System definition versus problem representation for system design

Pierre-Alain Yvars[1] and Laurent Zimmer[2]

[1]ISAE-Supméca, Quartz, Saint Ouen, France pierre-alain.yvars@isae-supmeca.fr (corr. author)
[2]Dassault-Aviation, Direction de la prospective, Saint-Cloud, France

Keywords: system modelling, problem modelling, analysis, simulation, synthesis, resolution.

1 Introduction

Our focus here is on the design of technical systems, whether physical (mechanical, electronic, electrical, etc.), software-intensive (embedded systems) or mixed (cyber-physical systems). In (Yvars and Zimmer, 2021) and (Pinquié and al, 2023), we find a proposed classification of technical system design problems, on which we will base ourselves. In terms of formalization, two complementary visions are possible: the definition of the system and the formalization of the design problem. In this paper, we propose to examine the founding principles of these two approaches, their differences and complementarity, and their positioning in the design process.

What all system definition formalisms have in common is that they represent a system architecture produced by the designer, with no guarantee as to its admissibility. Once the system has been modelled, it must be evaluated according to the point of view adopted by the language, so as to verify a posteriori the requirements considered. Some formalisms are generalist, while others are more specific to the system and/or the requirements to be verified. From the point of view of INCOSE Vision 2035 (Incose, 2021), they are used to formalize the result of the architecture activity. They include general formalisms from MBSE, such as SysML (SysML, 2019) or SysMLV2 (SysMLV2, 2017), as well as formalisms dedicated to an evaluation objective (e.g. Modelica (Fritzson, 2014), Event-b (Abrial, 2010) or S2ML (Batteux et al, 2015)) or to specific systems (e.g. AADL (aadl, 2017) for embedded systems).
Formalisms that are dedicated to the problem representation are better suited to the system architecture design stage, as they automatically produce pre-dimensioned and configured architectures that necessarily satisfy the requirements expressed. Some of them comes from the Operations Research community. They are called Algebraic Modelling Languages (AML) and provide a flat representation of a problem to be solved, using variables, matrices, vectors, equations and/or inequalities to be satisfied. They also enable the mathematical formalization of an objective function to be optimized. OPL (Opl, 2022), AMPL (Fourer et al, 2003) and GAMS (Rosenthal, 2007) are the examples of AMLs. These languages call efficient solvers, but they lack structuring and abstraction capabilities that are crucial for system design problem. More recently, two new formalisms have emerged that provide a structured representation of the elements of the problem to be solved: Clafer (Bak et al, 2014) is dedicated to the representation of software product line problems. DEPS (Yvars and Zimmer, 2024) is a declarative problem modelling language, structured around models and properties. All these formalisms also allow the use of appropriate solvers and/or optimizers to solve the expressed problem.

2 Methods

To illustrate our goal, we propose to base our presentation on a case study of an embedded camera design taken from the literature (Leserf et al, 2015). Two approaches will be developed. On the one hand, design by analysis, by modelling with SysML a definition of a camera candidate system and then evaluating its performance against expected requirements, on the other hand, the synthesis approach, in which we model the camera design problem and show how it is possible to automatically generate one or more solutions that are correct by construction, i.e. that necessarily satisfy the requirements. DEPS is used to model the design problem of the camera. The two whole processes will be detailed form the modelling step to the result step (i.e. the camera final definition).

3 Results

The results of the study are presented from several points of view for each of the two approaches:

- Design problem categories: what kind of system design can be addressed by the approach ?
- Methodological: what are the methodological principles underlying the approach ?
- Formalism: what are the founding myths of the modeling languages associated with the approach ?
- Model granularity: what model granularity does the approach support ?
- Solution computation methods: what are the numerical methods used to compute the proposed solutions ?
- Quality and diversity of solutions: what are the capacity of the approach to allow the designer a design space exploration ?
- Design step concerned: which stage of the system design process is concerned with the approach ?

4 Discussion

Far from being opposites, the two approaches are complementary. One of the fundamental issues to be discussed will be the interoperability of the two approaches. How to combine synthesis and analysis in such a way that we can define a more efficient system design process ?

References

AADL (2017). https://www.sae.org/standards/content/as5506c/

Abrial, J.R. (2010*). Modeling in Event-B: System and Software Engineering*. Cambridge Press

Bąk K., Diskin Z., Antkiewicz M., Czarnecki K. & Wąsowski A.(2014*). Clafer: Unifying class and feature modeling.* Software and Systems Modeling.

Batteux M., Prosvirnova T., Rauzy A. (2015), *System Structure Modeling Language (S2ML).* https://hal.science/hal-01234903/document

Fourer R., Gay D.M., & Kerdighan D.W. (2003). *AMPL A language for mathematical programming*. Duxbury & Thomson. https://ampl.github.io/ampl-book.pdf

Fritzson P. (2014). Principles of Object-Oriented Modeling and Simulation with Modelica. Wiley.

Incose se-vision (2021). https://www.incose.org/about-systems-engineering/se-vision-2035

Leserf, P. de Saqui-Sannes P., Hugues J., and Chaaban K. (2015). *Sysml modeling for embedded systems design optimization: A case study*. In Proc of 3rd International Conference on Model-Driven Engineering and Software.

Opl (2022). https://www.ibm.com/docs/fr/icos/22.1.1?topic=reference-about-language-quick

Pinquié R., Roucoules L., Yvars P.A., Chenouard R.(2024). *Operationalizing Community-Based Open Scientific Design Research Benchmarks: Application to Model-Based Architecture Design Synthesis"*, proc of the 18th International Design Conference, DESIGN2024, Dubrownik, Croatie.

Rosenthal R.E. (2007) *GAMS a users's guide.* GAMS Development Corporation. Washington

SysML (2019). https://www.omg.org/spec/SysML/1.6/.

SysMLV2 (2017) https://www.omgsysml.org/SysML-2.htm

Yvars P.A., Zimmer L. (2022). *Toward a correct by construction design of complex systems : The MBSS approach*, *Procedia CIRP,* Vol 109, Elsevier.

Yvars P.A., Zimmer L. (2024). *DEPS: A Model and Property Based Language for the synthesis of complex systems*, International Journal of Software and Systems Modeling (SoSym) (23). https://doi.org/10.1007/s10270-023-01129-x