The Journal of Open Source Software

PySD: System Dynamics Modeling in Python

Eneko Martin-Martinez [1*], **Roger Samsó** [1*¶], **James Houghton** [2], and **Jordi Solé** [1,3]

1 CREAF, Centre de Recerca Ecològica i Aplicacions Forestals, E08193 Bellaterra (Cerdanyola del Vallès), Catalonia, Spain **2** Computational Social Science Lab, University of Pennsylvania, Philadelphia PA, 19104, United States of America **3** Departament de Dinàmica de la Terra i l'Oceà, Universitat de Barcelona (UB) E08007, Catalonia, Spain ¶ Corresponding author * These authors contributed equally.

DOI: 10.21105/joss.04329

Software

- Review
- Repository
- Archive

Editor: Pierre de Buyl

Reviewers:

- @blsqr
- @sixpearls

Submitted: 25 March 2022
Published: 17 October 2022

Summary

System Dynamics (SD) is a mathematical approach used to describe and simulate the dynamics of complex systems over time. The foundations of the methodology were laid in the 1950s by Professor Jay W. Forrester of the Massachusetts Institute of Technology (Forrester, 1971). The building blocks of SD models are stocks, flows, variables, parameters, and lookup tables. Stocks represent cumulative quantities which take a certain value at each moment in time (integral), and flows are the rates at which those quantities change per unit of time (derivative). Variables express intermediate calculations, parameters set external conditions for the simulation, and lookup tables are single-valued functions that accept another model component as an argument. These components can combine to create feedback loops in which the state of the stock variables feeds back to influence flows in the model. The relationships between these model components can be represented using causal loop diagrams, see Figure 1.

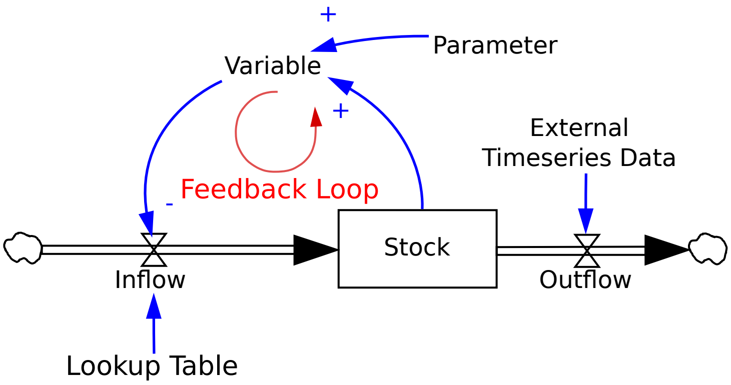

Figure 1: An example Causal Loop Diagram showing how the various components of system dynamics models are visualized by domain-specific modeling environments.

Since its inception, the SD methodology has been applied in different areas, including manufacturing, energy, climate, population, ecology and economics (Fetene Adane et al., 2019; Harvey, 2021; Moallemi et al., 2021; Sterman et al., 2012). In the 1990s, the popularity of the methodology gave rise to the development of several visual programming systems for building SD models. These domain-specific modeling environments were widely adopted by the SD community due to their convenient graphical user interfaces. Stella® and Vensim®, though not open-source, are two of the most popular commercial system dynamics modeling tools, but many others exist (see Wikipedia (2022);SDXorg (2021)).

PySD is a Python library that transpiles models built in Stella®'s or Vensim®'s domain-specific languages into Python, allowing the user to load, parametrize and execute SD models, and to take advantage of the extensive data-science capabilities of the Python ecosystem. PySD was first released in September 2014 by James Houghton (Houghton & Siegel, 2015).

The main functional elements of PySD are 1) a set of parsing expression grammars (PEGs) (and their corresponding node visitor classes) to parse models built using Stella® and Vensim® (in *.xmile* and *.mdl* formats, respectively); 2) isomorphic implementations of the most frequently used Stella® and Vensim® built-in functions and other basic operations; 3) a builder, to write the parsed model code in Python and 4) a fordward Euler solver to run the models.

In addition to the aforementioned core functionality, PySD also allows users to import model inputs from external sources (from spreadsheet files), modify model variables at runtime, split models into any number of modules and submodules (corresponding to Vensim® views), isolate parts of a model to be run individually, store intermediate simulation results and resume the simulation from that particular state, and run models using PySD alone (without Stella® or Vensim®). All these features are made available to the user through a command-line interface.

Despite its maturity, PySD is currently in a very active development phase, and a proof to that is that most of the extra features listed in the paragraph above were implemented in the 21 months that separate releases *v0.11.0* and *v3.7.0*.

The main novelty introduced in release *v3.0.0* is that the parsing process is now isolated from the building process, which opens the door to the development of additional model builders (in Python) to write the output models in any programming language. This has been achieved by creating an intermediate abstract model representation (AMR), as shown in Figure 2, that embeds all the information contained in the original models, plus additional information such as the order of operations, in pure Python objects.

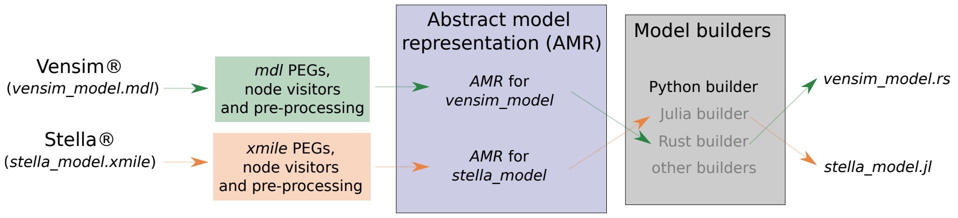

Figure 2: New model parsing-building logic in PySD v3.0.0. PEG acronym stands for Parsing Expression Grammar.

