

A service-oriented modeling and simulation framework for rapid development of distributed applications

W.T. Tsai ^a, Chun Fan ^a, Yinong Chen ^{a,*}, Ray Paul ^b

^a *Computer Science and Engineering Department, Arizona State University, Tempe, AZ 85287-8809, USA*

^b *Department of Defense, Washington, DC, USA*

Available online 15 December 2005

Abstract

The paper presents a service-oriented distributed modeling and simulation framework that supports the development and evaluation of large scale distributed systems such as network-centric and system-of-systems applications. The distinct features of the framework include a modeling and specification language, dynamic model checking for completeness and consistency, automated code generation from the specification, simulation of different architectures with a template-based platform builder, service-oriented multi-agent simulation for easy re-configuration and re-composition, formal policy specification and enforcement for dynamic verification, and dynamic analyses for evaluation and monitoring. The framework and most of the tools reported in this paper have been implemented and applied in several industrial projects. This paper presents the design principles, user interfaces, its runtime infrastructure, and the experiment results on its performance. © 2005 Elsevier B.V. All rights reserved.

Keywords: Distributed simulation; Multi-agent simulation; Verification and validation; Policy-based computing; Service-oriented architecture

1. Introduction

Network-centric and system-of-systems (SoS) design techniques emerge as a main approach supporting large scale distributed applications. The modeling and simulation framework presented in this paper aims at providing comprehensive support to the development and evaluation of network-centric and SoS applications in all their development phases. In the specification phase, a scenario-based modeling and specification language is provided for users to model and specify the system. The specification is then model checked to determine if it is complete and consistent. Test cases to be used in testing phase are generated during the Completeness and Consistency (C&C) analyses. If the specification passes the verification of model checking, an automated code generation tool will convert the specification into the executable code for simulation. The

* Corresponding author. Tel.: +1 480 965 2769; fax: +1 480 965 2751.
E-mail address: yinong@asu.edu (Y. Chen).

scenario specification and the automated code generation deal with the high level composition based on existing atomic functions such as library functions or services in service repository that can be searched, discovered, and invocated. This level of code generation support automated re-configuration and re-composition. Then the executable code will be tested (validated) in the simulation environment using the test cases generated during the verification. The purposes of testing are to validate that the code meets the functional requirements and to measure the performance and dependability (reliability, availability, safety, security, etc.) of the system under simulation.

The modeling and simulation framework also allows dynamic policy specification [14], verification, validation, and enforcement via simulation, which can provide support for policy-based computing systems such as Global Information Grid (GIG) [7].

Another feature of the framework is its Service-Oriented Architecture (SOA) [21] and Web Services (WS) implementation, which supports the simulation of SOA applications, distributed applications, multi-agent-based applications, highly dependable applications with fault-tolerance structures [18] such as redundancy and re-configuration, and recomposable applications that can dynamically change their functionality and form new applications at runtime.

In the entire process, the scenario specification needs to be written manually and the required atomic functions are assumed to exist in library or repository. All the remaining steps can be performed using automated tools in real time and at runtime. Furthermore, tool support is also provided for writing the specification, including a graphic interface, powerful syntax checking, and semantic checking in terms the completeness and consistence of the specification [22–24]. In other words, the simulation framework represented in this paper supports rapid simulation based on the specification.

There are two classes of distributed simulation techniques: distributed analytic simulations and distributed virtual environments [5].

One of the most successful analytic simulation techniques is the parallel and distributed discrete-event simulation. Georgia Tech Time Warp (GTW) is an example of applying discrete-event simulation [6]. Object-oriented technique has been adopted in this class of simulations to improve the system structure and component reusability [4,29].

Current distributed discrete-event simulation is still in the stage of model-code-run, which means that the users have to create a model of the target system, program the simulation based on the model, and then run the simulation, as discussed in GALATEA [3]. Several centralized discrete-event simulation frameworks such as SimuLink[®] in MatLab[®] [20], Arena [1], and SimProcess [19], provide the capability of model-and-run [25], i.e. after a user builds the model for a target system, the simulation framework can run the simulation without a manual coding step, which saves a great deal of effort. Current distributed virtual environments such as DIS and HLA, as discussed below, also need a manual coding step before the simulation can be executed.

The most well-known examples of distributed virtual environments are IEEE DIS (Distributed Interactive Simulation) [8,9] and IEEE HLA (High Level Architecture) [10–12].

HLA evolved from DIS aiming primarily at training simulations and the ALSP (Aggregate Level Simulation Protocol). It includes non-runtime components to specify the object model used by the simulation federation, which includes a set of object types chosen to represent the real world, the level of detail at which the objects represent the world, and the key models and algorithms that are to be used.

HLA provides powerful rules and templates for users to specify the simulation-federates and their interoperations. However, it does not provide consistent modeling methodologies and tools to model the target's behaviors and to verify and validate the correctness of the simulation. It is the responsibility of the user to ensure that the models built meet the requirement and the specification of the target system.

Comparing to HLA, our modeling and simulation framework is an integrated framework that provides model-and-run support for distributed simulation. The framework also provides a full set of tools to perform rigorous verification and validation in every step of the simulation development.

The rest of the paper is organized as following: Section 2 presents an overview and the various user interfaces supported by the distributed simulation framework. Section 3 explains the runtime infrastructure that powers the framework; Section 4 gives an example to illustrate how to build a simulation on the framework; Section 5 presents the experiments results; Section 6 concludes the paper.

متن کامل مقاله

دریافت فوری ←

ISIArticles

مرجع مقالات تخصصی ایران

- ✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
- ✓ امکان دانلود نسخه ترجمه شده مقالات
- ✓ پذیرش سفارش ترجمه تخصصی
- ✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
- ✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
- ✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
- ✓ دانلود فوری مقاله پس از پرداخت آنلاین
- ✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات