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

INTEGRATION OF EMULATION FUNCTIONALITY INTO AN ESTABLISHED SIMULATION OBJECT LIBRARY

Torben Meyer
Carsten Pöge

Volkswagen AG
Letterbox 1832
38436 Wolfsburg, GERMANY

Gottfried Mayer

BMW Group
Bremerstraße 6
80788 Munich, GERMANY

ABSTRACT

Simulation object libraries for a standardized management and modeling of simulation projects have been established in recent years. This paper discusses the integration of emulation functionality into an established simulation object library for the commissioning of control systems.

This article proposes a three-tiered approach. First, an analysis of the simulation object library is performed, which leads to a grouping of simulation objects from the emulation point of view. Then, the implementation of emulation logic is done for each group. In the third step, the connection between the emulation model and the real control system is being made.

After a presentation of the VDA automotive simulation object library, the integration of emulation functions into this object library is demonstrated on a prototype implementation.

1 INTRODUCTION AND RELATED WORK

Standardization in the field of simulation is a well-established topic. E.g., Abrams (1988) used an object library for parallel simulation. By the use of standardization, periodic or labor-intensive tasks can be simplified and become more efficient. This leads to a higher overall efficiency in simulation projects.

Simulation object libraries are frequently used. In addition to Abrams (1988), Ramis et al. (2008) introduced an object library for the sawmill sector. An object library was also built by a multi-company alliance in the shipyard sector (Steinhauer and König 2010). Moreover Mayer and Pöge (2010) discuss a multi-company simulation object library for the automotive industry. Similar solutions are also known in the simulation of container terminals (Fumarola, Seck, and Verbraeck 2010).

Emulation is the technology used for the commissioning of control systems and is often based upon the same tools as simulation is. While simulation is primarily used in the design and operating phase of manufacturing and logistics systems, emulation is used in the commissioning phase of these systems. Therefore, this paper suggests expanding the application of simulation object libraries to the topic of emulation.

The remainder of this paper is structured as follows: Section 2 contains a description of the VDA automotive simulation object library. Section 3 describes the method to integrate emulation in a simulation object library. This is done by using the VDA automotive object library as an example. Finally, section 4 presents the summary and the lessons learned.

2 AN AUTOMOTIVE OBJECT LIBRARY FOR SIMULATION IN PRODUCTION AND LOGISTICS

This section describes the organization and the motivation for the VDA work group "Process Simulation" for creating a simulation object library for purposes of simulations in the automotive industry.

2.1 Motivation and Objectives of the Cross-Company Cooperation

The global conditions of the automotive industry are challenging. Due to the high number of new plants and new car types, many ramp-ups have to be performed. Moreover, multi-brand plants and projects contribute to a complex planning process. Process simulation was identified as one means to tackle these challenges.

The German Association of the Automotive Industry (VDA) is the representation of more than 580 automobile manufacturers and automotive suppliers. The work group "Process Simulation" is a part of the higher level work group "Digital Factory". It was founded in 2005 with the members AUDI AG, BMW Group, Daimler AG and Volkswagen AG. Later, Adam Opel AG, Bertrandt AG, EDAG GmbH & Co. KGaA, Schäffler Group and ZF Friedrichshafen AG joined the work group.

The working group has the goal of developing methods, interfaces and tools in the field of process simulation and to standardize the use of process simulation in the automotive industry. In addition to this, results from other working groups, such as the ASIM Task Force on Simulation in Production and Logistics, will be analyzed regarding their impact on the simulation activities in the participating companies.

One goal of the cooperation is the positioning of the automotive industry in relation to other industries (e.g., the aviation industry) towards the software manufacturers. In order to do so, the working group can submit common requirements to the software manufacturers to reduce OEM-specific system solutions. This also results in cost savings by avoiding cost-intensive single solutions.

Moreover, the working group acts as a discussion platform for process pimulation in general, its organizational integration in the enterprises and its integration into the automotive planning process.

2.2 Strategy of the VDA Work Group "Process Simulation"

The above mentioned challenges for the automotive industry lead to a high number of simulation projects with an increasing complexity. To master this complexity, a high degree of standardization is essential. Therefore, it has been seen as one of the main tasks within the work group to achieve a standardized methodology for the development of simulation models. Because almost all members are using the simulation tool "Plant Simulation" from Siemens PLM Software, it was possible to think of a common simulation object library within this simulation tool. So, the idea of the VDA automotive simulation object library was born.

