ABSTRACT

A distributed system is a complex system. Developing complex systems is a demanding task when attempting to achieve functional and non-functional properties such as synchronization, communication, fault tolerance. These properties impose immense complexities on the design, development, and implementation of the system that incur massive effort and cost. Therefore, it is vital to ensure that the system must satisfy the functional and non-functional properties. Once a distributed system is developed, it is very difficult and demanding to conduct any modification in its architecture. As a result, the quantitative analysis of a complex distributed system at the early stage of the development process is always an essential and intricate endeavor. To meet the above challenge, this chapter introduces an extensive framework for performability evaluation of a distributed system. The goal of the performability modeling framework is to consider the behavioral change of the system components due to failures. This reveals how such behavioral changes affect the system performance.
INTRODUCTION

Conducting performance evaluation of a distributed system separately from the dependability evaluation fails to assess the anticipated system performance in the presence of system components failure and recovery. System dynamics is affected by any state changes of the system components due to failure and recovery. This chapter introduces the concept of performability that considers the behavioral change of the system components due to failures and also reveals how this behavioral change affects the system performance. But to design a composite model for a distributed system, perfect modeling of the overall system behavior is essential and sometimes very unwieldy. A distributed system behavior is normally realized by the several objects that are physically disseminated. The overall system behavior is maintained by the partial behavior of the distributed objects of the system (Khan & Heegaard, 2011). So, it is essential to model the distributed objects behavior perfectly for appropriate demonstration of the system dynamics and to conduct the performability evaluation (Khan & Heegaard, 2011). Hence, we adopt Unified Modeling Language (UML) collaboration, state machine, deployment, and activity-oriented approach as UML is the most commonly used specification language which models both the system requirements and qualitative behavior through an assortment of notations (OMG, 2017; Khan & Heegaard, 2011). Our way of utilizing UML collaboration and activity diagram to capture the system dynamics, provides the opportunity to reuse the software components. The specifications of collaboration are given as coherent, self-contained building blocks (Khan & Heegaard, 2011). Reusability of the software component is achieved by designing the collaborative building block, which is used as main specification unit in this work.

In order to guarantee the precise understanding and correctness of the model, the approach requires formal reasoning on the semantics of the language used and to maintain the consistency of the models’ specification. Temporal logic is a suitable option in this regard. In particular, the properties of super position supported by compositional temporal logic of action (cTLA) make it possible to describe systems from different viewpoints and by individual processes that are superimposed (Herrmann & Krumm, 2000). In this work, we focus on the cTLA that allows us formalizing the collaborative service specifications given by UML activities, deployment diagram and state machine (STM) model. By expressing collaborations as cTLA processes, we can ensure that a composed service maintains the properties of the individual collaborations it is comprised of. The semantic definition of collaboration, activity, deployment, and STM model in the form of temporal logic is implemented as a transformation tool which produces TLA\(^+\) modules. These modules may then be used as input for the temporal model checker (TLC) for syntactic analysis (Slåtten, 2007).
Delay-Range-Dependent Robust Stability for Uncertain Stochastic Neural Networks with Time-Varying Delays
[www.igi-global.com/article/delay-range-dependent-robust-stability/49131?camid=4v1a](www.igi-global.com/article/delay-range-dependent-robust-stability/49131?camid=4v1a)

Decision Making Approaches for Advanced Manufacturing Technology Evaluation and Selection
[www.igi-global.com/chapter/decision-making-approaches-advanced-manufacturing/64931?camid=4v1a](www.igi-global.com/chapter/decision-making-approaches-advanced-manufacturing/64931?camid=4v1a)