A SEMANTIC FRAMEWORK FOR SPECIFYING AND REASONING ABOUT COMPOSABLE DISTRIBUTED MIDDLEWARE SERVICES

Carolyn Talcott

SRI International, 333 Ravenswood Avenue, Menlo Park CA 94025SRI

clt@cs.stanford.edu

Abstract

Middleware such as Corba ORBs, DCOM, or Java Beans provide programming abstractions for interoperation of diverse components in an open distributed setting. More recent middleware initiatives focus on resource management, security, and dependability services supporting a wide range of coordination and Quality of Service (QoS) requirements and allowing applications programmers to focus on functionality and intrinsic performance issues. As computer systems become more and more complex and ubiquitous it is increasingly important to have robust and powerful middleware that can achieve a high degree of reliability and assurance where needed. For example, as networks become more programmable it will be essential to minimize failures not only by simulation and testing, but also by formal modeling and analysis at many levels of abstraction. For adaptability and extensibility middleware services themselves need to be built from component services. Middleware services for resource management such as scheduling, protocols providing security and reliability, load balancing and stream synchronization, execute concurrently with each other and with application activities and can therefore potentially interfere with each other in subtle ways.

The Two Level Actor Model (TLAM) is a mathematical framework designed for reasoning about the interaction and composition of resource management activities in open distributed systems, and their dynamic installation and modification. The TLAM is based on the actor model of object-based distributed computation. As the name suggests, in the TLAM, a system is composed of two kinds of actors, base-level actors and meta-level actors, distributed over a network of processing nodes. Base-level actors carry out application-level computation, while meta-level actors are part of the runtime system (middleware) that manages system resources and controls the runtime semantics of the

base level. The TLAM model has been applied to reasoning about services such as migration, distributed garbage collection, and QoS-based multimedia services.

The TLAM approach to modeling middleware components is to develop a family of specifications from different points of view and at different levels of abstraction and to establish entailment relations between specifications from different points of view. From a global point of view we specify the end-to-end service provided by a system in response to a request. This can be refined by expressing system wide properties in terms of abstract properties of the underlying network. From a local point of view we specify constraints on the behavior and distribution of a group of meta-actors. This local behavior point of view can be further refined by specifying algorithms for the actions of individual actors. The staging and refinement of specifications provides a form of modularity, scalability, and reusability. It reduces the task of implementation to that of implementing individual abstract behaviors or algorithms. Behavior level specifications can be used to guide or check implementations or even serve as executable prototypes.

In this talk, we will briefly review the TLAM model and methodology and discuss extensions to model communication and real-time services. We will also discuss the relation of the TLAM architecture to the architectures developed in several ongoing middleware initiatives, and show how the TLAM can be used to specify and analyze services in these platforms.