Today, the object library is the standard tool within the VDA and serves as the basis for most simulation projects with Plant Simulation. A coordinated release process, an extensive documentation, training materials and workshops for the application of the object library assist the users in applying the standard (Figure 1).

Because all simulation projects are different, the user needs considerable freedom in modeling. The object library allows this freedom, so that the use of the object library in itself will not guarantee the creation of standardized models. Logically, the next step was to develop a modeling guideline for the working with Plant Simulation and the object library. It is described which requirements must be fulfilled in the implementation of simulation models. The developed "Implementation instruction process simulation in the automotive and automotive component supply industry" is now used as the basis for nearly all simulation projects in the field of the automotive industry (see section 2.3.3).

2.3 Scope of the Object Library

2.3.1 Principles and Benefits

The object library is based on complete openness. This is reflected in the fact that all OEMs and their service providers can access and use the object library for free. However, the service provider must sign a confidentiality agreement. The advantage for the suppliers and the service providers is that they are confronted with common requirements from all OEMs.

In addition, it has been shown that the free access to all details of the object library is an important feature. Therefore, the possibility to encrypt individual procedures and to hide details from the public is not being used.

Standardization efforts are extended to the simulation objects themselves and the modeling process. This leads to an easy readability, even of third party simulation models a simulation engineer is opening for the first time. Moreover, the reusability of simulation models is increased by a standardized design and an update mechanism. Finally, the standardized modeling facilitates the integration of various simulation models (created by suppliers or internal simulation engineers) into overall models.

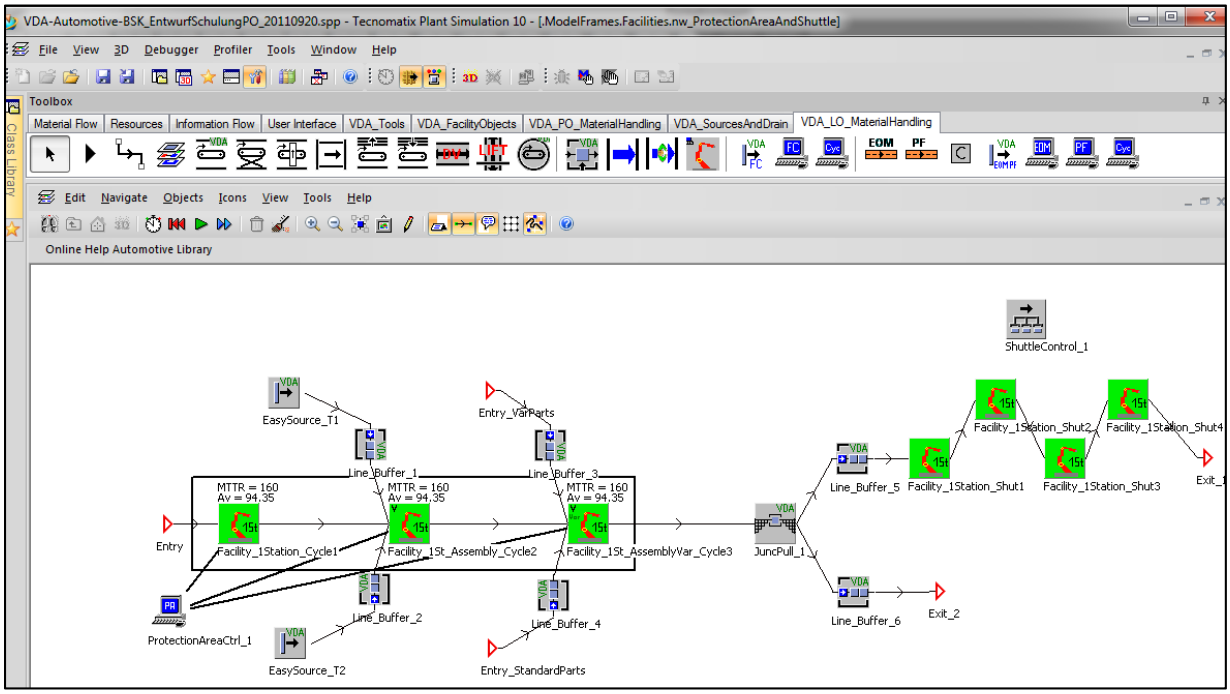

Figure 1: Simulation model for training purposes

2.3.2 Content of the Simulation Object Library

In this section, the individual components of the VDA-object library are briefly introduced. Due to the fact that the object library was designed exclusively for Plant Simulation, an object corresponds to a "brick" in Plant Simulation. The objects are divided into the following categories:

