An Efficient and Flexible Methodology for Modelling and Simulation of Heterogeneous Mechatronic Systems

Stefan Scherber, Christian Müller-Schloer
Institut für Rechnerstrukturen und Betriebssystem, Universität Hannover
Lange Laube 3, 30159 Hannover, Germany
email: {scherber,cms}@irb.uni-hannover.de

Abstract

The complexity of mechatronic systems increases continuously. The need of simulation to evaluate these systems in an early design stage is an evident fact.

This paper presents a new approach for simulation of heterogeneous mechatronic systems. The aim of the development was to combine openness and flexibility with a high simulation speed and accuracy. Additionally to the description of algorithms and methods, the implementation and an example application are discussed.

1. Introduction

The term „mechatronics“ means integration of electronics, information processing and mechanics. Mechatronic systems consist of sensorial, processing and actuating elements like measurement systems, engines, valves, controllers, ASICs and microprocessors.

Mechatronic system behaviour is determined by dependencies between different components. Therefore, an integrated and interdisciplinary engineering approach, which contains the various technical fields, is necessary. An analysis of the different subsystems with respect to capabilities, capacities, behaviour and interactions requires a simulation of the complete multitechnical system. For each of the different technologies specialized and proved Computer Aided Engineering (CAE) tools exist. However, simulators supporting all kinds of models are not available. Hence, unacceptable effort for the conversion of subsystem models is necessary.

An alternative is the use of a simulation backplane, which connects different CAE-tools during simulation. A disadvantage of this approach is that computing time, needed for the simulation of the complete system, increases. The main problems are the requirements on connected simulators and frequently a loss of accuracy. Many backplanes are not suited since only simulators with constant time steps [1] or only one analog simulator are allowed. However, analog simulators needed for a simulation of a mechatronic system model often use solving algorithms with variable time steps [2]. In many cases determination of the next event time [3] or the ability to save and restore simulation state is necessary [4].

In the following paper, we present a new concept for the simulation of heterogeneous mechatronic systems. The idea of this approach is based upon a priority analysis and scheduling. The concept allows variable degrees of integration of simulator modules: from a very fast function-based coupling to a very flexible backplane-like connection. In spite of a flexible and open interface for connecting different simulators, the execution of models is very fast and a high accuracy is obtained. There are only a few requirements for interconnecting a simulator, such that on the one hand a wide range of simulators are available, on the other hand the integration effort is minimal.

2. Modelling and simulation concepts

A system model consists of several concurrent *actors*. Each actor describes functional and timing behaviour of a part of the system, e.g. a software process or an interface device. The description form can be chosen without restrictions. In order to communicate with other actors or to make an internal state available an actor contains ports. A connection between ports is created via a channel. The basic idea of this communication concept is that actors are able to write or read onto a port without knowing source or target actors. Thus, each actor can be developed independently of its environment, so that interchangeability and reuse of actors is guaranteed.

In a second step the system model has to be translated into an executable system model. Therefore, an actor has to be controllable, i.e. an external controller is able to start and stop the execution. In general, a simulation module or a simulator executes the actor. Thus, a simulator executing an actor only has to reproduce functional and timing behaviour, to offer the possibility to an external port access and to simulate until a specified time limit.

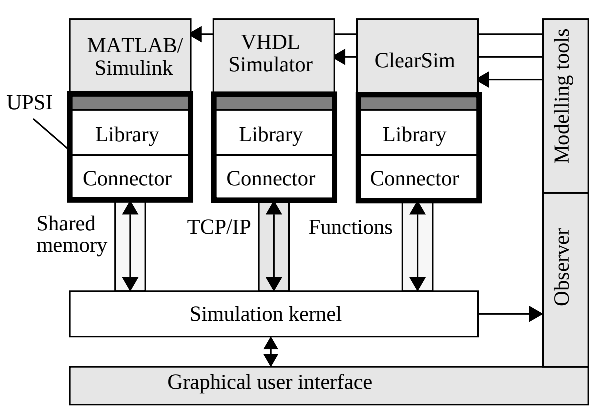
In order to determine maximal time of the next simulation interval a simulation backplane needs the global next event time. If this time is unavailable an optimistic approach can be used and with a rollback mechanism in the simulators a loss of simulation accuracy can be avoided. Without these features inaccuracies have to be accepted. Thus, existing simulator backplanes make a compromise between openness and accuracy. To overcome this limitation, in our approach a preprocessing step analyses simulator features. Together with the actor and channel structure including port types a *dependence graph* and for each actor a *priority* are calculated. With these values the scheduler determines optimal simulation order and maximal simulation intervals with respect to accuracy and simulation speed.

