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

GUISE - a tool for GUIding Simulation Experiments

Stefan Leye
Adelinde M. Uhrmacher

Albert Einstein Str. 22
University of Rostock
18059 Rostock, GERMANY

ABSTRACT

With the rising number and diversity of simulation experiment methods, the need for a tool supporting an easy exploitation of those methods emerges. We introduce GUISE, an experiment tool to support users in conducting experiments. We structure simulation experiments according to six tasks: specification, configuration of model parameters, simulation, data collection, analysis, and evaluation. This structure provides the required flexibility to seamlessly integrate various methods into the tool and combine them to pursue different goals (e.g., validation, optimization, etc.). To support experimenters in selecting and composing suitable methods, GUISE exploits machine learning techniques, which we illustrate at the example of steady-state estimation.

Motivation, Structure, and Functionality of GUISE

The variety of simulation experiment methods can be overwhelming for experimenters. Studies revealed that a better support is required, as users often lack the required mathematical background for systematic experimentation (Perrone, Cicconetti, Stea, and Ward 2009). Tools are needed that allow the seamless integration of different methods, guide the user through the steps of the experiment, and support him in selecting the most suitable method for the task at hand. This has been the motivation for developing GUISE (GUIding Simulation Experiments). It is based on the M&S framework JAMES II (Himmelspach and Uhrmacher 2007) and exploits its plug-in system to integrate various methods. It furthermore, provides an experiment workflow (Rybacki, Leye, Himmelspach, and Uhrmacher 2012) that is based on the six tasks of simulation experiments (Leye and Uhrmacher 2010), and extends the Simulation Algorithm Selection Framework (SASF) (Ewald 2010) to support the selection and use of experiment methods.

The six tasks of a simulation experiment are: specification — defining the experiment; configuration — selecting interesting model parameters; simulation — executing the model; data collection — collecting data of interest; analysis — analyzing the collected data; and evaluation — assessing the analysis results. Due to distinguishing these tasks, the experiment gets a clear and explicit structure which makes it more transparent to the user and supports guidance. Let us illustrate this with an example.

First, we specify the overall experiment, e.g., an optimization experiment with steady state mean as target function, based on a species-reaction model of $MgCl_2$. Now GUISE comes up with a set of optimization algorithms, (as all other methods in GUISE realized as plug-ins), the user selects the particle swarm algorithm, and as a search space the kinetic rates and initial concentrations of the model. Now the model shall be executed — again a set of (exact and approximate) execution algorithms is available, and the user selects the Gillespie Direct Method. The user defines the trajectory of Mg particles, as data that shall be collected. For the analysis of the produced trajectories again a set of steady state estimators is available, among them the user selects the MSER steady state estimator. Finally the overall results are evaluated by the particle swarm algorithm. In the described process, there exist different points where the

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

user has to select and configure methods, i.e., in the configuration an optimization algorithm needs to be selected and configured, in the simulation step, an execution algorithm needs to be selected and configured, and in the analysis step a steady state estimator needs to be selected and configured.

However, selecting among the variety of available methods the most suitable, is a task, most users are unwilling or unable to do. In (Ewald 2010), SASF exploits machine learning methods to select suitable execution algorithms. GUISE builds on this and aims at providing support for users in the other two steps as well, by orchestrating methods in ensembles. Referring to the analysis step, performance studies with steady state mean estimation have already shown the effectiveness of the approach (the created steady state estimator ensemble had a higher steady state detection rate than any other used steady state estimator). Figure 1 shows the general process.

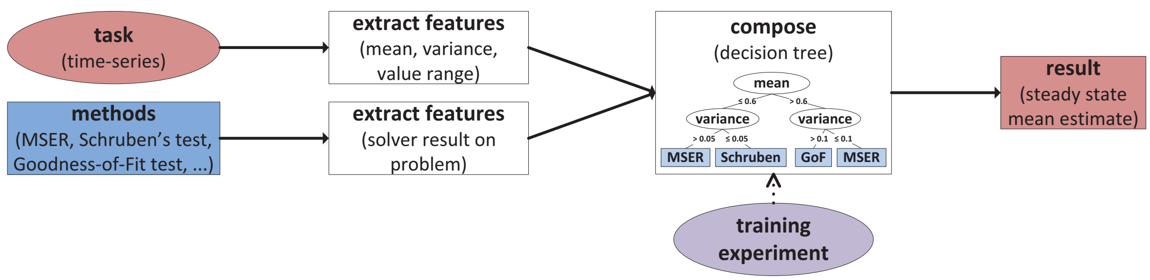

Figure 1: Process of steady state estimator composition in GUISE.

GUISE creates ensembles of methods — in our example ensembles of steady state estimators to solve the task of estimating the steady state mean of a time-series. Therefore, a set of feature extractors retrieve relevant features from the time-series (e.g., variance and mean of the time-series) and estimators (e.g., estimated mean). Considering these features, the results of the methods are composed to a new result. To train the composition procedure, all steady state estimators available in JAMES II are applied on an extensive set of time-series ($>$ 150,000) covering relevant properties for steady state estimation. Based on the training data, machine-learning techniques are used to generate a composition scheme e.g., expressed as a decision tree that considers given features to decide which steady state estimator provides the supposedly best steady state mean estimate.

ACKNOWLEDGMENTS

This research has been supported by the DFG (German Research Foundation), via research training group 1387 (*dIEM oSiRiS*) and the research project *CoSA*.

REFERENCES

Ewald, R. 2010. *Automatic Algorithm Selection for Complex Simulation Problems*. Ph. D. thesis, Faculty of Computer Science and Electrical Engineering, University of Rostock.

Himmelspach, J., and A. M. Uhrmacher. 2007, March. "Plug'n simulate". In *Spring Simulation Multiconference*, 137–143: IEEE.

Leye, S., and A. M. Uhrmacher. 2010. "A Flexible and Extensible Architecture for Experimental Model Validation". In *SIMUTools*: ICST/IEEE.

Perrone, L. F., C. Cicconetti, G. Stea, and B. C. Ward. 2009. "On The Automation of Computer Network Simulators". In *SIMUTools*: ICST/IEEE.

Rybacki, S., S. Leye, J. Himmelspach, and A. M. Uhrmacher. 2012. "Template and Frame based Experiment Workflows in Modeling and Simulation Software with WORMS". In *Workshop on Scientific and Engineering Workflows: Advances in Data and Event-Driven Workflows*, 25–32: IEEE.