Future releases will focus on cutting down simulation times and on including additional built-in Stella® and Vensim® functions. The most relevant performance-oriented development will be the migration from xarray to numpy to perform array operations, which in addition to the expected overhead reduction will also allow for further potential model optimizations (i.e. JIT compilation).

Statement of need

Stella® and Vensim® are excellent tools for model conceptualization and design using causal loop diagrams. Their solvers are easy to parametrize, efficient (implemented in C and C++) and thoroughly tested. However, the Python ecosystem offers powerful open-source tools for data visualization, sensitivity analysis, graph theory, machine learning and other data analysis tasks that are unavailable in the domain-specific modeling environments. Most importantly, though models created using Stella® and Vensim® can be exported into text format (*.xmile* and *.mdl*, respectively), users must install propietary software in order to execute the models and visualize the results.

PySD was designed to supplement the capabilities of the domain-specific modeling tools by bringing their outputs into the larger Python data analytics ecosystem. This approach allows users to integrate their models with the most up-to-date analytics tools, and allows the large Stella® and Vensim® user-base to make their models fully open-source and sharable.

PySD is in use by hundreds of modelers and data scientists in academia, industry, and government; and has been used in over two dozen academic publications. The latest improvements to the library have taken place in the context of the European H2020 projects MEDEAS (MEDEAS, 2016) and LOCOMOTION (LOCOMOTION, 2019), in which several authors of the present work participate. The MEDEAS project ended in early 2020, and aimed at developing an Integrated Assessment Model (IAM), named *pymedeas* (Samsó et al., 2020; Solé et al., 2020), to analyze different energy transition scenarios under biophysical constraints (e.g. climate change and resource availability). The LOCOMOTION project, which is still ongoing (2019-2023), aims to build a new and more complex IAM, departing from the one developed during the MEDEAS project. In MEDEAS the model was built using Vensim®, and later translated to Python using PySD, and the same approach is being used in the LOCOMOTION project.

Acknowledgements

This work and part of the development of PySD was supported by the European Union through the funding of the MEDEAS and LOCOMOTION projects under the Horizon 2020 research and innovation programme (grant agreements No 69128 and 821105, respectively), and by the Government of Catalonia through the contract programme between the Ministries of Territory and Sustainability and Business and Labour and the Centre for Research on Ecology and Forestry Applications (CREAF), approved by Government Agreement on March 16, 2021. The authors of this paper would like to acknowledge all contributors from the SD community, which have helped improve PySD and kept the project alive for the 8 years since it was created.

References

Fetene Adane, T., Bianchi, M. F., Archenti, A., & Nicolescu, M. (2019). Application of system dynamics for analysis of performance of manufacturing systems. *Journal of Manufacturing Systems*, *53*, 212–233. https://doi.org/10.1016/j.jmsy.2019.10.004

Forrester, J. W. (1971). *World dynamics* (pp. xiii, 142 p.). Wright-Allen Press Cambridge, Mass.

Harvey, J. T. (2021). Feedback economics: Economic modeling with system dynamics (2021). *System Dynamics Review*, *37*(4), 363–366. https://doi.org/10.1002/sdr.1695

Houghton, J. P., & Siegel, M. (2015). Advanced data analytics for system dynamics models using PySD. *Proceedings of the 33rd International Conference of the System Dynamics Society*, *2*, 1436–1462. ISBN: 9781510815056

LOCOMOTION. (2019). *H2020 LOCOMOTION project website*. Available at https://www.locomotion-h2020.eu/, (2022/01/19).

MEDEAS. (2016). *H2020 MEDEAS project website*. Available at https://medeas.eu/, (2022/01/19).

Moallemi, E. A., Bertone, E., Eker, S., Gao, L., Szetey, K., Taylor, N., & Bryan, B. A. (2021). A review of systems modelling for local sustainability. *Environmental Research Letters*, *16*(11), 113004. https://doi.org/10.1088/1748-9326/ac2f62

Samsó, R., de Blas, I., Perissi, I., Martelloni, G., & Solé, J. (2020). Scenario analysis and sensitivity exploration of the MEDEAS europe energy-economy-environment model. *Energy Strategy Reviews*, *32*, 100582. https://doi.org/10.1016/j.esr.2020.100582

SDXorg. (2021). *SD tools: List of open source system dynamics tools*. Available at https://github.com/SDXorg/SD-Tools, (2022/01/19).

Solé, J., Samsó, R., García-Ladona, E., García-Olivares, A., Ballabrera-Poy, J., Madurell, T., Turiel, A., Osychenko, O., Álvarez, D., Bardi, U., Baumann, M., Buchmann, K., Capellán-Pérez, I., Cerný, M., Carpintero, O., Blas, I. de, Castro, C. de, Lathouwer, J.-D. de, Duce, C., … Theofilidi, M. (2020). Modelling the renewable transition: Scenarios and pathways for a decarbonized future using pymedeas, a new open-source energy systems model. *Renewable and Sustainable Energy Reviews*, *132*(110105). https://doi.org/10.1016/j.rser.2020.110105

Sterman, J., Fiddaman, T., Franck, T., Jones, A., McCauley, S., Rice, P., Sawin, E., & Siegel, L. (2012). Climate interactive: The c-ROADS climate policy model. *System Dynamics Review*, *28*(3), 295–305. https://doi.org/10.1002/sdr.1474

Wikipedia. (2022). *Comparison of system dynamics software — Wikipedia, the free encyclopedia*. http://en.wikipedia.org/w/index.php?title=Comparison%20of%20system%20dynamics%20software&oldid=1043139275, (2022/01/19).