Towards enabling heterogeneous model inter-operation across abstraction levels ⋆

Marina Rantanen Modeer [*] **Christian Sonntag** [**]
Sebastian Engell [***]

[*] *Technische Universität Dortmund, (e-mail: marina.rantanen-modeer@ tu-dortmund.de).*
[**] *Technische Universität Dortmund, (e-mail: christian.sonntag@ tu-dortmund.de).*
[***] *Technische Universität Dortmund, (e-mail: sebastian.engell@ tu-dortmund.de).*

Keywords: Heterogeneous modelling, Model-based systems engineering, Model interaction, Abstraction

1. INTRODUCTION

Cyber Physical Systems (CPS) are inherently complex and incorporate interacting heterogeneous subsystems. Model based design of complex CPS is a multi-scope, multi-aspect challenge. In the design process of such systems, usually many different models are built. These models describe elements of the system at different levels of detail and under different assumptions, or the overall behaviour of the system in a coarse manner. Currently much effort is spent on testing and on the often costly correction of design flaws which could be reduced if an integrated model-based design approach were used. On the other hand, the required effort for building all-encompassing models of complex systems is often prohibitive, so the modeling efforts should be focused on those aspects which are crucial for the correct interaction of the elements and system function. Therefore a methodology for the integration of heterogeneous models on different levels of detail and abstraction is needed.

Based on available literature it is evident that there are currently two main approaches to this integration - co-simulation frameworks and abstractions of component models. Co-simulation frameworks enable the simulation of numerous inter-connected sub-models, but assume models on the same level of detail to enable a tight coupling. On the other hand, e.g. Sabetta et al. (2005) suggest transformations of component models into abstractions, assuming model components based on the Unified Modeling Language (UML). For a heterogeneous set of models on different levels of detail and different levels of abstraction that are formulated using different modelling concepts, another approach must be taken. The European project MULTIFORM (Hüfner et al. (2012) and Moneva et al. (2011)) introduced a design framework for the management of heterogeneous CPS models. The work presented here builds on ideas from MULTIFORM. We introduce the idea of generating abstractions of detailed models and simulations and embedding them into models that are set up in a different formalism. Useful information can thus be conveyed in a simplified manner from one model to the other.

2. COMPREHENSIVE CPS MODELING USING PARTIAL MODELS

The following hypotheses are assumed based on typical industrial development practice:

(1) Models are built to answer specific design questions and serve different purposes during the different design stages of a CPS. These models therefore cover different elements, aspects and scopes of the system being developed which gives rise to a large amount of models that are expressed in different formalisms using specialized software tools and languages. Together they constitute a partial representation of the system as a whole.
(2) Information which is properly propagated across the abstraction hierarchy can efficiently support the development of a CPS by facilitating and enabling design activities that are dependent on information generated by earlier or parallel investigations.

In typical design sequences, information is propagated from one design model to another in an ad hoc manner analysing and abstracting simulation, optimization or verification results without algorithmic support. To enable model management and information propagation using a computer based approach, model abstractions that provide a simplified representation of a detailed component model or of its behavior from one model to the other should be generated. For instance, detailed models that are developed to provide information corresponding to what is directly observable in a real system, i.e. comparable to measurements of an implementation of the system or component could be stripped to the essentials for other design activities. Such an abstraction has the potential to support model inter-operation between different levels and horizontally on the same level of abstraction. This approach also enables design tasks that would otherwise

⋆ The presented work has received funding from the European Union's Horizon 2020 Framework Programme for Research and Innovation under grant agreement no 674875.

not be possible, i.e. a design activity can embed information that has not be accessible and thereby can include simulations that were not possible earlier. This approach will first identify model dependencies, and then fit one or several models in the target modelling formalism to simulated data.

3. USE CASE: WATER LOCK DESIGN

The challenges discussed above are described through a pertinent use case that was carried out together with a large European organization that commissions, develops and maintains water locks. The organization uses an extensive model-based approach in all major design steps. Currently model interactions are heavily dependent on informal exchanges of information between model designers, and model management is mainly documentation based. In the overall water lock development process a large set of tools are used as there are specialized tools available for each specific design task.

