

FACILITATING EMULATION PROJECT ANALYSIS THROUGH THE USE OF PROTOCOL STATE MACHINES

Torben Meyer

Volkswagen AG
Letterbox 1832
D-38448 Wolfsburg, GERMANY

Steffen Strassburger

Ilmenau University of Technology
P.O. Box 100 565
D-98694 Ilmenau, GERMANY

ABSTRACT

Emulation is a well-established technology which supports the software development and the commissioning phase of control systems by connecting the real control system with a simulated material flow system. This article suggests the use of protocol state machines to firstly formally describe the interface communication and secondly analyze emulation experiments.

1 INTRODUCTION TO EMULATION

Emulation is defined as a “model that accepts the same inputs and produces the same outputs as a given system” (IEEE 1989). The basic idea is to connect a real control system with the simulated technical system instead of the real technical system. The emulation model (often based upon a discrete event simulation) has the capability to communicate with the real control system (e.g., a manufacturing execution system) by the use of an interface protocol. In addition to the data that is needed for a conventional simulation, a detailed description of the interface (for example Telegrams over TCP/IP) between control system and technical system is obligatory.

The control system on one side analyzes the system situation in total with a focus on production planning and controlling subjects. The emulation on the other side focuses on the representation of the material flow details. For this reason, the different aspects of understanding potential emulation malfunctions typically cause detailed investigations of all log files and the systems behavior. The approach suggested in this paper can significantly improve this analysis process.

2 USING A PROTOCOL STATE MACHINE TO DESCRIBE PROCESSES IN EMULATION

A protocol state machine (PSM) describes the lifecycle of objects and specifies which operations can be performed in which order on them. PSMs have a resemblance with event graphs (Buss 2006) which are used for the description of event-based simulation models. However, instead of describing the transfer of a systems state based on different event types and stochastic influences, PSMs describe state changes of system entities based on messages received from and sent to the control system. The UML consortium has published a suggestion for the visualization of protocol state machines which is also described by Selic (1998). An operation in the PSM represents information or data that is transmitted via the interface from the control system to the emulation model or vice versa. If the interface uses telegrams over TCP/IP, each telegram stands for an operation in the protocol state machine.

To automatically process the information contained in the visualization of protocol state machines, it is essential to save it in a computer readable format. It is thinkable to describe the content of a telegram in different formats. In a prototypical tool for demonstrating the use of PSMs for emulation analysis, we used an XML-format to enhance the telegram description with the PSM information.

The retracing process of the interface log file is essential to understand the material and information flow in the emulated system afterwards. All telegrams of the interface log are interpreted in chronological order. The syntax of all telegrams from the interface log file is checked to detect mistakes in the composition of the telegrams.

In addition to this, our tool traces the status change of objects by the operations/telegrams to the protocol state machine. The analysis tool holds a PSM for each object and remembers their last states. When receiving a new telegram, it checks if the associated operation can be performed depending on the previous state. If this operation cannot be executed because the current state of the object and the required state of the operation do not match, this could be the reason for a malfunction. If both parameters do match, the tool saves the new state of the object and continues with the next telegram.

All telegrams from the saved emulation experiment are parsed this way. The results are written as an output file containing the trace file data and will be used for a visual analysis in a dedicated analysis tool. For a good acceptance of all involved persons (emulation engineers as well as control engineers), the analysis tool is not integrated into the control system or the emulation. Another requirement was the ability to replay previous experiments. This means that the user can use the recorded log files afterwards to see the movements of the objects and to jump forward and backward in the simulation time line. The tool's purpose is to build a discussion platform for all participants.

3 CONCLUSION AND FUTURE WORK

The described method transfers the idea of protocol state machines to the area of emulation. Protocol state machines have been used to describe the systems behavior and to analyze the emulation experiments. Each PSM represents a moveable unit in the emulation. Moreover, the control orders between the control system and the emulation model are used equally to operations in the protocol state machine. By retracing the control orders from the interface log file the emulation experiment can be analyzed and understood easily. The concept was validated in a prototype.

The future work focuses on the integration of additional information sources into the analysis. In addition to the interface log file, the integration of control system and emulation model logs will lead to a holistic insight during the analysis.

REFERENCES

- Buss, A. H. 2006. "Modeling with event graphs". In Proceedings of the 2006 Winter Simulation Conference, Edited by L. F. Perrone, 153–160. Orlando : IEEE.
- IEEE. 1989. IEEE Standard Glossary of Modeling and Simulation Terminology. New York: IEEE Publishing.
- Selic, B. 1998. "Using UML for modeling complex real-time systems". In Lecture Notes in Computer Science, Edited by F. Mueller, and A. Bestavros, 250–260. Berlin: Springer Verlag.

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 worked in the simulation and emulation department of Dematic in Offenbach, Germany. He holds a master degree in industrial engineering and management from the Ilmenau University of Technology. His email address is torben.meyer@volkswagen.de.

STEFFEN STRASSBUGER is a professor at the Ilmenau University of Technology in the School of Economic Sciences. Previously he was head of the "Virtual Development" department at the Fraunhofer Institute in Magdeburg, Germany and a 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. He is a member of the editorial board of the Journal of Simulation. His research interests include distributed simulation as well as general interoperability topics within the digital factory context. His web page and email address are www.tu-ilmenau.de/wil and steffen.strassburger@tu-ilmenau.de, respectively.