Recent studies [4], [5] have shown that communication and task switch time are the main reasons for a low execution speed in case of a simulation with a backplane. Most simulation backplanes use a socket connection on a TCP/IP network for communication, in order to enable a distributed simulation. By applying a communication via shared memory computing time is reduced decisively. A virtualization of the processing model allows to replace the kind of communication without changing program code. A further performance improvement is obtained by using threads instead of processes.

To avoid communication and task switch time completely a simulator can be connected functionally. In this case the source code of the simulator or a library for external access is necessary. By this method a program is created, which is comparable to one integrated system simulator. Nevertheless, the program contains a backplane with several standalone simulators. In other words, a combination of a simulator backplane and one integrated system simulator is realized.

3. Implementation and Results

The environment consists of three parts: the *simulation kernel*, which controls the complete simulation, the interface called *unified and portable simulation interface* (UPSI) for connecting simulation modules, and a graphical user interface for modelling and controlling. The figure below shows the architecture.



Architecture of the simulation environment.

The UPSI consists of a *connector*, which interlinks the interface with the kernel, and a library for connecting simulators. The library is available and equivalent for all communication and execution methods. Therefore, a method change, e.g. from a functional linking to a socket-based communication in order to perform a remote execution, requires only a new compiler link.

Currently the following simulators are connected to the unified and portable simulation interface:

Simulator	Integration
execution-driven software simulator ClearSim [6]	Thread / Shared memory
timed software emulator ClearEmu	Functional
EFSM-Simulator ClearEFSM	Functional
Modelica simulator ClearPhy [7]	Functional
MATLAB/Simulink	Functional (ActiveX)
visualisation program WinCC	Functional (Database)

As example application an electro-hydraulic multipurpose proportional servovalve was evaluated. The servovalve contains an integrated digital control electronic based upon a Siemens C167 microcontroller. The execution-driven software simulator ClearSim performed the simulation of the microcontroller kernel and the application software. The microcontroller on-chip peripherals were modelled and simulated with the EFSM-Simulator. To model mechanical and hydraulic parts of the servovalve the modelling language Modelica was used. Furthermore, this latter model was translated to a Simulink model, in order to compare the simulators.

With our new integration and simulation methods speed-up improvements up to 6 have been obtained in comparison with a socket-based communication.This result clearly indicates that simulation of a heterogeneous system can be speeded-up significantly by applying the priority scheduling and the combination of a simulation backplane and a system simulator.

The authors would like to thank Siemens AG for support of this work.

References

[1] J. Scholliers: *Development of an integrated environment for the simulation of multitechnical systems*. European Simulation Multiconference ‘94, Barcelona, Spain.

[2] F. Breitenecker, I. Husinsky: *Model exchange - illusion or future reality?* Eurosim ‘95.

[3] M. Zwolinski, C. Garagate, Z. Mrcarica, T.J. Kazmierski, A.D. Brown: *Anatomy of a simulation backplane*. IEE Proc.-Comput. Digit. Tech, Vol. 142, Nov. 1995

[4] M. Niemeyer: *Das Simulatorkopplungs-System SiCS*. ASIM ‘91.

[5] S. Scherber: *Beschreibung der Kommunikationsbibliothek und der Laufzeituntersuchungen der Kommunikationsmechanismen*. Technischer Bericht Sie-98-02-003.

[6] J. Spitzkowsky, C. Müller-Schloer: *An execution-driven real time simulator for embedded control systems*. European Simulation Multiconference ‘96, Budapest, Hungary.

[7] H. Elmqvist, S.E. Mattson: *Modelica - The next generation modelling language. An international design effort*. World Congress on System Simulation (WCSS‘97), Singapore.