- **Point-Oriented** Objects for Conveying Equipment (PO). This category contains point-oriented objects for conveying equipment. A frequently used object from the PO part of the object library is the Kanban buffer. It is used to implement a pull controlled material flow. The object supports up to ten different variants and corresponds with upstream and downstream Kanban objects via a standardized procedure.

- **Line-Oriented** Conveying Equipment (LO): In contrast to the PO objects, the LO objects are line-oriented conveying equipment. One example is the lifter object that transports moveable units bidirectional in one dimension. It supports one or more lifter on one track. The track handles an unlimited number of stopping positions which can be used in forward and backward direction. In addition to this, one lifter can manage more than one moveable unit in each dimension (side by side, in a row and one upon the other). All line-oriented conveyors have an automatic length calculation feature, therefore no length parameterization is required. This feature is based upon the graphical modeling. The objects adapt the length of the graphic, when the length is changed.
- Special simulation objects for the **paint shop**: One example for the paint shop objects is ColorSort. This object controls and administrates the sortation of line buffers to build color blocks. The administration of this object is made in one single table. Moreover, new conveying elements can be added by the use of drag&drop. The main benefit of this object is that the most common control strategies are already implemented and can be used even with marginal previous knowledge.
- **Powertrain**: The portal is one of the commonly used objects in the powertrain section of the object library. A portal can transport parts between the conveying equipment and the connected machining centers via one or several loaders. It unloads the conveying equipment (receipt of unfinished parts), unloads the machining centers (unloading of finished parts), loads the machining centers (loading with unfinished parts) and places finished parts back onto the conveying equipment. The sequence of the portal processes and the permissible transports are dependent on the selected portal logic. In the present object library two frequent portal logics are modeled. If required, further individual logics can be integrated.
- With **GSL** a file interface belongs to the scope of the object library. E.g., data can be exported from a CAD program and stored in a XML file. The data include information about conveyor and monorail systems. By importing this file, the simulation model is built partially automated.
- **Additional Tools**: This category contains other objects that are responsive to the requirements of an automotive simulation engineer. An example is the Random Control that manages the random number streams and seed values centrally. In addition, the object StatNet creates standardized statistical reports that are needed in the automotive industry (e.g., the turbulence profile of car bodies during the production process which defines how much the measured sequence deviates from the planned sequence; Figure 2).

In addition to the mentioned categories, there are also simulation objects for logistics, laser welding and facilities.

The validation of the object library is supported by test models. The results of the test model's experiments between different versions of the object library or different versions of the simulation tools are evaluated and compared.

2.3.3 Implementation Instruction for Process Simulation in the Automotive Industry

Due to the fact that a large number of simulation service providers, plant suppliers and internal simulation engineers perform simulation projects, standardization in the design of simulation projects is mandatory. The Implementation Instruction also leads to a higher comparability of simulation results. This section provides a very brief overview on the content of the Implementation Instruction for VDA simulation projects. The VDA Implementation Instruction is divided into two sections.

The first section describes the general project plan of a simulation project, beginning from the project schedule to the final return of simulation results. Moreover, the manner of modeling and programming is predetermined. This includes the structure of the model, data management and documentation of the procedure calls. Finally, the first section contains directions about data validation, model verification and model validation for VDA simulation projects.

