

Engineering of Coupled Simulation Models for Mechatronic Systems

Petr Novák, Petr Kadera, Václav Jirkovský, Pavel Vrba
and Stefan Biffi

Abstract Simulation models play a crucial role in testing and fine-tuning of control systems for complex industrial systems. They are important parts of the sustainable, service-oriented manufacturing value chain by facilitating early and efficient defect detection and risk mitigation. However, the design phase of simulation models is time-consuming and error-prone, thus it should be improved to become more efficient. The design of simulation models for mechatronic systems has to cope with two basic challenges: (1) the heterogeneous nature of mechatronic systems, which are described with various overlapping engineering plans, and (2) the separation of monolithic simulations into distributed simulation modules, to better conquer the computational complexity of simulation models. This paper addresses both challenges: (a) we propose an application of semantic integration and linked data for sharing and capturing knowledge for simulation model design between various engineering plans; (b) we explain how to structurally connect simulation modules, which are dynamically coupled. The proposed method utilizes the extended bond-graph theory.

P. Novák (✉) · S. Biffi

Christian Doppler Laboratory for Software Engineering Integration for Flexible Automation Systems, Vienna University of Technology, 1040 Vienna, Austria
e-mail: petr.novak@tuwien.ac.at

S. Biffi

e-mail: stefan.biffi@tuwien.ac.at

P. Novák · P. Kadera · V. Jirkovský · P. Vrba

Czech Technical University in Prague, 121 35 Prague, Czech Republic

P. Kadera

e-mail: petr.kadera@fel.cvut.cz

V. Jirkovský

e-mail: vjirkovsky@ra.rockwell.com

P. Vrba

e-mail: pavel.vrba@fel.cvut.cz

V. Jirkovský

Rockwell Automation Research and Development Center, Prague, Czech Republic

The simulation modules work independently of each other but with interaction, similar to a multi-agent system. Since the computational execution of coupled simulations is a crucial obstacle especially for mechatronic systems, we show how performance analysis can significantly improve the definition of simulation workflows.

Keywords Dynamic simulation • Semantic integration • Performance analysis • Bond graphs • Industrial processes

1 Introduction

A computer simulation aims at approximating the behaviour of real systems. The simulation is useful for key tasks in the manufacturing value chain, such as model-based control, estimation of unmeasured states, operator training, or decision-making support. Although simulation models can be used to improve a sustainable operation of real plants, to reduce waste, and to save energy, their design phase is time-consuming and error-prone, hence their use is still limited [8]. The engineering process of simulation models should be improved in order to bring simulation benefits into daily industrial practice. This paper addresses the design phase of dynamic simulation models for industrial processes as well as the simulation model performance analysis.

Since industrial processes are becoming large-scale and complex, it is not efficient to develop monolithic simulation models for these systems any more. A current trend is the distribution of simulation models into a set of inter-linked simulation modules. Such a complex simulation workflow is frequently called coupled simulation and can even include real hardware in the simulation loop. The separation of simulations into modules can be compared to an agent-based approach. Simulation modules correspond to agents and one of the research goals is to support the encapsulation of modules according to the holonic approach known from manufacturing control. As well, it is crucial to analyse performance and bottlenecks in the simulation workflows. The presented research is related to the development of the Simulation Integration Framework, see Novák and Šindelář [6] or Novák et al. [7] for further details.

To improve the design phase of coupled simulation models for mechatronic systems, the following research issues were identified:

RI-1: Semantic integration of engineering knowledge. Semantic integration Noy and Doan [9] should address the following features and requirements: (1) Industrial systems are large-scale and complex and they are described by various kinds of engineering plans. (2) Most of these systems are of a mechatronic nature. They are engineered based on a collaborative multi-disciplinary process combining diverse, partly overlapping plans. Engineering concepts are locally described in heterogeneous engineering plans and these local representations should be integrated in order to allow describing simulations and their workflows. (3) Engineering process support would save costs and effort for (re-)design and re-use.

RI-2: Module-based simulation. To improve the design and execution phases of simulations models, they should be internally created as coupled simulations consisting of simulation modules. Benefits of this simulation architecture can be summarized as follows: (1) Parallelization of simulation execution; (2) Easier maintenance and (re-)design of simulations; (3) Significantly faster initialization of simulation environments; and (4) Simplified testing of simulation modules and their fine-tuning.

