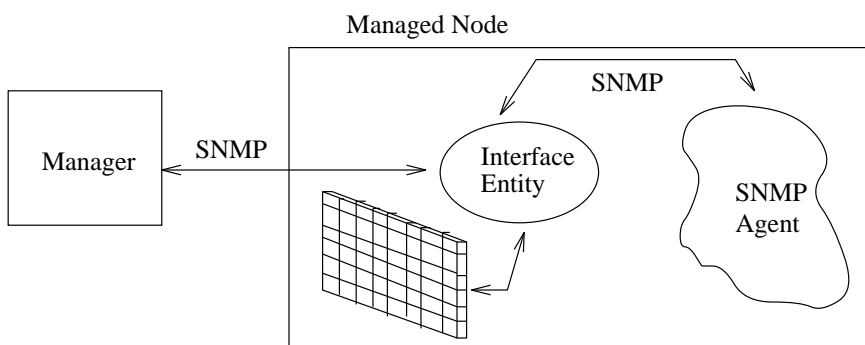
The essential component of the spreadsheet paradigm is an abstraction of a spreadsheet. A spreadsheet is composed of *cells* arranged in a two-dimensional matrix consisting of *rows* and *columns*. The number of rows and columns in a spreadsheet is arbitrary and is only limited by memory constraints in the implementation or platform. A cell is the fundamental unit



(a) Spreadsheet at Managing Node



(b) Spreadsheet at Proxy Node



(c) Spreadsheet at Agent Node

Figure 2: Different scenarios using the spreadsheet paradigm

of operation in the spreadsheet paradigm. A cell contains a *control information part* and a *data part*. This is shown in Figure 1. 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 as its control part the condition “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 linking 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.

In addition to the spreadsheet abstraction, there is an *interface entity* that interacts with the spreadsheet to interpret the control information in the cells and to input data into the spreadsheet (both control and data). The spreadsheet abstraction may be located at the manager, a proxy node or at an agent. Figure 2 depicts the various scenarios. In Figure 2(a) the spreadsheet functions as the managing entity (i.e., manager). In this case, a user enters the network management commands using a user interface to set up the spreadsheet. The interface entity translates the control information stored in the cells to SNMP requests which are then dispatched to the SNMP agents. Under this scenario all the MIBs in the network management domain are abstracted by the spreadsheet. In Figure 2(b), the spreadsheet is located at a node that functions as a proxy. In this scenario, the proxy contains the interface entity that accepts SNMP requests from the manager and enters data into the spreadsheet that is maintained at the proxy. The interface entity enters data and interprets stored control information to send the equivalent SNMP requests to the agents that form the back-end. In this case the spreadsheet abstracts the MIBs that are located at the agents for which the proxy node forms the front-end. The proxy accepts commands from the manager to update the spreadsheet. The control information is interpreted locally. The interpreted information is then mapped to SNMP requests that are forwarded to one or more agents. The agents’ responses are processed in conjunction with the control information part of the spreadsheet cell. 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 manager can use SNMP requests to read the information that has been stored in the cells. In Figure 2(c) the spreadsheet is located at the agents. This scenario does not buy the user too much power since the spreadsheet is now limited to dealing with MIBs local to its own node.

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) (Figure 3) and use those events to build subsequent events. In order to support event definitions in a flexible and easy way, an event specification language is required [Hol89]. To facilitate event definitions, a language or any other method chosen must provide arithmetic, logical, conditional, iterative and temporal expressions. Of these only the temporal expressions need further description. A temporal expression adds time as a dimension to the event specification [Hol89]. In order to perform

