Dependable-LQNS: A Performability Modeling Tool for Layered Systems

Olivia Das and C. Murray Woodside
Department of Systems and Computer Engineering,
Carleton University, Ottawa, Canada
Email: odas@sce.carleton.ca, cmw@sce.carleton.ca

***Abstract*- Dependable-LQNS is a software tool for modeling and evaluating performability of fault-tolerant layered distributed applications that use a separate architecture for failure detection and reconfiguration. It takes into account the effects of management architecture, application software architecture, failure of management and application components in the dependability computation. It uses a combination of minpath algorithms, AND-OR graphs, non-coherent fault trees and Layered Queueing modeling in the analysis.**

I. INTRODUCTION

Dependable-LQNS models the class of applications that has layered software architecture with a general fault management architecture, as considered in [1]. The computation strategy of this tool has the following steps:

- find the different reachable operational configurations of the application
- compute the probability of each operational configuration (using a non-coherent fault tree)
- determine the reward rate for each operational configuration (using a Layered Queueing Network model)
- combine the probabilities and the rewards to determine the average performance measures.

This tool supports an architecture evaluation language that allows a user to:

(i) describe the application and management components (processors, application software tasks, manager tasks, agent tasks) and pseudo-components representing additional shared factors of failures (such as power-supply failure, failure due to mechanical vibrations).

(ii) specify the interactions between the components in the management architecture (as for example, monitoring, notification).

(iii) specify the functional dependencies among application components from layer to layer.

(iv) performance and dependability parameters (for example, execution demand of a task on its host, number of service requests made by one task to the other from layer to layer, failure and repair rates for the components and the redundancy mechanisms employed).

The flow-chart of the tool is given in Figure 1. The tool uses the help of Aralia[2] for non-coherent fault-tree computation and LQNS[3] for performance analysis. The tool has been written in C++ for Windows.

The input to the tool can be a plain-text file containing the above descriptions or there is also a graphical user interface for this tool developed with Java that takes inputs from the user and automatically generates the input file for the back-end compiler. Figure 2 shows a snapshot of the GUI.

The output of the tool consists of the different operational configurations, their probabilities, their

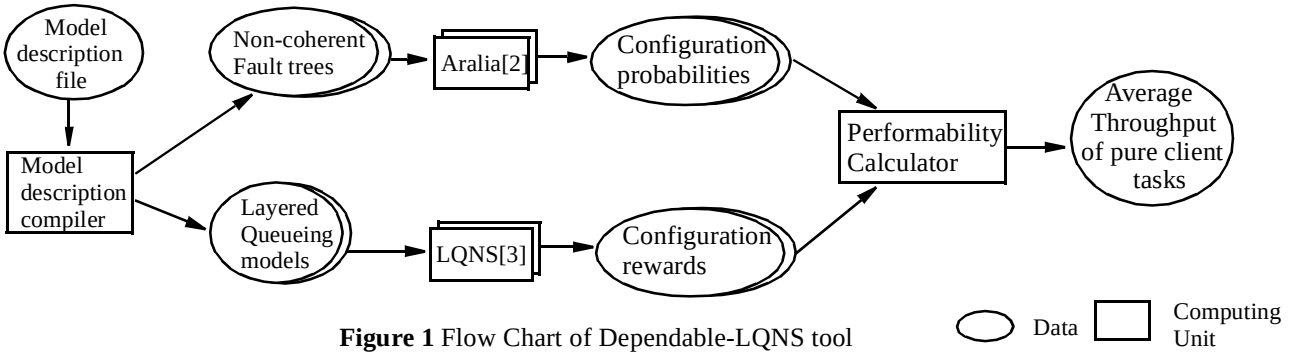

Figure 1 Flow Chart of Dependable-LQNS tool

associated reward rates and the average reward for the whole application. Presently, the mean throughput of the pure client task is taken as the reward rate for a particular configuration.

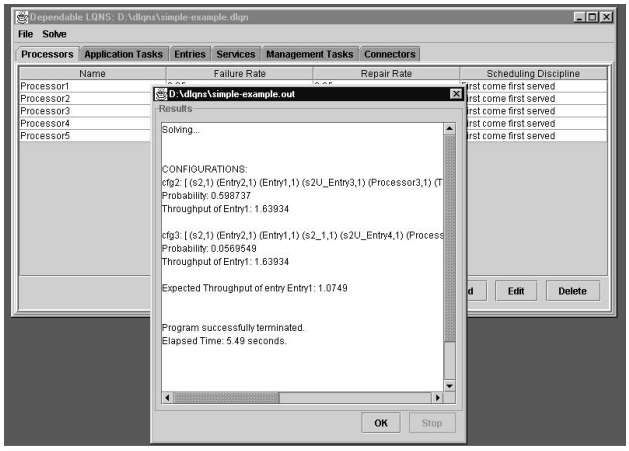

Figure 2 GUI snapshot of Dependable-LQNS tool

ACKNOWLEDGMENTS

This work was supported by a scholarship from Nortel Networks, and by the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

[1] O. Das and C. M. Woodside, "Modeling the Coverage and Effectiveness of Fault-Management Architectures in Layered Distributed Systems", IEEE DSN 2002, Bethesda, Maryland, June 2002, pp. 745-754.

[2] Y. Dutuit and A. Rauzy, "Exact and Truncated Computations of Prime Implicants of Coherent and non-Coherent Fault Trees within Aralia", Reliability Engineering and System Safety, 58, 1997, pp. 127-144.

[3] G. Franks, S. Majumdar, J. Neilson, D. Petriu, J. Rolia, and M. Woodside, "Performance Analysis of Distributed Server Systems", in the Sixth International Conference on Software Quality (6ICSQ), Ottawa, Ontario, 1996, pp. 15-26.