RI-3: Performance analysis of coupled simulation. An important limitation for simulation use is the computational performance. Coupled simulations should be equipped with performance analysis reflecting: (1) Limited insight of simulation experts into coupled simulation assemblies; (2) Required analysis methods and tools to estimate the performance requirements; and (3) Results of performance analysis should be provided to experts in such a form that identifies bottleneck modules. Based on such results, simulation experts can redefine the simulation modules in order to improve the performance of the whole simulation workflow.

The remainder of this paper is structured as follows. Section 2 proposes the utilization of semantic integration for heterogeneous engineering knowledge integration. Section 3 explains how bond graphs improve the design of coupled simulations and a structure of integration glue modules. Section 4 explains the role of simulation performance analysis for simulations. Section 5 concludes and proposes future work.

2 Semantic Integration of Heterogeneous Engineering Data

This section addresses the *RI-1*, which is focused on the semantic integration of engineering knowledge for the model-driven design of simulation models and the configuration of their runtime workflows.

Semantic integration is a key interoperability enabler for the Semantic Web (Ontology matching) as well as a needed functionality in many classical data integration tasks dealing with the semantic heterogeneity problem. It takes schema descriptions and data (e.g., ontologies in the case of Semantic Web) and determines as its output a semantic alignment—a set of correspondences among the semantically related entities. The correspondences can be utilized for data translation, query answering, or navigation on the web of data.

Semantic heterogeneity means variance in the contents of information and intended meaning that may cause application unusability. Dealing with semantic heterogeneity is thus crucial in every extensive complex system. Three main causes of semantic heterogeneity can be distinguished in these systems [3]:

- *Confounding conflicts* occur when information items seem to have the same meaning, but differ in reality, e.g., owing to different temporal contexts.
- *Scaling conflicts* occur when different reference systems are used to measure a value. Examples are different currencies or length measures, e.g., feet/meters.

- *Naming conflicts* occur when naming schemes of information differ significantly. A frequent phenomenon is the presence of homonyms and synonyms.

To describe the similarity of entities, the following basic similarity measures can be used. The most common techniques are **string-based**, which are methods comparing strings—entity labels (e.g., *n*-gram similarity measure). On the other hand, **language-based** techniques are based on Natural Language Processing methods for extracting meaningful terms (e.g., cosynonymy similarity for WordNet). The next data characteristic to match can be the structure of entities found in the data source schema—**structure-based** similarity measures (e.g., structural dissimilarity on a hierarchy).

Unfortunately, these basic techniques are suitable only for a specific type of dissimilarity. Therefore, they can be considered as the cornerstones of similarity aggregation systems. The problem is how to combine these basic techniques.

Independent similarity measures can be aggregated by means of **multidimensional distances**. Well known multidimensional distance is Minkowski distance:

$$sim(x, x') = \sqrt[p]{\sum_{i=1}^n sim_i(x, x')^p},$$

where $sim_i(x, x')$ is the *i*th similarity measure of objects x, x' . The distance is equal to the Euclidean distance for $p = 2$ and to the Chebyshev distance for $p = +\infty$.

More sophisticated aggregation systems are based on **machine learning** techniques. These systems are able to extract unknown correspondences among entities and obtain considerably improved performance when compared to that of existing systems [4].

Similarity measure aggregation suitable for engineering data has to assure the best precision and recall what can be achieved. The typical characteristic of fully automatic aggregation systems is their capability of handling big data structures quickly, but on the other hand, they cannot achieve sufficient precision that the simulation design application needs. The compromise between the speed and quality of a matching output are semi-automatic systems. The MAPSOM framework described in Jirkovsky et al. [5] seems to be a promising solution for engineering applications. The MAPSOM framework is based on similarity measure aggregation with the help of a self-organizing map and user involvement for tuning up the results by means of active learning.

An exemplary engineering plan to be semantically integrated with other plans and software tools is a piping and instrumentation diagram (P&ID), which is depicted in Fig. 1 for the case of a simple tank system. This system should be simulated and its behaviour should be split into three modules, as depicted by module cuts in Fig. 1. This use-case will be used later on in the paper to demonstrate the proposed solution.