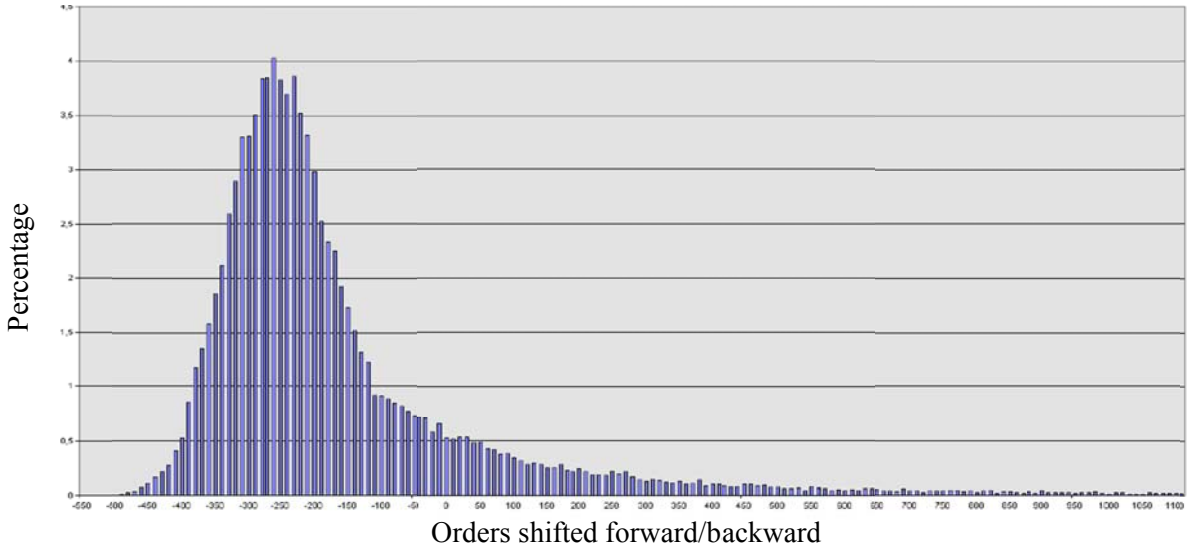

Figure 2: An example turbulence profile of car bodies during the production process

The second section discusses in detail the simulation for the various stages of production (car body shop, paint shop, final assembly, powertrain, logistics and factory simulation): The used input data, the level of detail for the modeling, the statistical parameters of the evaluation and more things are standardized this way.

To sum up, the standardized approach throughout the simulation project allows a large number of service providers to demonstrate their expertise, so that not only the established service providers will benefit.

3 INTEGRATE EMULATION INTO A SIMULATION OBJECT LIBRARY

This section describes an approach to integrate emulation functionality to simulation object library introduced above. The section starts with the motivation for emulation topics and continues with a general concept for creating an emulation framework. Moreover, the special needs of the simulation object library are analyzed and a prototype illustrates the successful implementation.

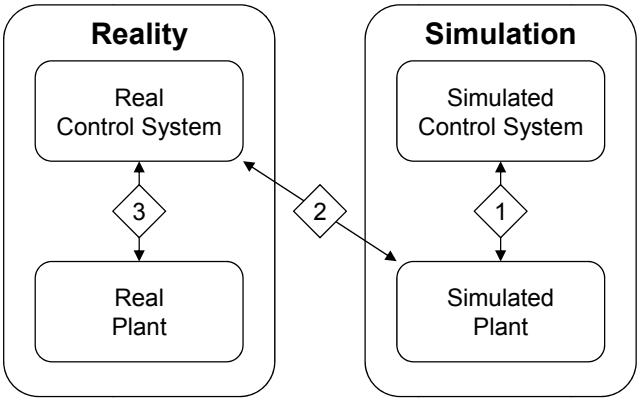

Figure 3: Emulation (Smith and Cho 2008)

3.1 Motivation for Emulation

Due to the rising complexity of automobiles, the complexity of production processes and production facilities rose as well. Today, the manufacturing execution systems (MES) or material flow computers (MFC) regulate differentiated manufacturing and logistics areas. Both control the material flow and build the base for the automatization. The commissioning of the MES or MFC systems is fully integrated in the automotive ramp-up process which means that a delay in the commissioning of these systems may result in a delay of the total ramp-up process.

The principle functionality of emulation for the commissioning of plant control systems was described by Schludermann, Kirchmair, and Vorderwinkler (2000) and is illustrated in Figure 3. On one side, the simulated control system is connected to the simulated plant (1). On the other side, the real control system (e.g., the MES or MFC) is connected to the real plant (3). The linking of the real control system and the simulated plant is known as emulation or soft commissioning (2).