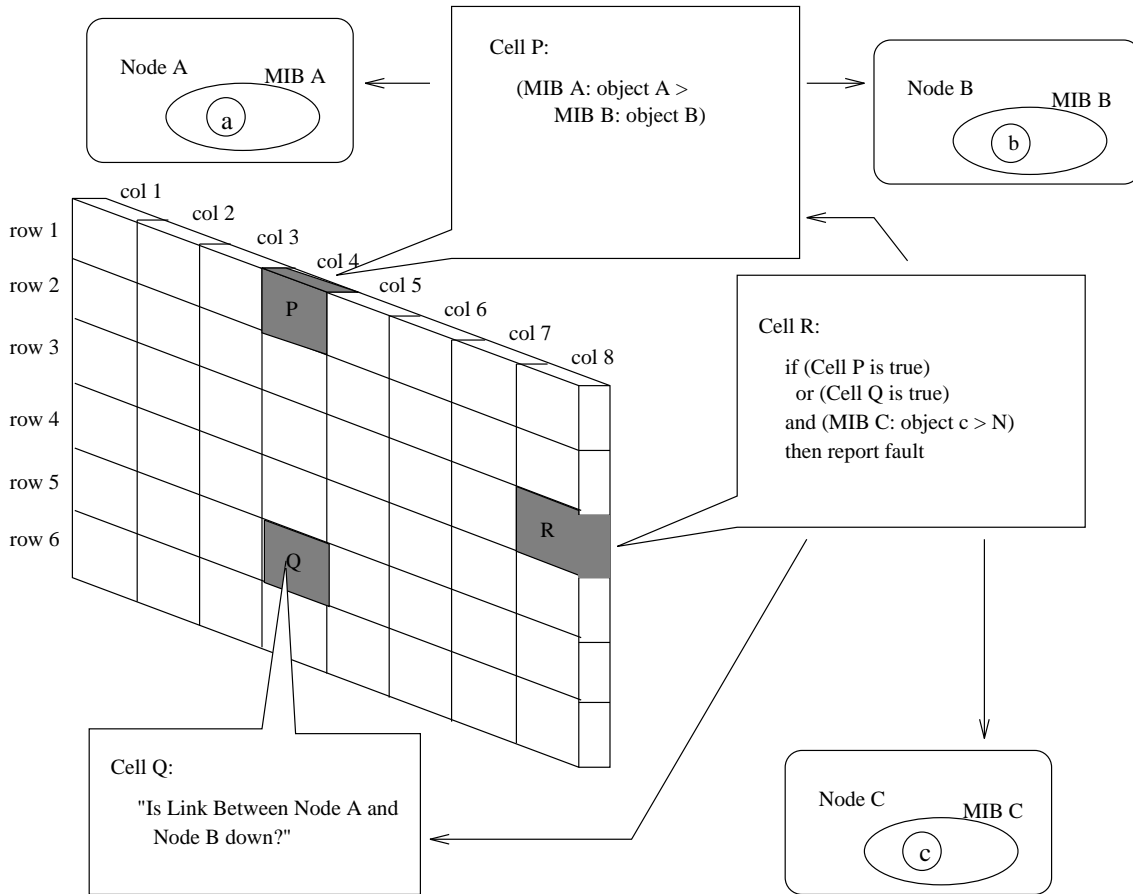


Figure 3: Managed Object Relationships using the spreadsheet

management functions properly, a user should be able to specify (at least) the following temporal expressions:

**before event:** This expression allows a user to specify occurrence of an event at a time instance before the occurrence of another event, but not necessarily immediately before. This basically allows a user to specify ordering of events.

**within time period:** This allows a user to specify a limiting time period for event definition. This is normally useful when a user is interested in knowing if a particular event occurred within a specified time interval or not.

**in sequence:** This expression allows a user to indicate an event definition that specifies the occurrence of a set of events that occur in a particular order. It could be the case that other events occur in the management system, but the specified events should occur in a specified order.

**for time interval:** This expression is useful for specifying procedures that are to be executed for a particular time interval. Examples of such events could be polling a particular managed object, data collection for a user specified period and monitoring a particular variable in one or more nodes.

**at time:** This expression allows a user to specify the occurrence of an event at a particular time instant. This expression could be used to schedule an event at a particular time

instant. Examples of such events could be to start management data collection during peak hours.

For instance, consider the case where a network manager would like to perform collection of data during peak hours. This could be specified by using the **for time interval** and **at** expressions to start the event collection at the beginning of the peak hour period and for the duration of the peak hour period as defined for that particular network.

In addition to the above, any scheme proposed must ensure that there is enough expressiveness to specify elaborate data collection, monitoring and reporting. In addition, the language support must support facilities to include fault detection and diagnosis.

To achieve the spreadsheet perception, the paradigm must allow the creation, deletion and modification of cells. Other operations on cells include: a) **link** which allows a user to link two different cells or groups of cells b) **rearrange** which allows a user to redefine events using existing events c) **store** which allows a user to store control information in a cell. d) **exists** which allows a user to determine if a particular cell is present or not. e) **read** which allows a user to read the contents of a cell and f) **refresh** which changes the control and data part of dependent cells in the spreadsheet when the value of a cell changes. This operation ensures that the data in the cells are updated to reflect changed values.

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 interface entity to handle such object dependencies.

## 4 CONCLUSIONS

In this paper, we have presented a spreadsheet paradigm that allows distribution of control at proxy nodes. This paradigm allows a management user to specify events and relationship between objects located at different nodes. The paradigm is suggested based on the existing architecture and informational model of the Internet management framework. The paradigm proposed allows hierarchical event building which provides the user significant flexibility in performing management tasks. There are many aspects related to this paradigm that need further exploration. These include:

- Further refinement of the spreadsheet abstraction to facilitate easy and efficient network management.
- Definition of syntax and semantics for control specification.
- Design of the interface entity and control interpretation.
- Object Relationship definition aspects.
- Mapping event definitions to SNMP services.
- Implementational aspects of such a paradigm and integration of the paradigm into the existing SNMP framework.
- Analysis of the software solution.

We hope that this paradigm, by virtue of its simplicity, generality and seamless integration with an existing architecture, will prove to be a tool that is not only useful and powerful but will also ease the task of network management.

## References

- [CFSD90] J. D. Case, M. S. Fedor, M. L. Schoffstall, and C. Davin. *Simple Network Management Protocol (RFC 1157)*. DDN Network Information Center, SRI International, May 1990.
- [CMRW93] J. D. Case, K. McCloghrie, M. T. Rose, and S. Waldbusser. *Protocol Operations for version 2 of the Simple Network Management Protocol (SNMPv2) (RFC 1448)*. DDN Network Information Center, SRI International, April 1993.
- [Hol89] D. Holden. Predictive Languages for Management. In *Integrated Network Management, I*, pages 585–596, Boston, May 1989. North Holland, Amsterdam.
- [IS191a] International Organization for Standardization. *ISO IS10040, Information processing systems - Open Systems Interconnection - Systems management overview.*, July 1991.
- [IS191b] International Organization for Standardization. *ISO IS10165-1, Information technology - Open Systems Interconnection - Management Information Services - Structure of Management Information - Part 1: Management Information Model*, July 1991.
- [RM90] M. T. Rose and K. McCloghrie. *Structure and Identification of Management Information for TCP/IP based internets (RFC 1155)*. DDN Network Information Center, SRI International, May 1990.
- [Wal91] S. Waldbusser. *Remote network monitoring Management Information Base (RFC 1271)*. DDN Network Information Center, SRI International, November 1991.