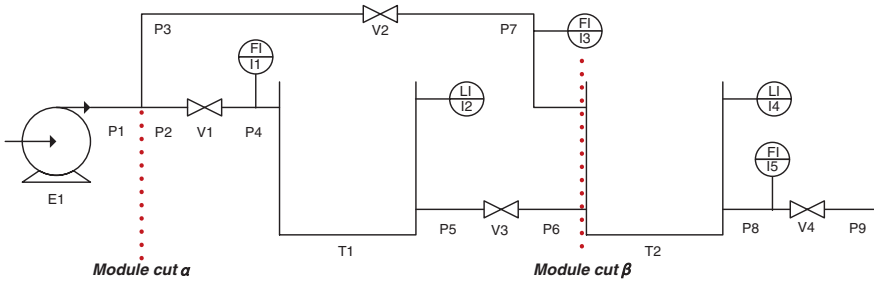


Fig. 1 P&ID of the tank system use-case with selected cuts of the plant into modules

3 Integrating Simulation Modules with Extended Bond Graphs

This section addresses the research issue *RI-2*, which is focused on problems with the design of simulation modules and their integration into simulation workflows. The entry point for this task is a description of a real system obtained by the semantic integration of engineering plans.

A dynamic mathematical model for a physical system can be systematically created based on the bond-graph theory [2]. Bond graphs are based on three types of analogies: (i) signal analogy, (ii) component analogy, and (iii) connection analogy. Bond graphs are based on describing power flows between system components. Power is the product of flow and effort signals that are generic signals defined by the signal analogy. An abstraction of component connection is introduced by the connection analogy. Bond graphs provide 0-junctions as models for parallel connections of components and 1-junctions as models for serial connections of components. Both types of junctions are tackled in a very similar way. An important feature of bond graphs is causality, which is the main difference between bond graphs and power graphs. Causality assignment denotes which of the signals flow and effort is the output variable and which one is the input variable. Causality is denoted in the graphical expression with a small stroke on such side of each bond, where a connected component transfers effort to flow. Further details about bond graph theory and practice can be found in Gawthrop and Bevan [2].

The authors proposed an extended bond graph theory in Novák and Šindelář [6] as a design method to support creating simulations from gray-box components. On top of this method, we explain how to design modularized simulations now.

In order to increase the modularity and computational performance of simulations, the simulation models are frequently required to be split into several simulation modules. These modules are relatively independent, but can be dynamically coupled. We assume that splitting simulations into several modules is driven by a human simulation expert, while the proposed method supports the expert with semi-automating structural and technical tasks. The method supports two basic types of cuts of a plant into modules: (1) cuts on the junction level, and (2) cuts on the bond level.

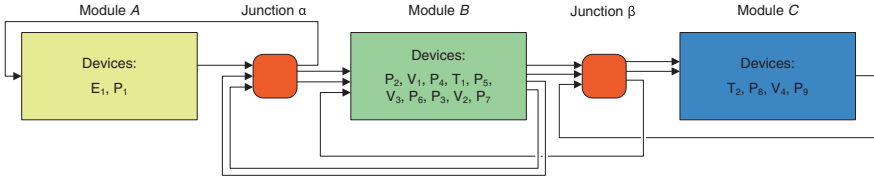


Fig. 2 Coupled simulation for the tank system which consists of simulation modules, integration junctions as “glue modules”, and signal routing

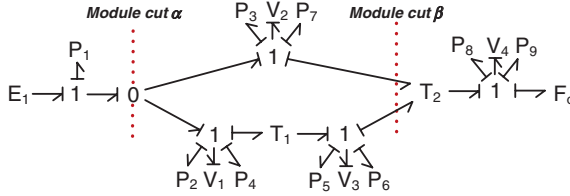


Fig. 3 Bond graph with cuts of the tank system into simulation modules

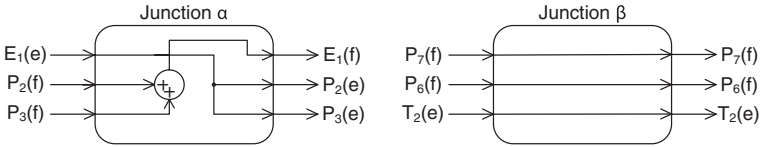


Fig. 4 Internal structure of integration junctions, which is inferred from a system bond graph

A practical example of splitting a simulation of the tank system is depicted in Fig. 2. The system is simulated with three modules, which are connected with two junctions. The junctions can be considered as glue modules, integrating simulation modules. One of the core contributions of this paper is shifting the bond-graph approach to an upper level (i.e., from a device level to a module level) as well as the way how to implement internal structure of these “glue modules” for integration.