The complexity of MES and MFC commissioning has its reasons in the efforts for the adaptation and configuration of the systems for new fields of application (new plants, new production lines etc.). Moreover, the adaptation of the systems includes the implementation of the control strategies that were the outcome of the simulation study.

For the development and the commissioning phase of MES and MFC projects, it is without question very helpful for software developers (or control engineers) to have direct access to the real system or alternatively to a test system. In most cases the real system does not yet exist, is already in production or is not accessible for software development purposes due to other reasons. For these cases, the developer needs a time-dynamic process simulation model that supports the interface of the real-world systems and that is optionally based upon an earlier simulation study.

The emulation model enables the adaptation of the MES or MFC to late changes of the real system. Furthermore, the emulation model enables benchmark tests of different system versions by repeating test cases.

To sum up, emulation can help to short the commissioning and development time (Grillitsch and Mayer 2010). Additionally, it increases the software quality because it enables additional software test scenarios which have to handle too much data or would impose to many worksteps for conventional test procedures. Finally, the reduced time and the better software quality result in cost savings.

3.2 Concept for the Integration of Emulation Features into an Simulation Object Library

In this subsection, a concept is introduced to add emulation functionality to an already existing simulation object library.

The integration approach is structured as follows (Figure 4): Keep the existing simulation model as it is, transfer all emulation logic for the bidirectional interaction of material and information flow to a dedicated module and swap out the interface protocol related logic to a dedicated interoperability module (Figure 5).

The first step is to analyze and to group all relevant objects of the simulation object library from the emulation engineer's point of view. This means that there are simulation objects that have an almost identical behavior in emulation matters. These object groups have to be identified and described. The analysis and the grouping for a part of the VDA automotive simulation object library can be found in the next subsection.

The next step is about the structure of the emulation logic. As mentioned above, one important step to efficiency is standardization and modularization. Each object group that was identified during the initial step has common requirements which have to correspond with a standardized emulation functionality. Furthermore, all simulation objects have common requirements, which can be tackled by general emulation functions (e.g., translation between external and simulation names).

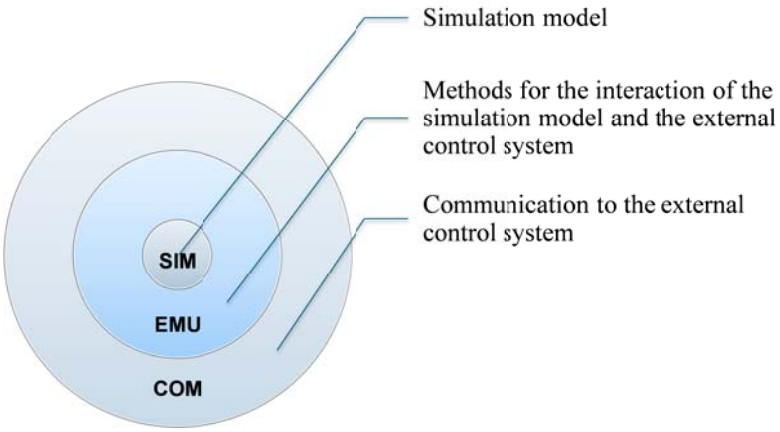

Figure 4: Structure of the integration approach

A lot of interface protocols exist in the context of process simulation. Follert and Trautmann (2006) list three protocols that are used in their case study: OPC (OLE for Process Control) for the fieldbus level communication, message queuing systems for decoupled communication and telegrams over TCP/IP communication. In addition to this, Figure 5 lists some additional interface protocols.

Given this number of different interface protocols, it follows that the burden of implementing new protocols in every emulation project has to be taken away from the emulation engineer. Thus, the object library has to support the most frequent interface protocols by setting up standardized emulation-internal functions to use the interface protocols.

To sum up, adding emulation functionality to an existing simulation object library means to extend the standardization from the simulation objects to the emulation part. The following subsection shows, how this abstract concept was applied to the VDA automotive simulation object library.

