ilmedia

TECHNISCHE
UNIVERSITÄT
ILMENAU

Raab, Michael; Schulze, Thomas; Straßburger, Steffen :

Experiences from the application of HLA-based distributed simulations in the production of vehicles

EXPERIENCES FROM THE APPLICATION OF HLA-BASED DISTRIBUTED SIMULATIONS IN THE PRODUCTION OF VEHICLES

Michael Raab[a], Thomas Schulze[b], Steffen Straßburger[c]

[a]Fraunhofer Institute IFF, Magdeburg, Germany
[b]University of Magdeburg, Germany
[c]Technical University of Ilmenau, Germany

[a]Michael.Raab@iff.fraunhofer.de, [b]tom@iti.cs.uni-magdeburg.de, [c]Steffen.Strassburger@tu-ilmenau.de

ABSTRACT
This article discusses the application of distributed simulation in the context of vehicle production planning. The experiences are derived from a real industrial project which aimed at connecting up to seven individually developed simulation models. The article reports on lessons learned which include the need for efficient ways to manage and control HLA-based distributed federations, to maintain a single code base for the models as well as lookahead considerations for synchronization.

Keywords: distributed simulation, production planning, vehicle production

1. INTRODUCTION

Distributed Simulation refers in general to the concurrent execution of several simulation models on one or multiple processors (Fujimoto 2000). Resulting benefits are possibilities to reuse existing simulation models in an expanded context as well as to couple simulations models of different software manufacturers.

In factory design and furthermore in factory operation simulation models of different factory sub sections are often developed independently and with differences in level of detail and utilization focus. These models are usually well suited to analyze the respective factory sub sections. To model the interdependencies between the several sub sections, a complex model is needed. Dependencies of sub sections are mostly influenced by existing connection modules like storages or transport operations.

How can such a complex model be created? The classical approach would be the development of a new monolithic overall model, often with a level of detail which is lesser than the one provided by existing sub models. A different possibility (which is suggested in this contribution) is the coupling of the pre-existing simulation models establishing a complex distributed model while keeping the achieved level of detail. This approach has the following advantages yielding mainly from the reuse of the existing simulation models (Strassburger et al. 2006, Rabe und Jäkel 2001):

- Saving model development time in respect to the development of a new monolithic overall model
- Assuring the obtained level of detail
- Ensuring consistence between sub models and distributed overall model

The paper describes authors' experiences from an industrial project targeting HLA-based distributed simulation in the utility vehicle sector production. The following chapter explains very shortly basic tasks of distributed simulation and characterizes the usability of distributed simulation in industrial applications. Subsequently the starting position for this project and the derived requirements will be explained. Some implementation details are commented in the following chapter. Calculated results from the complex distributed simulation model are presented in the subsequent chapter. An outlook finishes the paper.

2. BASISC TASK INSIDE DISTRIBUTED SIMULATION

To achieve the needed interoperability between the pre-existing simulation models, there are three main tasks derived from the distributed simulation paradigm which must be fulfilled (Fujimoto 2000). At first an agreement on common objects of interest inside all simulation models has to be created. All participants have to use the same interpretation of sent and received data. This agreement has to consider the semantic interoperability between all components of the distributed model. The second task is to exchange data based on common objects which are defined in the agreement. Typically publish and subscribe mechanisms are applied. Synchronization of simulation time inside components is the third main task. In general every component has its own simulation clock and has to react to external events from other components.

The existing approaches can be divided into two categories. The first category includes solutions which couple only homogenous simulation models from one simulation model family. In this style, different models within one simulator (e.g. Plant Simulation) can be coupled based on internal features of that simulator.

This is a rather inflexible solution as it is bound to the proprietary capabilities of that simulator.

More flexibility is provided by solutions from the second category. Theses solutions are independent from a single simulation software manufacturer and are based on a framework or standard providing interoperability between the models. They are not bounded on proprietary interfaces. Known approaches are for example FAMAS (Boer 2005) and HLA (IEEE 1516-2000). The HLA-approach is a world-wide used and standardized architecture for distributed simulation, which allows to couple simulation and non-simulation components with different time advance mechanisms. Based on rather good user-acceptance, high flexibility and our own experiences with HLA we decided to use HLA for an industrial application.

