# SHAMAN - An Environment for Distributed Management Applications [1]

*A.S. Sethi, D. Zhu, P. Kalyanasundaram*[2]

*University of Delaware*
*Department of Computer and Information Sciences*
*Newark, Delaware, USA*
*Email: (sethi, dzhu)@cis.udel.edu, pramodk@ignitus.com*

**Abstract**

SHAMAN is a framework for hierarchical management of networks with SNMP that provides flexible and dynamic network management by permitting distribution of control and management functions over a hierarchical management structure. We have implemented a prototype software package that contains a SHAMAN Intermediate Manager (IM), a GUI for controlling the IM and for developing applications for it, and an example application of location management in a mobile tactical battlefield network.

## 1. Introduction

The most popular management framework in use today is the Internet management framework employing a centralized management model in which managing entities (managers) communicate with managed entities (agents) that control the resources in the network nodes. Management by delegation is a well-known strategy [1] for implementing hierarchical management, which can be an effective means of managing the large and complex internetworks that are in operation today. While an effort was made in the IETF Working Group on distributed management (disman) [2] to enhance the capabilities of the SNMP framework with features for distributing application functionality, by and large the SNMP community so far has been unable to take advantage of it because the delegation primitives have not been integrated with the SNMP framework.

Our research group at the University of Delaware has designed a new system called SHAMAN (**S**preadsheet-based **H**ierarchical **A**rchitecture for **MAN**age-ment) that enhances the existing Internet management framework with hierarchical scripting features [3, 4]. The main objectives of SHAMAN are to introduce a powerful intermediate manager that enhances (but is fully compatible with) the existing SNMP framework, provides value-added functions, supports delegation, allows user configurability of management information, and provides an environment for the rapid development of distributed management applications. A prototype implementation of SHAMAN

[1] Prepared through collaborative participation in the Advanced Telecommunications/Information Distribution Research Program (ATIRP) Consortium sponsored by the U.S. Army Research Laboratory under the Federated Laboratory Program, Cooperative Agreement DAAL01-96-2-0002.

[2] P. Kalyanasundaram is now with Ignitus Communications, Acton, MA, USA.

has been developed at the University of Delaware; this paper presents an overview of the SHAMAN architecture, its implementation, and a demo location management application.

## 2. SHAMAN Architecture

The main component of the SHAMAN architecture is an Intermediate Manager (IM) with a MIB called a spreadsheet MIB (SSMIB). A top-level manager can download scripts into the SSMIB using SNMP SetRequests. The IM autonomously executes the scripts to carry out management operations. The SSMIB presents a logical view of a spreadsheet with a two dimensional matrix of cells; the control part of each cell contains a script while the data portion reflects the associated data values resulting from script execution. The IM contains a polling subsystem which periodic polls MIB objects at various agents as required by the scripts in the cells. The IM also contains an Event Model which monitors events occurring within the IM and allows the cell scripts to subscribe to events and associate actions with such events.

SHAMAN supports a scripting language called Spreadsheet Scripting Language (SSL). The language includes features commonly found in procedural languages, e.g., assignments, conditional and loop constructs, and also provides syntactic constructs for cell access and assignment. Its most important characteristic is the integration of network-management-specific constructs, e.g., recognition of OIDs and hostnames within variables in expressions, ability to perform SNMP operations such as Get, Set, and periodic polling, and loop constructs to operate on SNMP tables. In addition, it also permits event specifications to be associated with actions in cells. The SSL 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 detailed description of the SSL language and grammar is available in [5, 6].

A prototype implementation of SHAMAN has been developed at the University of Delaware and is available on the SHAMAN Web page [6]. This implementation is based on the existing UCD SNMPv2 package [7] and runs under Solaris 2.6. The standard SNMP agent in the UCD package has been extended to implement SHAMAN's IM (Intermediate Manager). The extensions include support for the Spreadsheet MIB, the Spreadsheet Scripting Language (SSL), and the event model.

Figure 1 shows the software architecture of the IM and the interdependencies of the various modules that constitute the IM. Among these modules, the MIB Module, the Interpreter Module, and the Cell Module together implement the three logical components of the IM, i.e., SSMIB, SSL (interpretation) and the event model. The Polling Subsystem implements the polling of the agents. The other modules perform support functions like timer services and providing a communication interface for polling the agents.

