

Translator from Extended SysML to Physical Interaction and Signal Flow Simulation Platforms

Raphael Barbau^{2,1}, Conrad Bock¹, and Mehdi Dadfarnia¹

¹National Institute of Standards and Technology,
Gaithersburg, MD 20899 USA

²Engisis LLC,
Bethesda, MD 20817 USA

raphael.barbau@nist.gov
conrad.bock@nist.gov
mehdi.dadfarnia@nist.gov

Software DOI: <https://doi.org/10.18434/M32010>

Key words: lumped-parameter simulation; model-based systems engineering; model transformation; SysML.

Accepted: June 17, 2019

Published: July 2, 2019

<https://doi.org/10.6028/jres.124.017>

1. Summary

Designing complex systems often requires engineers from multiple disciplines (mechanical, electrical, production, and so on) to communicate with each other and exchange system design information. Systems engineering models are a cross-disciplinary foundation for this process, but are not well-integrated with specialized engineering information, leading to redundant and inconsistent system specifications. The software provided here translates system models in the Systems Modeling Language (SysML) [1] to physical interaction and signal flow (also known as lumped-parameter, one-dimensional, or network) files on two simulation platforms used in many engineering domains.

The translator implements the SysML Extension for Physical Interaction and Signal Flow Simulation (SysPhS), published by the Object Management Group [2]. It can generate Modelica¹ or Simulink/Simscape files from SysML models extended according to the SysPhS specification.

Several example models are provided to demonstrate the translator in various engineering domains.

¹Certain commercial equipment, instruments, or materials are identified in this paper to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

2. Software Specifications

| | |
|-----------------------------|---|
| NIST Operating Unit | Engineering Laboratory, Systems Integration Division |
| Category | System model translator |
| Targeted Users | Systems engineers, simulation engineers |
| Operating Systems | See Java 6 systems requirements |
| Programming Language | Java 6 |
| Inputs/Outputs | Input: SysML models creating using the SysPhS profile Output: Modelica or Simulink file corresponding to the model |
| Documentation | Instructions on how to run the translator are provided in the software package. The translator is an implementation of the SysPhS specification, available at https://www.omg.org/spec/SysPhS/1.0 . The source code of the translator is available at https://doi.org/10.18434/M32010 |
| Accessibility | N/A |
| Disclaimer | https://www.nist.gov/director/licensing |

3. Software Details

The software translates SysML models extended with SysPhS into simulation files for Simulink/Simscape and Modelica, per the SysPhS specification. The SysML models must be serialized in XML Metadata Interchange file format (XMI) [3]. The software deserializes these files into instances of the SysML metamodel using the Eclipse Modeling Framework (EMF) library and its extension for the Unified Modeling Language (UML) [4], on which SysML is based. EMF was used to create metamodels of the Simulink/Simscape and Modelica languages. Java libraries were automatically generated from these metamodels. These libraries are used to instantiate the metamodels based on the input SysML models. The result is serialized into simulation files that can be opened by their simulation platforms. Additional explanation is available in [5, 6].

The translator also supports reverse translation, from Simulink/Simscape and Modelica files generated by the translator to extended SysML.

4. Methods for Validation

A set of SysML example models are included to demonstrate the translator. These sample models cover multiple aspects of SysPhS:

- An electrical circuit model demonstrates how to use physical interaction modeling in the electrical domain
- A hydraulics model demonstrates how to use physical interaction modeling in the hydraulic domain
- A signal processor model (with low-pass and high-pass filter) demonstrates how to use signal-flow modeling
- A humidifier system model demonstrates how to use signal-flow modeling to describe the operation of a humidifier in a room. The model also includes simple state machines.

- A cruise controller model demonstrates a mixed usage of physical interaction and signal-flow. The signal flow portion covers speed sensors and engine actuators, while the physical interaction portion covers the flow of mechanical energy that powers the car and makes it move.

All these models were translated into Modelica and Simulink/Simscape files that simulate successfully, showing how the systems will behave over time.

Acknowledgments

The authors wish to thank Ion Matei and Nerijus Jankevicius for their contributions to the mapping between SysML and simulation platforms.

5. References

- [1] Object Management Group (2015) Systems Modeling Language, version 1.5.
- [2] Object Management Group (2018) SysML Extension for Physical Interaction and Signal Flow Simulation, version 1.0.
- [3] Object Management Group (2015) XML Metadata Interchange, version 2.5.1.
- [4] Object Management Group (2015) Unified Modeling Language, version 2.5.
- [5] Bock C, Barbau R, Matei I, Dadfarnia M (2018) An extension of the systems modeling language for physical interaction and signal flow simulation. *Systems Engineering* 20(5):395–431. <https://doi.org/10.1002/sys.21380>
- [6] Dadfarnia M, Bock C, Barbau R (2016) An improved method of physical interaction and signal flow modeling for systems engineering. *Conference on Systems Engineering Research (CSER 2016)*. Available at <https://www.nist.gov/publications/improved-method-physical-interaction-and-signal-flow-modeling-systems-engineering>

About the authors: *Raphael Barbau is a senior consultant at Engisis LLC, and a Computer Scientist in the Systems Integration Division at NIST. He has been involved in various projects aimed at improving the interoperability among systems engineering tools.*

Conrad Bock is a Computer Scientist in the Systems Integration Division at NIST. He leads the Model-Based Systems Definition and Analysis Integration for Smart Manufacturing in the Model-Based Enterprise Program, and chairs the SysML Extension for Physical Interaction and Signal Flow Simulation Revision Task Force at the Object Management Group.

Mehdi Dadfarnia is an Electrical Engineer in the Systems Engineering Group under the Systems Integration Division at NIST. He is currently pursuing a PhD in Mechanical Engineering at the University of Maryland, College Park, where he also previously received a B.S. in Electrical Engineering and an M.S. in Systems Engineering. He has been working on the interoperability of systems structure and behavior models with simulation tools.

The National Institute of Standards and Technology is an agency of the U.S. Department of Commerce.