
Zusammenfassung

Im Bereich Modellbildung und Simulation ist die Erstellung von Modellen üblicherweise mit einem hohen Entwicklungsaufwand verbunden. Es ist daher sinnvoll, vorhandene und geprüfte Teile eines komplexen Simulationsmodells (sog. Modellkomponenten) auch in weiteren Simulationsstudien erneut einzusetzen. Die Wiederverwendung von Modellkomponenten über die Grenzen eines einzelnen Simulationssystems hinweg wird jedoch kaum praktiziert, da die Kopplung verschiedener Simulatoren oft eine große technische Hürde darstellt. Bisherige Ansätze wie web-basierte Simulation und verschiedene plattformübergreifende Systeme versuchten zwar, die technischen Randbedingungen für eine Kopplung vorhandener Komponenten zu verbessern, konnten sich aber nicht durchsetzen, weil sie stark abhängig waren von den Werkzeugen eines bestimmten Herstellers. Den aktuellen Stand der Entwicklung auf dem Gebiet der verteilten Simulation verkörpert die Simulationsinfrastruktur „High Level Architecture“ (HLA). Doch selbst hier ist noch keine Interoperabilität verschiedener Realisierungen gegeben. Weiterhin sind die HLA-Mechanismen zur Selbstbeschreibung von Teilmodellen, zur Codierung benutzerdefinierter Daten und zur Durchführung kurzer Simulationsläufe nicht ausgereift. Deshalb ist ein neuer Ansatz notwendig, der die herstellerunabhängige und plattformübergreifende Simulation in verteilten Systemen unterstützt und somit die technischen Hindernisse auf dem Weg zur komponentenorientierten Simulation beseitigt.

Diese Arbeit stellt einen solchen Ansatz namens CoSim (Component Based Distributed Simulation) auf Basis der Kommunikationsinfrastruktur CORBA vor. Dazu entstand eine neue Art von Schnittstellenbeschreibungen für Modellkomponenten, die den Einsatz benutzerdefinierter Datentypen erlaubt. Um den Austausch von Simulationsdaten zu regeln, wurde ein Konzept für Zugangspunkte von Modellkomponenten aus Zeiglers Formalismus „Discrete Event System Specification“ (DEVS) übernommen. Zusätzlich berücksichtigt CoSim verschiedene Phasen der Modellnutzung und ermöglicht feingranulare Synchronisationsverfahren zwischen Modellkomponenten. Die Konzeption von Werkzeugen zum Aufbau von Modellbeschreibungen, zur Komponentenerstellung, Komponentenkopplung und zur Experimentdurchführung, die auf den Schnittstellenbeschreibungen aufbauen, vervollständigt den Gesamtentwurf. Der Ansatz dieser Arbeit ist ein modellunabhängiger Beitrag zur verteilten Simulation. Lösungen für spezielle Simulationsanwendungen (z.B. Simulation von Rechnernetzwerken oder Klimaberechnungen) zeigen eine deutlich bessere Performanz, lassen sich jedoch nur schwer für andere Szenarien adaptieren.

Um die Tragfähigkeit des CoSim-Ansatzes zu demonstrieren, wurde ein Beispielmodell realisiert, das die wesentlichen Leistungen des Gesamtsystems auch tatsächlich abfordert. So sind die Teile des Beispielmodells auf unterschiedlichen Plattformen in verschiedenen Programmiersprachen implementiert. Weiterhin verlangt die Synchronisation der Teilmodelle Maßnahmen zur Vermeidung von Verklemmungen. Dazu wurden alle konzipierten Werkzeuge so tief prototypisch realisiert, dass der Modellbildungs- und Simulationsprozess vollständig unterstützt wird. Damit sind Modellkomponenten als Objekte in einer verteilten Umgebung zugänglich und der Wiederverwendung dieser Objekte stehen keine technischen Hindernisse mehr im Wege.

Abstract

Creation of models for simulation usually entails considerable development efforts. Therefore, it is reasonable to reuse existing valid parts of complex simulation models (model components) in other simulation studies. But reuse is hardly ever seen across the boundaries of a single simulation system since the coupling of different simulators often incorporates large technical obstacles. Past approaches like web-based simulation and some cross-platform systems tried to improve the technical settings for the coupling of model components. None of these approaches were widely accepted as they are tied to products of particular vendors. The „High Level Architecture“ (HLA) is said to be state of the art in the area of distributed simulation, but does not achieve full interoperability between different implementations. Furthermore, HLA’s mechanisms for describing model parts, coding model specific data and executing multiple short simulation runs are not yet mature. A new approach is needed to support vendor independent, cross-platform simulation in distributed systems, so that the main technical obstacles towards component oriented simulation can be overcome.

This dissertation presents such an approach named CoSim (Component Based Distributed Simulation), which uses CORBA as the underlying communication infrastructure. A new kind of model interface descriptions permits the usage of model specific data structures. To control the exchange of simulation data the port concept of Zeigler’s formalism „Discrete Event System Specification“ (DEVS) was adopted. In addition, CoSim regards different phases of model usage and enables model components to execute fine grained synchronisation algorithms for causality assurance. The concept is rounded out by tools for creating model components and their descriptions, for coupling model components, and for executing simulation runs. This approach is a model independent contribution to distributed simulation. Solutions for special application domains of simulation (e.g. internetworking of hosts, or climate calculations) will show a much better performance, but can be hardly adapted to other simulation domains.

To demonstrate CoSim’s feasibility a model example was implemented, which claims most of the systems capabilities. The model parts are implemented on different platforms in different programming languages and demand a synchronisation scheme that avoids deadlocks. All tools are realized as deep vertical prototypes, so that the whole cycle of modeling and simulation is supported. Thus, model components become accessible as objects in a distributed environment and technical obstacles for the reuse of these objects are overcome.