

System Composition Strategies, Panel Introduction

Stephanie White, Panel Chair
Long Island University
E-mail: Stephanie.White@liu.edu

Abstract

This short paper is an introduction to System Composition Strategies, for the ECBS 2001 panel on this subject. Significant advances in system composition strategies and supporting infrastructure are needed before projected growth in system interoperability can take place. Without these advances, expensive failures will occur, including operational disasters in mission and safety critical systems. The goal of the panel is to identify ongoing research and promote new research in the area. This paper discusses projected growth in system interoperability, current capabilities to support this, and required capabilities for future design synthesis environments.

1. Projected growth in interoperability

The message in the January 2001 issue of IEEE Spectrum, "Always on, Living in a Networked World", is that almost every device in use will be computerized and will interact with other products through computers. Digital devices in a smart space will recognize a new device entering the space, and establish communication with it. Connected across the Internet, a system's environment may be worldwide.

Projected growth in system interoperability is significant in every application. Millions of networked sensors in the atmosphere, each the size of a mote of dust, will help predict weather, and thousands of networked sensors on bridges will help predict earthquakes. Distributed digital brake by wire systems in cars will have separate computers and actuators at each wheel, interacting with each other and with brake pedal sensors. Eventually interacting steering, suspension, and braking systems will interact with radar collision avoidance systems and lane following vision systems, and cars may receive signals that an airbag has inflated in a car in front of them, so that they can respond.

2. Current capabilities

Considerable progress has been made in developing infrastructure to support interoperability, but the capabilities are still primitive.

The Department of Defense Modeling and Simulation Organization High Level Architecture (HLA) consists of a common ontology for system conceptualization and exchange of information, standards for inter-system communication, middleware for handling system interoperation for new and legacy systems, and a library for storing and retrieving HLA models [1]. Simulations conforming to HLA, legacy systems wrapped for conformance, and live systems interact using the Run Time Infrastructure (RTI). These capabilities are a major step forward, but do not provide sufficient synthesis support during the design phase. For example, teams developing interoperating simulations find it difficult to evaluate each other's designs to determine whether they are consistent with their own.

Commercial industry has several component models, supported by infrastructure. These include DCOM (Microsoft's Distributed Component Object Model), CORBA (The Object Management Group (OMG) Common Object Request Broker Architecture), and EJB (Sun Microsystems' Enterprise JavaBeans). In each case the component models specify a means for components to publish their interfaces, send messages, and pass data. Perhaps the most likely future scenario (for domains where exchanges are low frequency and have high semantic content) is that there will be a variety of component models suited to different platforms or environments, integrated through XML, and transported over a variety of transport mechanisms [2]. As with HLA, there is little support in the commercial world for design synthesis.

3. Needed design synthesis technologies

Advanced system/software synthesis features and highly automated infrastructure are needed to evaluate designs, ensure consistency across components, and

facilitate the dynamic composition and re-composition of systems in case of failure. These synthesis technologies include: (i) simulation-based design, (ii) methods for systematic handling of the many goals, dependencies, and constraints involved in the design process, (iii) evaluation of multi-level, multi-component, hierarchically specified models, (iv) performance evaluation techniques, (v) evaluation and tradeoff of other system qualities such as maintainability, availability, and scalability, and (vi) frameworks for experimenting with alternative designs.

To capture information relevant to the design of a system, multiple-view models must explicitly represent the dependencies and constraints among the various views. Modeling and simulation infrastructure should accommodate different levels of granularity so that the developer can work efficiently to answer questions about the model.

During system synthesis, designers evaluate capability and perform tradeoffs among design properties such as performance, safety and reliability. Progress has been made in evaluating performance and reliability, but little progress has been made in developing other metrics such as those for evaluating maintainability, availability, and scalability [3]. Research in evaluating these system qualities is needed.

Synthesis environments should support detailed modeling, simulation, and analysis of the operational system working within its environment. Especially in high-assurance applications, software components must be carefully matched to the context of their use [4]. Documented disasters, such as the Ariane 5 Flight 501 and the Therac-25 x-ray therapy machine, where software was reused, demonstrate the need.

In summary, greater emphasis should be placed on research related to system composition strategies and infrastructure because of the expected growth in system interoperability.

References

- [1] Department of Defense Modeling and Simulation Organization (DMSO), "High Level Architecture (HLA)", URL: <http://www.dmsomil.com>.
- [2] Jon Hopkins "Component Primer", Communications of the ACM, October 2000, pp. 27 – 30.
- [3] John Hennessy, "The Future of Systems Research", IEEE Computer, August 1999, pp. 27-33.
- [4] Michael R. Lowry, "Component-Based Re-configurable Systems", IEEE Computer, April 1998, pp. 44-45.