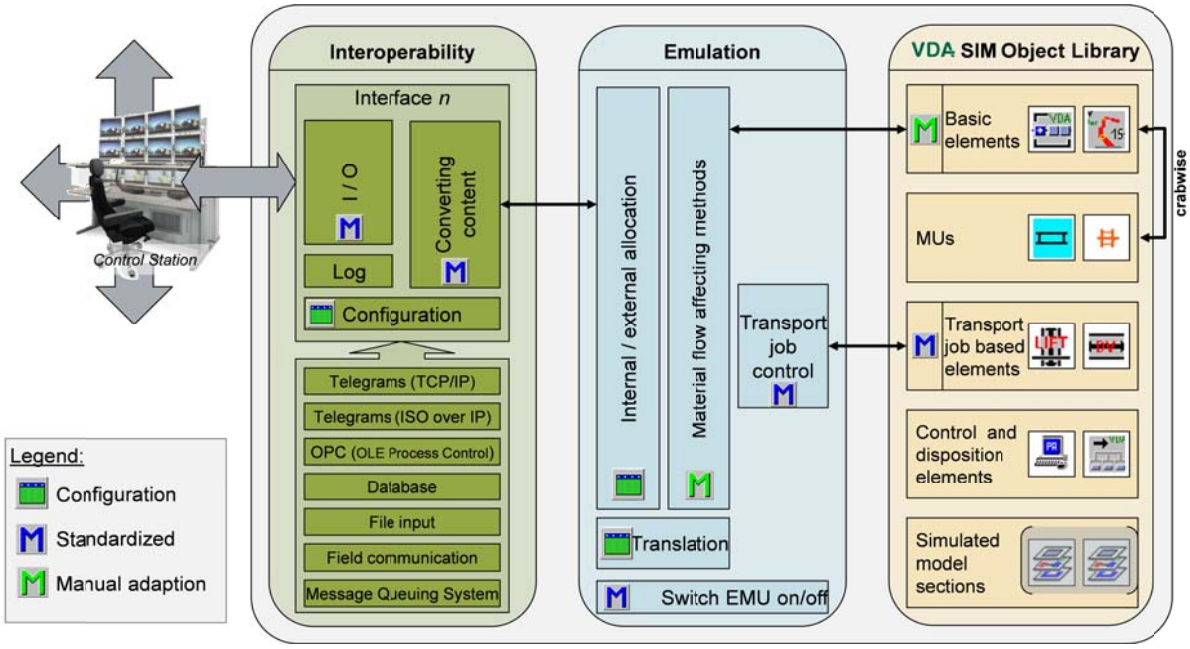

Figure 5: Adding emulation functionality to the VDA automotive object library

3.3 Analyzing the VDA Automotive Object Library

To add emulation functionality to the simulation object library, the object library has to be analyzed from the emulation engineer's point of view (Figure 5). This means to group all objects by their common properties. Objects from one group are handled the same way from the control system. Analyzing the VDA automotive object library resulted in five object groups which are described below.

- **Basic elements** are processing movable units. Objects of this group are interacting with the control system and changing their modus based upon the external input. Moreover, basic elements send status information about the current status and the progress. Examples: Line conveyor, generic facility, assembly station
- The group "**moveable units**" (MUs) contains simulation objects that do not communicate directly with the control system. MUs do not move on their own but are moved by other objects. They can contain AutoID information (e.g., RFID, Barcode etc.). The communication of movable units is done over other objects. The typical information that is saved within the AutoID is e.g., the transportation target or a unique ID. Examples: Car body, Skid, EOM-Carrier
- **Transport job based elements** are transporting movable units and are often made of conveyors. In most cases, these kinds of elements have more than one entry, more than one exit and as the case may be more than one transporting element. This group of elements administrates transport jobs that are created, changed and deleted by the communication to the external control system. The objects send information about incoming MUs, status information about the current progress of MUs within this element and information about the end of transport jobs to the control system. Example: Lifter.
- The "**control and disposition elements**" group contains dispositive objects that are independent from the emulation controls. The objects do not have any special requirements related to the emulation. This is due to the fact that these objects only offer basic control mechanisms and that they do not communicate to the control system.
  An example is the object "protection area control". All machines belonging to the protection area are stopping their production if one of the machine stops (e.g., because of a failure). The functionality of the protection area control is equally relevant to simulation and emulation in equal measure and does not communicate to the control system.
- **Simulated model sections** are a part of the emulation model but fulfilling simulation functionality only. Objects of this group are not involved in the bidirectional communication with the control systems. Examples: pulk source, Kanban buffer

The next subsection describes a case study of an emulation prototype that uses an emulation framework based upon the automotive object library and the introduced concept.

