Proceedings of the 2012 Winter Simulation Conference
C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, eds.

INTEGRATING OBJECT ORIENTED PETRI NETS INTO THE ACTIVE GRAPH DATABASE OF A REAL TIME SIMULATION SYSTEM

Juergen Rossmann
Michael Schluse
Ralf Waspe

RWTH Aachen University
Institute for Man-Machine Interaction
Ahornstrasse 55
D-52074 Aachen, GERMANY

1 INTRODUCTION

Most modern interactive 3D simulation systems are at their core quasi continuous. However, user interaction is mostly based on discrete events, like the push of a button. Furthermore, many algorithms can ideally be described in a state oriented or discrete event approach. In addition to this, when interfacing physical hardware in robotics and automation, the state of real machines is monitored and controlled via I/O boards, which are inherently discrete. We introduced the Petri-net based State Oriented Modeling Language (SOML++) in (Schluse 2002) and (Rossmann, Schluse, and Waspe 2008), which provide a common basis for the integration of state oriented or discrete event based ”processes” into 3D simulation systems.

2 THE REAL-TIME SIMULATION DATABASE

We developed an architecture for our 3D simulation system, based on a small kernel, an object-oriented real-time graph database (see Gyssens et al. 1994), containing all the data and algorithms needed for simulation applications. It provides the central building blocks for data management, meta information, communication, persistence and user interface. In addition to ”build-in” nodes, the meta information system allows to generate new node types during runtime (for example for object oriented scripting or new data models). For real-world applications the core must be extended by new data schemes and simulation algorithms like 3D simulation, State Oriented Modeling or GUI programming.

3 STATE ORIENTED MODELING LANGUAGE

SOML++ is a language that describes objects that can contain Petri-nets (see Bastide 1995). For ease of use the syntax of SOML++ is closely related to that of C++. In an object-oriented fashion these objects can be derived from other objects and can encapsulate data. In addition to ”classical” petri-net features like places, transitions, arrows and tokens an object can also contain functions.

For the representation of the SOML++ code within the simulation database, a new data schema has been developed. When a piece of SOML++ code is loaded by the simulation system, equivalent nodes are created for classes, objects, arrows, etc. Contained places and transitions are sub nodes of the object nodes, arrows are modeled as sub nodes of the originating place and transition nodes. Since every SOML++ object has now become a ”normal” database node, net functions can use all the functionality provided by the database or the meta information system, respectively. Net functions can interact with the rest of the database by creating new instances (using all classes known to the simulation system, not only those defined in the SOML++ code), obtain references to other database nodes and call functions of these nodes.

978-1-4673-4781-5/12/$31.00 ©2012 IEEE

As show in Figure 1 our simulation system also provides are comprehensive set of SOML++ debugging tools. All tokens can be traced and it is possible to set breakpoints within functions.

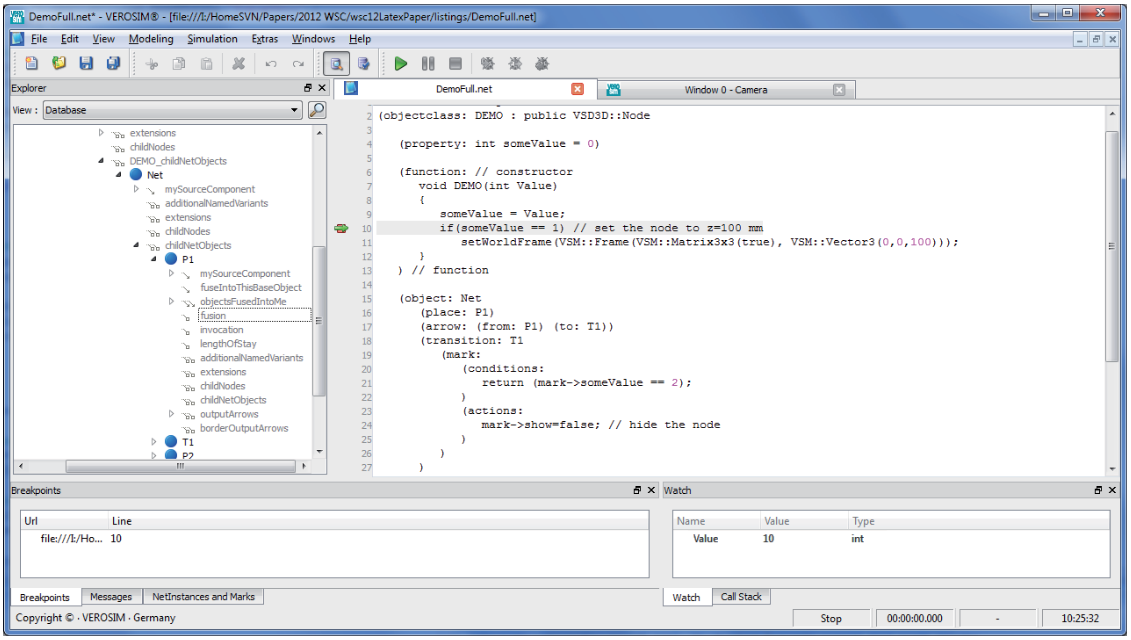

Figure 1: Debugging SOML++.

4 CONCLUSION

Due to its meta information management and the structure of the object oriented graph database our simulation system can adapt to new data schemes even at run time of the simulation, without the need for further programming. This approach allows us to integrate Petri-net objects as described in the State Oriented Modeling language. Petri-net objects become an integral part of the simulation database and enables supervisory control of quasi continuous simulation applications or discrete event simulations. The result is a hybrid simulation system which has proven its applicability in large variety of applications, ”classical” simulation applications like driving or production simulators but also new fields of applications like GUI modeling or Virtual Testbeds.

REFERENCES

Bastide, R. 1995. “Approaches in unifying Petri nets and the Object-Oriented Approach”. In *In Proceedings of the Application and Theory of Petri Nets*.

Gyssens, M., J. Paredaens, J. van den Bussche, and D. van Gucht. 1994. “A graph-oriented object database model”. *Knowledge and Data Engineering, IEEE Transactions on* 6:572 –586.

Rossmann, J., M. Schluse, and R. Waspe. 2008. “3D Discrete Event Systems: An Efficient Way to Model and Supervise Dynamic Behavior in Virtual Environments”. In *ASME 2008 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2008)*, Volume 2008, 1503–1511: ASME.

Schluse, M. 2002. *Zustandsorientierte Modellierung in Virtueller Realitt und Kollisionsvermeidung*. Ph. D. thesis, Institue Of Robotics Research, University Of Dortmund, Germany.