

# Debugging Network Management Scripting Applications<sup>1</sup>

D. Zhu, A.S. Sethi

Department of Computer and Information Sciences

University of Delaware, Newark, DE 19716

USA

Email: {dzhu, sethi}@cis.udel.edu

## Abstract

A *network management scripting application* consists of a set of distributed and possibly mobile *delegated management scripts* [1, 2], as well as other non-script network management processes, which cooperate with each other to accomplish a given network management task. Such applications require debugging support: firstly, they are distributed applications, and distributed applications are notoriously difficult to debug. More importantly, the network management and the scripting environment exhibit many peculiar characteristics which necessitate special debugging support. The overall issue addressed by this paper is: *how can we effectively debug distributed and mobile network management scripting applications?*

Race conditions and non-deterministic behavior in scripting applications make the *cyclical debugging* technique not directly applicable. *Record-replay debugging* solves this problem by deterministically “replaying” an execution using the information gathered in the initial execution. Of the two variations of this approach, *simulation replay* and *instant replay*, we found the former to be more suitable to our environment because it does not need the full participation and control of all the processes in the application. Another very powerful technique, *global property-based predicates*, allows questions to be asked about the distributed computations based on local and global states to help understanding the computations.

One difficult problem that arises in replay due to a peculiar characteristic of network management scripting applications is that it is not always possible to *synchronize* the replay of the scripts. This happens when a synchronization is through a non-script process because of which we are unable to observe and replay it. Possible approaches to this intrinsic problem include simply ignoring the problem by not providing replay synchronization, introducing physical clocks which is not an easy task by itself, and putting the burden on the user

---

<sup>1</sup>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. This work was also supported in part by University of Delaware 1998-1999 academic year University Competitive Fellowship.

by asking the user to specify how the scripts should synchronize in the replay.

Starting to run the debugger after the application has been started is called *attaching* the debugger to the application. On-the-fly debugger attachment is especially useful for network management applications, which, very often, consist of many ongoing processes that run for a long time. For such management applications, starting the record phase from the beginning of the application would result in too many recorded events which may not be very useful in the initial phase of the debugging. Also, when we do replay, the suspected error may happen long after the replay is started. Attachment can also be useful for monitoring which is very important for continuously running applications.

Debugging mobile scripts poses even more problems than debugging distributed management scripts. During the lifetime of a mobile script, it interacts with many mobile script execution environments at different locations, which makes the script execution more difficult to monitor and control, and the execution traces more difficult to track. How to do record-replay in the mobile script context and how to record-replay a subset of scripting framework agent nodes of interest are all challenging issues.

In the paper we have also presented a simple integrated simulation-replay debugger and explained its usage. Currently we are trying to synthesize various debugging approaches and techniques in order to arrive at a comprehensive solution. A prototypical debugger design has started which binds our results to SHAMAN [3], a spreadsheet-based integrated scripting framework for SNMP.

### Keywords

Management by delegation, delegated scripts, scripting applications, debugging management scripts

## References

- [1] Y. Yemini, G. Goldszmidt, and S. Yemini. Network management by delegation. In *Integrated Network Management II*, I. Krishnan and W. Zimmer (editors), North Holland, Amsterdam, 1991, pages 95–107.
- [2] D. Zhu, A. Sethi, and P. Kalyanasundaram. Towards integrated network management scripting frameworks. In *Proceedings of the 9th Annual IFIP/IEEE International Workshop on Distributed Systems: Operations and Management*, Newark, Delaware, USA, October 1998, pages 233–246.
- [3] P. Kalyanasundaram, A. Sethi, C. Sherwin, and D. Zhu. A spreadsheet-based scripting environment for SNMP. In *Integrated Network Management V*, A. Lazar, R. Saracco, and R. Stadler (editors), Chapman & Hall, London, 1997, pages 752–765.