3.4 Prototype

A prototypical development (Figure 6) for the integration of emulation functionality was created for a manufacturing system. It is based upon the VDA automotive simulation object library. This manufacturing system produces components for the powertrain unit and has a high degree of automatization. It is controlled by one central MES that controls several manufacturing cells. The material flow is connected by two cranes.

The MES communicates with the manufacturing cells and the cranes via an interface protocol that uses telegrams over TCP/IP. The task was to build an emulation model for the above described production system.

The basic architecture of the emulation is depicted in Figure 5. The following enumeration shows the projects course of action.

1. First of all, the existing simulation model was extended by a central variable to switch between emulation and simulation. Furthermore, all simulation methods that are actually containing the control instructions for the material flow were modified to check this central variable: If the mod-

el is used for a simulation then the simulation logic will be executed. Otherwise, the model calls new emulation logic.

2. The emulation has to support the interface protocol. This was separated in two parts: On one side, a module "I/O" was used for the general support of the socket connection (TCP/IP protocol, connect/disconnect server, send/receive strings etc.). On the other hand, a second module was used to process the input from the "I/O" module (in this case the telegram string) and to decompose the string into the single information items. Afterwards, this information is forwarded in a standardized format to the emulation logic. For the ease of customizing, the decomposition was based on one single configuration table.
3. Finally, both ends (the calls from the interface connection and from the simulation logic) have to be connected with the emulation logic. One common task is the translation between external and internal objects names which can also be standardized easily.
   In addition to this, the grouping of simulation objects can be helpful in standardizing emulation logic: In this case study, the cranes were identified as transport job based elements. It follows that all emulation logic for creating, deleting and modifying crane transport jobs was consolidated in a "transport job control" module (Figure 6).

The prototype shows that a lot of work of the emulation engineer can be supported by standardization and modularization. But for all that, the main task of the emulation engineer is to implement the individual logic for objects that are out of any standard.

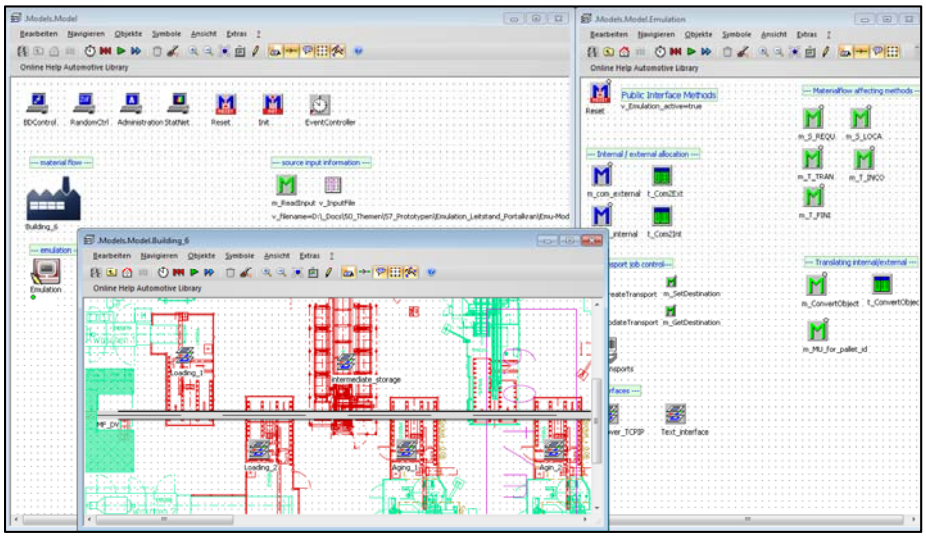

Figure 6: Screenshot of the emulation model

4 CONCLUSION AND FURTHER WORK

The VDA automotive simulation object library is a continuing success story for inter-company cooperation. Simulation objects have been created, covering the needs of the automotive industry. These results have been shared openly with the simulation engineers in the automotive industry and with the simulation service providers.

Against the current challenges in the automotive industry, the emulation (in terms of a virtual commissioning) gains importance. In contrast to the equipment manufacturers, the OEM plant operators have to support a wide range of different systems with the emulation. This paper has shown how a simulation object library can be extended by emulation functions.