3. STARTING POSITION

Within design and construction of a vehicle production factory, independent simulation models were developed to simulate the behaviour of seven interrelated assembly and paint sections. These models are used after the design phase to support ongoing factory operations. The material flow inside the factory is presented in figure 1. The aim of our industrial project was to connect the pre-existing sub section models of a factory and to integrate them into a complex distributed simulation model. This complex model should be used to investigate the interrelationships between the different sub sections. Therefore the dimensioning of buffers between sections and the coordination of production schedules were investigated in order to assess bottlenecks in the complex system. In our experience this scenario of having to couple different independently developed simulation models represents a typical use case in industrial applications.
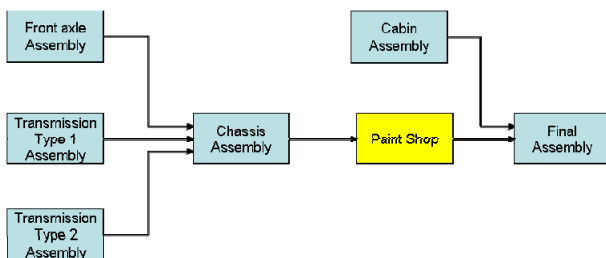

Figure 1: Schematic View on the Production System

Differences which typically need to be solved concern the implementation of the simulation models in varying simulation software, levels of detail in the models, time advance mechanisms and time dependencies. Our models can be divided into two application categories: assembly and paint shop models. Each category uses its own generic SLX simulation model, which is used in combination with an appropriate configuration file to generate an executable SLX simulation model suitable for one production area. Due to existence of a generic HLA interface for the SLX simulator, HLA with RTI 1.3NG was selected to perform as middleware for distributed simulation (Straßburger 2001).

4. REQUIEREMENTS

To use a complex distributed model in this context following requirements need to be fulfilled:

- Unique model version both monolithic and distributed for using legacy simulation models,
- Simple management of distributed simulation tasks,
- Presentation of additional results related to the context of distributes simulation and
- Observance of simulation run-time.

An important requirement within the presented industrial project was the need to maintain a single code base for the monolithic and the distributed model versions of the models. This was a mandate to keep the models consistent as they are used in daily operations. The configuration of the models had to allow the flexibility to run a model either in stand-alone mode or connected to other models. The distributed model version has to contain additional features which allow for example reactions on external events, synchronisation with other distributed models and modelling of transport to the successor model. These additional features have to be hidden.

The HLA-runtime environment offers services of a very low level for managing basic tasks. For example, a distributed simulation model has to be started, has to join and resign the federation, and has to be finished in well defined way. Within our project scope it was advisable to develop a dedicated user interface that coordinates the necessary services for these tasks and hides them from the user.

One of the main project goals was to obtain new insights from the distributed complex model, which can't be obtained from single monolithic models. New features for collecting information and statistics between the models and their presentation in user-friendly environment had to be developed.

The run-time of distributed simulation models is strongly influenced by size of lookahead and used synchronisation methods. 100 days is a typically value for simulation run inside this industrial application area. Monolithic simulation models based on SLX execute these simulations in some seconds of run-time. The project has to guarantee run-times for distributed simulation which are in acceptable time slots.

5. IMPLEMENTATION

From the set of necessary implementation tasks, the customization of the generic models, the modelling of the material flow, the influence of the lookahead and the development of a user interface were chosen to be explained in more detail.

5.1. Customization of legacy generic simulation models

There were two different types of monolithic legacy generic simulation models. One for simulation of assembly processes and the other for simulation of painting processes. Both monolithic model types differ in modelling approaches; they are specialized for their application area. But they are unique in their implementation language SLX and generic concept. The generic models starts and will be adapted by using configuration files to a specific simulation model. Both legacy generic simulation models have to be extended through:

- HLA-services for joining and resigning a federation as soon as sending and receiving of object attributes and interactions,
- Integration of external events into their local event list and responding on this external events and
- Modelling of part movement between coupled models.

The requirement was the customization of the pre-existing generic monolithic simulation models to be able to participate in the distributed simulation. A basis condition was to avoid the development of different model versions for monolithic and distributed applications. The generic model should be able to generate both a monolithic and a distributed version of the simulation model to ensure the consistency between both models. The simulation language SLX offers the possibility to extend the source code during compilation. By using this SLX feature different model versions can be generated smartly. Figure 2 shows the approach to customize the pre-existing generic models.
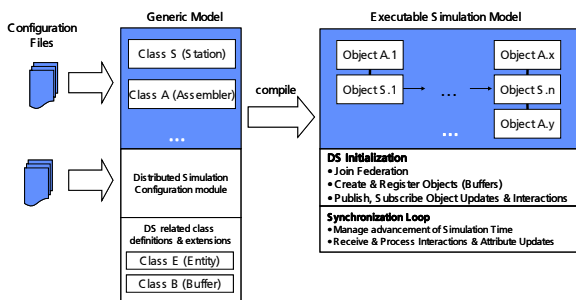