Various design alternatives are evaluated based upon coarse, system wide models. After a concept has been decided upon, numerous Computational Fluid Dynamic (CFD) simulations are carried out with the ultimate goal to determine the filling and emptying times based on surrounding conditions and the specific canal and lock component dimensions. The simulations aim at capturing worst-case scenarios in order to formulate conservative design requirements. Subsequent activities involve mechanical, electrical and control systems design.

In the following, we discuss the interaction between a lock specific CFD model and a model that is used to synthesize supervisory controllers for water locks using hybrid automaton models of the lock operations. These are developed in the open source tools LOCKSIM and the Compositional Interchange Format (CIF) (Reniers et al. (2017)). The supervisory controller that is generated on the basis of the CIF operations model developed by Reijnen et al. (2017) is validated using a hybrid-automaton plant model that was also developed in CIF. The plant model is based on a number of assumptions about the hydrodynamic behaviour - assumptions that are based on lock design requirements. Only static information is communicated through this strategy and the model is therefore disregarding hydrodynamic conditions and events that could potentially affect the emptying and filling times that are relevant for the validation, such as an abundance or deficiency of rain water and the associated soil conditions.

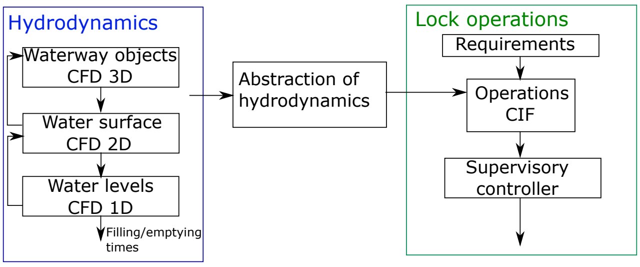

Fig. 1. Concept of supporting design activities by embedding abstracted models that result from other simulation studies.

Inside the plant model developed by Reijnen et al. (2017), the evolution of the water level is currently described by a differential equation (DE) in a separate automaton. This DE provides a coarse approximation of the evolution of the change of the water level based on sensor readings of the lock paddle positions and culvert flows. Extending the plant model with another set of automata that contain different dynamic models extracted from the CFD model, would allow the supervisory control designer to carry out more accurate validations and to update the requirements accordingly. Figure 1 illustrates the suggested generation and embedding of abstractions.

The CFD simulations clearly showed that emptying and filling times are dependent on the state of the environment. If for instance the ground is saturated with water due to rain, water flow into and out of the water lock chamber is significantly affected. To create an abstraction of the CFD model, a number of simulations are run with different inputs to cover the range of probable water conditions relevant to the lock operations model. Simulation results are then approximated by a set of models, for which the situation can be assumed to change very slowly compared to the time scale of the reaction of the supervisory controller. These different approximate models are represented as automata in the CIF plant model and are connected one by one to the overall model to generate new simulated sensor readings of the evolution of the water levels depending on given sensor inputs on lock paddle positions and culvert flows from the plant model.

4. CONCLUSION

This paper introduced the idea of generating abstractions of detailed design models to embed into models expressed in other formalisms during the development of complex CPS. The idea is described through a use case of water lock development in which detailed hydrodynamic models can be embedded into hybrid automata developed for the generation of supervisor controllers.

REFERENCES

Hüfner, M., Fischer, S., Sonntag, C., and Engell, S. (2012). Integrated model-based support for the design of complex controlled systems. In *Proc. European Symposium on Process Systems Engineering*, volume 15, 19–23.

Moneva, H., Hamberg, R., and Punter, T. (2011). A design framework for model-based development of complex systems. In *32nd IEEE Real-Time Systems Symposium 2nd Analytical Virtual Integration of Cyber-Physical Systems Workshop, Vienna.*

Reijnen, F., Goorden, M., Mortel-Fronczak, J.v.d., and Rooda, J. (2017). Supervisory control synthesis for a waterway lock. *2017 IEEE Conference on Control Technology and Applications (CCTA).*

Reniers, M., van de Mortel-Fronczak, J., and Roelofs, K. (2017). Model-based engineering of supervisory controllers for cyber-physical systems. In *Industrial Internet of Things*, 111–136. Springer.

Sabetta, A., Petriu, D.C., Grassi, V., and Mirandola, R. (2005). Abstraction-raising transformation for generating analysis models. In *International Conference on Model Driven Engineering Languages and Systems*, 217–226. Springer.