Important future issues in the emulation context are the emulation at the field level, the transition between simulation and emulation and the creation of a unified modeling process for emulation.

The next steps in the development of the VDA automotive simulation object library include energy and fluid simulation and the connection of 3D visualization to process simulation (time dynamic animation in the digital factory mock up). In addition, the simulation assistance and the modeling of employees in the assembly areas are currently in the development phase.

REFERENCES

Abrams, M. 1988. "The Object Library for Parallel Simulation (OLPS)." In *Proceeding of the 1998 Winter Simulation Conference*, Edited by M. Abrams, P. Haigh, and J. Comfort, 210–219. New York: Association for Computing Machinery.

Follert, G., and Trautmann, A. 2006. "Emulation intralogistischer Systeme." In *Simulation in Produktion und Logistik 2006*, Edited by S. Wenzel, 521–530. Erlangen: SCS Publ. House.

Fumarola, M., Seck, M., and Verbraeck, A. 2010. "A DEVS component library for simulation-based design of automated container terminals." In *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques (SIMUTools '10),* Edited by ICST. Brussels: ICST.

Grillitsch, U., and Mayer, G. 2010. „Auf dem Weg zum Standard - Virtuelle Inbetriebnahme von IT-Steuerungssystemen in der Produktionssteuerung." In *Integrationsaspekte der Simulation*, Edited by G. Zülch, and P. Stock, 591–598. Karlsruhe: KIT Scientific Publishing.

Mayer, G., and Pöge, C. 2010. "The road to standardisation – from the idea to the realisation of the VDA automotive toolkit." In *Integrationsaspekte der Simulation*, Edited by G. Zülch, and P. Stock, 29–36. Karlsruhe: KIT Scientific Publishing.

Ramis, F. J., Concha, P.,Neriz, L., and Sepúlveda, J. A. 2008. "An Object Oriented Library for Sawmill Simulation." In *Proceedings of the 51st International Convention of Society of Wood Science and Technology*, Edited by E. K. Pepke, WS-42. Madison, Wisconsin: Society of Wood Science and Technology.

Schludermann, H., Kirchmair, T., and Vorderwinkler, M. 2000. "Soft-commissioning: Hardware-in-the-loop-based verification of controller software." In *Proceedings of the 2000 Winter Simulation Conference*, Edited by J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 893–899. New York, New York: Association for Computing Machinery Press.

Smith, J. S., and Cho, Y. 2008. "Offline commissioning of a plc-based control system using arena." In *Proceedings of the 2008 Winter Simuation Conference*. Edited by S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, and J. W. Fowler, 1802–1810. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc..

Steinhauer, D., and König, M. 2010. "Konzept zum effektiven Aufbau von Simulationsmodellen für die Unikatproduktion." In *Integrationsaspekte der Simulation*, Edited by G. Zülch, and P. Stock, 157–164. Karlsruhe: KIT Scientific Publishing.

AUTHOR BIOGRAPHIES

TORBEN MEYER is a Ph.D. student at the Ilmenau University of Technology and a researcher for the "Digital Factory" project at Volkswagen in Wolfsburg, Germany. Previously, he was working in the simulation and emulation department of Dematic in Offenbach, Germany. He holds a master degree in industrial engineer and management from the Ilmenau University of Technology and a bachelor degree from the Wedel University of Applied Sciences. He is mainly interested in simulation/emulation topics for production and logistic processes. His email address is torben.meyer@volkswagen.de.

CARSTEN PÖGE is the project manager for the field of process simulation at the Volkswagen Group IT Department and holds a Diploma degree in economic sciences from the Otto-von-Guericke-University Magdeburg. He is the responsible IT-representative in the work group "Process Simulation" within

Volkswagens "Digital Factory" project and the assistant speaker of the VDA work group "Process Simulation". His email address is carsten.poege@volkswagen.de.

GOTTFRIED MAYER is responsible for the field of simulation at the BMW IT Department. He is the speaker of the VDA work group "Process Simulation" and the assistant speaker of the VDA work group "virtual ramp up". He is a member of the ASIM task force "simulation in production and logistics" and the VDI board "simulation and modeling". His e-mail address is gottfried.mayer@bmw.de.