The SNMP Communications Module deals with encoding, decoding, sending, and receiving SNMP PDUs; this module is a modified version of the UCD SNMP agent. The MIB Module provides an implementation of the SSMIB internal structures and

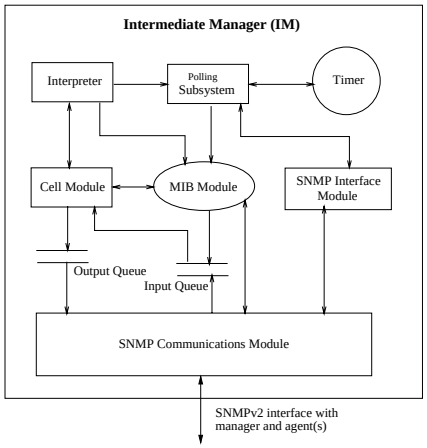


**Figure 1: Software Architecture of an Intermediate Manager (IM) in SHAMAN**

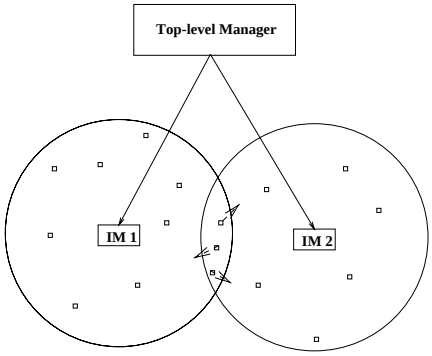


**Figure 2: Hierarchical Location Management for Mobile Nodes in a Battlefield Network**

controls access to it. The Interpreter Module interprets the SSL and interfaces with the Polling Subsystem to create or delete polling entries, and with the MIB module to retrieve or modify SSMIB data table variables. The Cell Module implements the cell abstraction including interfaces for cell creation, deletion, and execution. In addition, this module manages the context of each cell and performs event dependency processing.

In addition to the Intermediate Manager implementation, SHAMAN includes a frontend GUI with a collection of development and design tools to facilitate configuration, script creation, downloading, and management. The UI, which was built using X-Motif and the X toolkit intrinsics, has a language assistant that provides a friendly editing environment to create SSL scripts.

## 3. Battlefield Location Management

The SHAMAN prototype implementation includes a demo application for location management in a battlefield network scenario with many mobile nodes that are controlled from command centers which need to constantly keep track of the current location of each node. Consider a group of nodes that individually move on a battlefield such that each node requires to be periodically monitored by a manager that keeps track of the current location of the node and the amounts of fuel and ammunition left. The manager may take appropriate action if these amounts fall below specified limits. The total number of such nodes to be managed may be too large for a single manager to handle. Moreover, there may be distance constraints so that we may wish to have a node be managed by a manager that is located close by. Figure 2 depicts a hierarchical management solution using the SHAMAN approach that is appropriate for this situation.

In this application, each node has an SNMP-manageable MIB which includes vari-

ables for the x and y coordinates of the node, and its remaining fuel and ammunition. We designate two Intermediate Managers, named IM1 and IM2, with management authority over nodes within their spheres of management as shown by the circles in the figure. Each IM periodically polls each node in its management domain to obtain its current variable values. If any action is required for the fuel or ammunition, then the top-level manager is informed.

As the nodes in this system move around, they may migrate from one IM's management domain to that of the other IM. This may necessitate a "handoff" of the management authority over this node to the second manager. Each time a node's location is polled, it can be determined by the responsible IM if the node has entered an intermediate zone (shown in Figure 2 as the intersection of the two management domains). If it has, and if it is rapidly moving towards the second zone, the top-level manager is informed which then initiates the handoff of the node to the second IM. This information about rapid movement can be determined from the previous and current position of the node.

The power of SHAMAN is evident from this location management example. Delegation is achieved by downloading the scripts to the IM and letting the IM perform the location tracking task; the concept of MIB views is demonstrated using the polling cells where only the relevant variables are selected for monitoring from the agents; event hierarchy is demonstrated using the dependencies between the polling, history and status cells. Many features of SSL are illustrated using the scripts contained in the cells.

# References


[1] Germán Goldszmidt. *Distributed Management by Delegation*. PhD thesis, Columbia University, New York, NY, 1995.

[2] D. Levi and J. Schoenwaelder. *Definitions of Managed Objects for the Delegation of Management Scripts (RFC 2592)*, May 1999.

[3] A. S. Sethi and P. Kalyanasundaram. A Spreadsheet Paradigm for Network Management. In *Proceedings of the 5th IFIP/IEEE Workshop on Distributed Systems: Operations and Management*, Toulouse, France, October 1994.

[4] P. Kalyanasundaram, A.S. Sethi, C. Sherwin, and D. Zhu. A Spreadsheet-Based Scripting Environment for SNMP. In A. Lazar, R. Saracco, and R. Stadler, editors, *Integrated Network Management V*, pages 752–765. Chapman & Hall, London, 1997.

[5] P. Kalyanasundaram. *Proxy Architectures for Hierarchical Scripting Frameworks and Interoperability of Management Models*. PhD thesis, University of Delaware, Newark, DE, 1998.

[6] SHAMAN web page URL http://www.cis.udel.edu/∼shaman.

[7] UCD SNMP Agent web page URL http://www.ece.ucdavis.edu/ucd-snmp.