Bond graphs can be utilized for designing glue modules as follows. The position of plant cuts is inserted in the bond graph as it is shown in Fig. 3 for the case of two cuts of the tank system. Having such bond graph is important for the design of glue modules. A more complex situation emerges in case of cuts on the junction level, where signals should be added or subtracted. The mathematical expression is obtained based on surrounding bonds and their causality assignments. For the tank system, the glue modules are depicted in Fig. 4. In the case of cut α , the power flows into the 0-junction via a bond from the left 1-junction. Effort is an input of the 0-junction and the junction calculates output flows as the sum of the two flows to the rest of the system on the right. We can see that the inner implementation of the glue module depends on neighbouring bonds only. However, it

is necessary to create the bond graph for the whole system in order to be able to assign causality and power flows correctly. If the cut is on the bond level, the situation is easier and only the pair of signals is assigned as input or output of the glue module. This situation is depicted on the right-hand side of Fig. 4 as the cut β and we can see that the solution is trivial in this case.

In both cases of cut types, the glue modules are important for (i) timing aspects, where the module can provide aggregation of data, re-sampling and synchronization. In addition, (ii) the glue module is useful for the performance analysis as a probe into the system. The proposed method solves the simulation module integration problem from the structural point of view. In addition, it is required to analyse the performance of modules and find bottlenecks in the simulation model schema in order to provide engineers with support for the simulation system optimization and redesign. Such issues are discussed in the following section.

4 Performance Analysis for Coupled Simulations

Performance modelling involves multiple approaches. We have considered some of them for modelling interactions between coupled simulations with focus on the identification of the system bottlenecks and the maximal system performance. Particularly, we have considered the usage of the following performance modelling notations, including their pros and cons:

- Bounded Analysis Based on Operational Laws [1]
 - Pros: Easy to use, no special skills required, low computational demands;
 - Cons: Low expressivity (neither component interactions nor synchronization);
- Queuing Networks
 - Pros: Simple notation, separated characteristics of specific components;
 - Cons: No support for modelling synchronization and joint probabilities;
- Queuing Petri Nets
 - Pros: Transparent graphical modeling notation;
 - Cons: Non-trivial applications generate more states than is feasible to handle;
- Stochastic Process Algebras
 - Pros: High expressivity (synchronization, exclusive access, connections);
 - Cons: Requires special skills, aggregation of states cannot be easily automated.

Based on the pros and cons of these different notations, we focused our attention on the Bounded Analysis (BA), which is based on operational laws [1] and provides a reasonable compromise between the expressivity, usability and runtime solvability. This approach identifies the system bottlenecks and the maximal system throughput. The operational laws are directly applicable in single class cases, but the multi-class cases are more complicated. Therefore, we use a method based

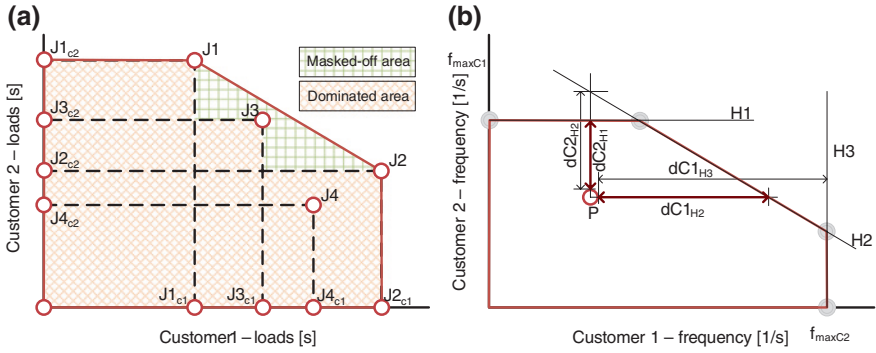


Fig. 5 **a** Loads of individual stations by customer classes and their convex hull, **b** transformation from loads to the frequency space and identification of the reserves