Figure 2: Customizing pre-existing generic simulation models for use in distributed simulations

Monolithic simulation models usually destroy their flowing material parts when they reach the end of the modeled process chain. In a distributed simulation it becomes necessary to hand over subsets of these parts to successive models. The transfer conditions of our industrial pilot project include the transfer of material to a bounded buffer of a successive model. In the distributed version the same simulation model has to pass the material part to next model. Enhancements to the simulation code had to be made to allow this behaviour.

Regarding their interoperability requirements our distributed simulation models are therefore compliant with a Type A.2 Interoperability Reference Model. The classification used here is derived from the Draft Standard for Commercial-Off-The-Shelf Simulation Package Interoperability Reference Models (Taylor, S. 2007) The A.2 reference model type describes the synchronized transfer of entities into a buffer with limited capacity. In our solution implementing this transfer and its conditions, we use an HLA interaction for the entity transfer and HLA attribute updates to communicate the current buffer content of the respective input buffers. Figure 3 shows a principle view and some important data structures.
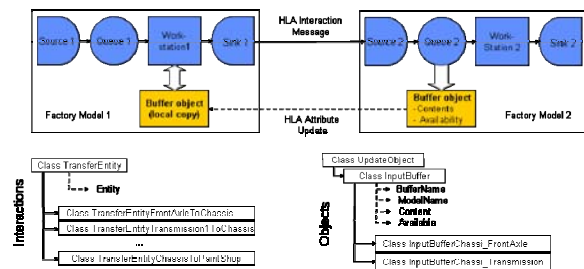

Figure 3: Modelling the Material and Information Flow beyond the Borders

For allowing the operation in a distributed simulation certain data about local objects and its attributes (which are of interest to the other models) has to be communicated regularly to other simulation models.

The Factory Model 1 in Figure 3 has to know the availability of the input buffer (Queue 2) in Factory Model2. Only if this buffer is available Factory Model 1 can transfer a part to Factory Model 2. The buffer object is published to all other models with the attributes Contents and Availability. Factory Model 2 publishes all changes of the attributes to other simulation models. A local copy of the buffer from Factory Model 2 is used inside Factory Model 1. If an attribute update from the original buffer is sent then the local copy will be updated automatically. So consistency between original and copy is ensured and Factory Model 1 decides about part transfer depending on the local copy of the buffer in Factory Model 2. For the publication and subscription as well as the sending and receiving of such data certain HLA functionalities have to be used. This behaviour has to be implemented, in the best case in a manner transparent to the end user.

5.2. Influence of lookahead

In distributed simulation the lookahead describes a period of time in which a sub model will not create external events for other sub models. Using a lookahead greater zero conservative synchronisation methods can calculate a time limit for other sub models. In general an increasing lookahead enforces parallelism of sub model execution and leads to decreasing execution time of distributed simulation models. Results of

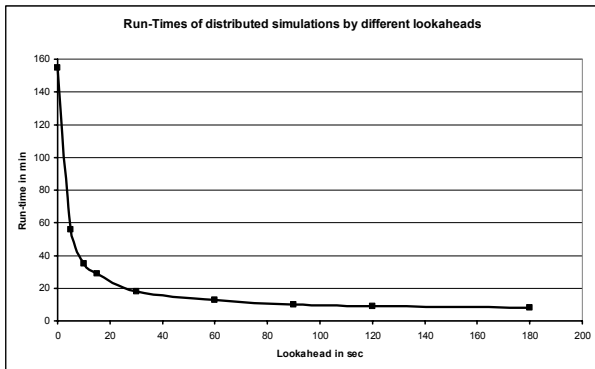experiments with different sizes of lookahead are shown in Figure 4.

Figure 4: Run-Times of Distributed Simulations by Different Lookaheads

To use a large lookahead is useful from the point of performance. But a lookahaed > 0 is influencing the logic inside execution of events. This is shown using one example (Compare Figure 3). Factory Model 1 can not deliver parts if the buffer of Factory Model 2 is not available. Assume that the buffer attribute Availability has been updated inside Factory Model 2 at time t1. The other models will be notified about the new value will at a timestamp t2 = t1 + lookahead. The value of the local copy of the buffer object inside Factory Model 1 will therefore not be updated until time t2. As a result the logic inside Factory Model 2 uses an incorrect value about availability until time t2. This can lead to undesired reactions inside Factory Model 2. For example, if the buffer is not available and the Factory Model 2 doesn't know this fact and then factory Model 2 could send a part into the full (not available) input buffer.