on convex polytopes to identify service stations that may be bottlenecks under certain circumstances. The modelled system must be described by a *Loading matrix* $L[n \times m]$ (n is the number of job centers, m is the number of customer classes), where l_{ij} represents the load (time in seconds) that puts the customer j on the job center i . The loading matrix can be transformed to a m dimensional space, where each axis represents one customer class, and job centres ($J1$ – $J4$) are points in this space that indicate, how each customer class loads the particular job centre. The potential system bottlenecks ($J1$ and $J2$) lie on the convex hull of this point (and their projections on all axes) (see Fig. 5a). This plot can be transformed to frequency space of customer arrivals (see Fig. 5b). The current state of the system is represented as a single point P and the distance to the hyperplanes ($H1$, $H2$, $H3$) indicates the maximal frequency increase before a job centre gets saturated. In other words, the system that is in state P can increase the arrival frequency of *Customer 1* by $dC1_{H2}$ ($\min(dC1_{H2}, dC1_{H3})$) or the arrival frequency of *Customer 2* by $dC2_{H1}$ ($\min(dC2_{H1}, dC2_{H2})$).

5 Conclusions and Future Work

This paper summarizes the three important and inter-related problems in the area of simulation model design for industrial mechatronic systems. The paper addresses the research issue dealing with integrating heterogeneous engineering knowledge for simulation model design. It also addresses cutting simulation models into a set of simulation modules. The paper proposes a methodology for designing glue modules integrating the simulation modules. Finally, the paper addresses the problem of performance analysis, which is crucial for analysing computational performance of the sets of coupled modules within the simulation workflow. The main contribution is the application of the extended bond graph

theory for designing the structure of glue modules for integration of simulation modules. As well, the paper contributes to improve computation time required for simulation as it proposes to utilize the Bounded Analysis for finding modules being bottlenecks in complex simulation workflows. The results of this paper can be used for decomposing a simulation into several modules for physical systems. The paper also explains how this decomposition can be performance-aware. The proposed approach was validated based on three software prototypes and based on conceptual studies analysing the entire proposed methodologies. In future work, we plan to propagate the results of performance analysis back to the semantic integration of heterogeneous engineering sources automatically and to match the weak performance with the original plan data.

Acknowledgments This work was supported by the Christian Doppler Forschungsgesellschaft, the Federal Ministry of Economy, Family and Youth, and the National Foundation for Research, Technology and Development—Austria; and by the Grant Agency of the Czech Technical University in Prague, grant No. SGS12/188/OHK3/3T/13.

References

1. Cortellessa, V., Marco, A., Inverardi, P.: Model-based software performance analysis. Springer, Berlin (2011)
2. Gawthrop, P., Bevan, G.: Bond-graph modeling. *IEEE Control Syst. Mag.* **27**, 24–45 (2007)
3. Goh, C. H.: Representing and reasoning about semantic conflicts in heterogeneous information systems. Ph.D. thesis, MIT (1996)
4. Ichise, R.: Machine learning approach for ontology mapping using multiple concept similarity measures. In: Proceedings of the 7th IEEE/ACIS International Conference on Computer and Information Science. IEEE (2008)
5. Jirkovsky, V., Ichise, R.: MAPSOM: user involvement in ontology matching. In: Proceedings of the 3rd Joint International Semantic Technology Conference (JIST) (2013)
6. Novák, P., Šindelář, R.: Component-based design of simulation models utilizing bond-graph theory. In: To appear in 19th IFAC World Congress. 6 pages, Cape Town (2014)
7. Novák, P., Melik-Merkumians, M., Steinegger, M., Moser, T., Šindelář, R., Zoitl, A.: Extraction of automation system engineering knowledge for mapping plant and simulation interfaces. In: Borangiu, T., Thomas, A., Trentesaux, D. (eds.) *Service Orientation in Holonic and Multi Agent Manufacturing and Robotics—Studies in Computational Intelligence* 472, vol. 3, pp. 247–261. Springer, Berlin (2013)
8. Novák, P., Šindelář, R., Mordinyi, R.: Integration framework for simulations and SCADA systems. *Simul. Model. Pract. Theor.* **47**, 121–140 (2014). ISSN 1569-190X
9. Noy, N.F., Doan, A., Halevy, A.Y.: Semantic integration. *AI Mag.* **26**, 7–9 (2005)