None general solutions for this problem have been found inside this project. Special application-depending approaches were used to avoid such problems.

5.3. User-Interface
For real industrial usage of distributed simulation it is necessary to support the end user in his operational work with the distributed simulation. The existing features and user interfaces of commercial simulators do not support the work with models in distributed environments. During the implementation of the industrial pilot project (which included up to seven individual models) it became very soon clear that the existing manual control features for starting, closing and controlling the distributed simulation were insufficient. For this reason the development of a user-interface (called commander) was initiated.

The version of the user-interface (called commander) currently developed provides the following services:

- Encapsulation of RTI-Services against the user
- Supporting the user in creating the composition of the complex model

- Creation of model visualization and calculation of simulation statistic
- Starting, controlling and closing down entire executions of the distributed simulation

The commander provides the opportunity to compose and manage distributed simulation models on an abstract layer without having the need to access the HLA API. Such a layer is mandatory to achieve high acceptance of distributed simulation in operational environments. The commander was implemented in C++ to be able to access the RTI-services provided by the HLA API. To store the commander configuration, an XML-file is used. Figure 5 shows an example of the Commander user interface.
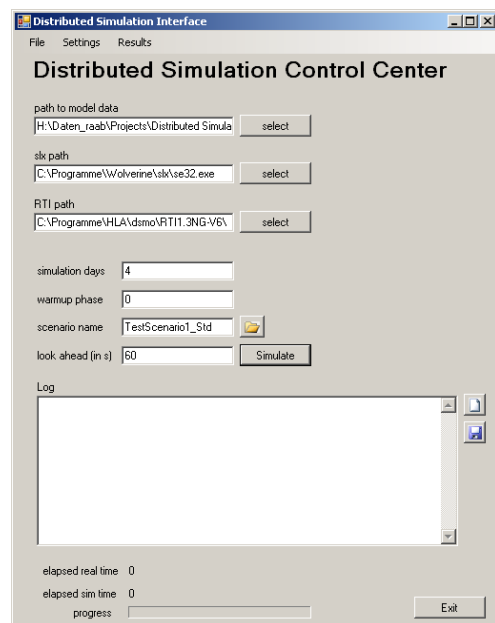

Figure 5: Form for the User-Interface

6. RESULTS
Besides the standard numerical result preparation within the pre-existing monolithic models, an "Observer-Federate" was used to collect data regarding the distributed simulation including the flow of material between the connected models and the history of buffer content changes. These data can be collected from the regular communication traffic between the models. For further characterization of dependencies between the different factory sections, each model collects additional statistics on waiting times due to full buffers and waiting times due to missing materials. This information relieves the search for bottlenecks in the complex system.

Table 1 shows results from model M4 for waiting times. Models M1, M2, M3 and M4 deliver parts to model M4. The production in model M4 has to stop, if needed parts from provider models are not available. Statistic for the duration of this models state is gathered.

Delays due missing incoming sub-models			
Model	Provider Models		
M4	M1	M2	M3
# of delays	0	10	4
Minimum delay	0s	30m	4m
Average delay	0s	3h 30m	45m
Maximum delay	0s	7h 34m	2h23m
Sum delay	0s	17h 30m	4h34m

Table 1: Delay Times for Model M4 Due Missing Incoming Parts from Provider Models

Figure 6 shows an example about demand and simulated production output of a sub model. The sub model depends on other sub models which deliver parts. The production output, depending on the behaviour of provider models can be calculated only in a complex model.
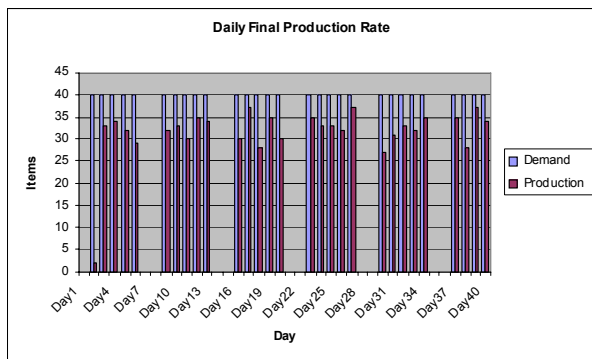

Figure 6: Daily Demand and Simulated Production Output for a sub model

For each of the monolithic simulation models in our industrial project an individual visualization model had been created. These models were inappropriate for the visualization of the distributed simulation, as for this use case only the material flow and the buffer fill levels *between* the individual models were of interest (i.e., the internal actions *within* a model were not to be visualized). Therefore a new visualization including statistic features for the interesting outputs had to be created (compare Figure 7).
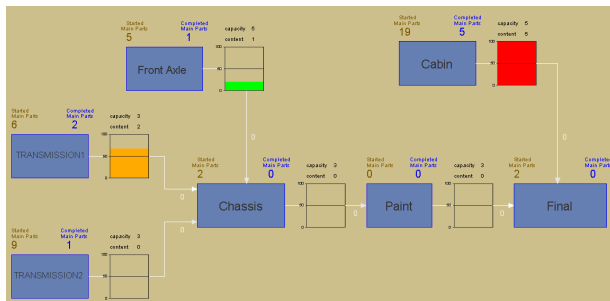

Figure 7: Visualisation of Material Flow and Buffer Contents

7. FUTURE WORK
The goal of this industrial project to use legacy monolithic simulation models in a distributed simulation models has been achieved. The results that have been derived from the complex distributed model are relevant for project members. This project shows that distributed simulation in industrial fields is necessary, possible and manageable.

Additional work will be done in two directions: The development of more general solutions for the lookahead problem and the improvement of management services for more simply use an application of distributed simulation models.

REFERENCES

Boer, C. (2005). Distributed Simulation in Industry. Doctoral Thesis Eramus University Rotterdam

Fujimoto, R. (2000). Parallel and Distributed Simulation Systems . Wiley-Interscience

IEEE 1516-2000 (2000) IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules.

Rabe, M.; Jäkel, F.-W. (2001). Non military use of HLA within distributed manufacturing scenarios. In Schulze, T, Schlechtweg, S., Hinz, V.. (eds) *Proceedings der Tagung Simulation und Visualisierung 2001*. S.141-150. SCS Europe, 2001

Straßburger, S. (2001). Distributed Simulation based on the High Level Architecture in Civilian Application Domains. Doctoral Thesis, University of Magdeburg, Germany.

Strassburger, S., Schulze, T., Lemessi, M. (2006). Applying CSPI Reference Models for Factory Planning. In: *Proceedings of the 2007 Winter Simulation Conference*, eds. S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, pp. 603-609. December 9-12, 2007. Washington, D.C., USA.

Taylor, S.J.E., Mustafee, N., Strassburger, S., Turner, S.J., Low, M.Y.H., Ladbrook, J. (2007). The SISO CSPI PDG standard for commercial off-the-shelf simulation package interoperability reference models. In: *Proceedings of the 2007 Winter Simulation Conference*. S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, eds., pp 594-602

AUTHORS BIOGRAPHY
MICHAEL RAAB is an employee at the Fraunhofer Institute in Magdeburg, Germany. He holds a Masters's degree in Computer Science from the Otto-von-Guericke-University Magdeburg. His research interests include manufacturing simulation, distributed simulation and virtual reality applications.

THOMAS SCHULZE is a professor in the School of Computer Science at the Otto-von-Guericke-University, Magdeburg, Germany. He received the Ph.D. degree in civil engineering in 1979 and his habil. Degree for computer science in 1991 from the University of

Magdeburg. His research interests include modeling methodology, public systems modeling, manufacturing simulation, distributed simulation with HLA and online simulation. He is an active member in the ASIM, the German organization of simulation. His web page can be found via <www-wi.cs.uni-magdeburg.de>.

STEFFEN STRASSBURGER is a professor at the Ilmenau University of Technology in the School of Economic Sciences. In previous positions he was working as head of the "Virtual Development" department at the Fraunhofer Institute in Magdeburg, Germany and as researcher at the DaimlerChrysler Research Center in Ulm, Germany. He holds a Ph.D. and a Diploma degree in Computer Science from the University of Magdeburg, Germany. His research interests include the topics simulation and distributed simulation as well as general interoperability topics within the digital factory context. Mr. Strassburger is also the Vice Chair of SISO's COTS Simulation Package Interoperability Product Development Group. His web page can be found via <www.tu-ilmenau.de/fakww>.