Towards a novel multi-purpose simulation software of water distribution systems in Python

David B Steffelbauer, Olivier Piller, Camille Chambon, Edo Abraham

14th International Conference on Hydroinformatics
Water INFLUENCE – Water INFormatic soLUtions and
opEN problems in the cycle from Clouds to ocEan
4-8 July 2022, Bucharest, Romania

Towards a novel multi-purpose simulation software of water distribution systems in Python

David Steffelbauer[1,*], **Olivier Piller**[2,3,*], **Camille Chambon**[2], **Edo Abraham**[4]

[1] Department of Civil and Environmental Engineering, NTNU, S.P. Andersens Veg 5, 7031 Trondheim, Norway

[2] INRAE, ETBX Research Unit, Aqua Department, F-33612 Cestas, France

[3] School of Civil, Environmental and Mining Engineering, University of Adelaide, Adelaide SA 5005, Australia

[4] Water Management Department, TU Delft, Stevinweg 1, 2628 CN Delft, Netherlands

* Corresponding authors: david.steffelbauer@ntnu.no, olivier.piller@inrae.fr

Short abstract. A novel object-oriented modelling framework for Water Distribution Systems (WDSs) is presented that is suited for multiple purposes (asset management, hydraulic simulations, demand modeling, etc.). Additionally, this work provides efficient standard structures for storing WDS data (and metadata) and motivates why Python is the future programming language for WDS modeling.

Keywords. hydraulic modeling, system design, asset management, object-oriented programming, stochastic demand modelling.

1. Introduction

Mathematical models are used in water distribution system (WDS) analysis for a wide range of applications, including system design tasks, asset management, operational management, or water demand estimation. In the past, divergent software solutions had emerged for these purposes, often tailored to different needs and requiring different levels of detail based on their scope. This resulted in differing representations of a WDS, although the underlying physical system is the same.

Sometimes, the need for a hybrid representation arises. To give an example, for risk assessment, asset management software must be coupled with hydraulic models to simulate the probability and impact of pipe failures. Vice versa, water pressure and velocities affect the lifespan of pipes.

To solve these hybrid problems, many tailor-made solutions were produced by interfacing two or more programs. Usually, the output of one model is transformed to fit the input format of another model, which often results in an information loss. Real unification of different computer models has been hampered by the structure of the different software solutions. Due to historic reasons, most computer solutions are utilizing procedural code paradigms and are dependent on custom-made (sometimes even not standardized) input file formats. Without doubt, the WDS community would tremendously benefit from a real unified approach, which goes beyond the production of more task-specific interfaces between different solutions.

To reach this goal, we have to (i) rethink the way we represent WDS in a computer program to increase genericity; and (ii) we should not be afraid of rewriting old procedural code from scratch to produce a novel and innovative representation of WDSs. This novel representation should be object-oriented since this has huge benefits over procedural code; some of which are, higher flexibility, reuse through inheritance, or higher modularity that enables remote collaboration amongst multiple developers.

Finally, we have to think about a programming language that (i) is capable of designing object-oriented software, (ii) has the ability to solve a wide range of problems (fast numeric for hydraulic simulations, stochastic programming for demand and failure modeling, optimization algorithms for optimal system control, efficient implementation of graph theoretic algorithms, artificial intelligence, ...) and (iii) has to be easy to learn and use so that it can attract a large community of researchers and professionals. This work introduces this novel object-oriented structure for WDS modelling.

2. Materials and methods

Through an extensive literature review on WDS software, advantages and drawbacks of different simulation tools have been identified. We specifically focused on object-oriented implementations as well as open-source software. Table 1 gives a non-extensive list of software tools. From that, we tried to identify the best approach for WDS modeling. Figure 1 shows the structure of the different base classes of the implementation of our solution.

Table 1. A non-extensive list of WDS software tools (OO=object oriented, Y=Yes, N=No), programming languages are in brackets.

Name	OO	Purpose	Comments
EPANET	N	Hydraulics; Water quality	Industry standard (C)
OOTEN	Y	Hydraulics	OO-layer on top of EPANET (C++)
Porteau	Y	Hydraulics; Water quality; Design	Designed with a CASE tool (Java/C++)
Casses	Y	Asset management	Pipe failure modeling (Java)
OOPNET	Y	Hydraulics; Water Quality	Based on EPANET (Python)
SIMDEUM	N	Demand modelling	Stochastic End-Use model (MATLAB)
WNTR	Y	Resilience under disaster scenarios	Based on EPANET (Python)

3. Why Python?

Python is one of the most-widely used high-level programming languages designed for code readability [1] and it allows procedural, object-oriented and functional programming. Recently, Python has gained immense popularity, which is reflected in various reputable rankings, e.g., “PYPL - Most popular language 2019” [2] or “Top programming language 2019” [3]. Python is used in multiple water related projects, e.g., EPANET’s new user interface, WNTR [4] and OOPNET [5]. Additionally, it is used within Q-GIS or for failure prediction modeling (lifelines), which are valuable tools for system management besides hydraulic modeling. Currently, Python lists 130.000 packages, many of them for machine learning, data analysis or scientific computing.

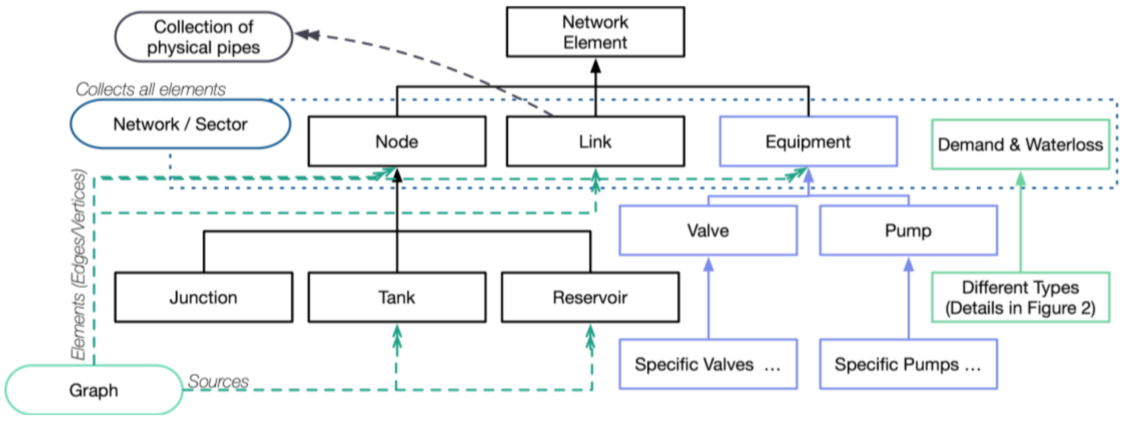

Figure 1. Class diagram of object-oriented structure.

4. Advantages of our implementation

Valves and pumps belong to the “Equipment” class. Links can be equipped with multiple instances of “Equipment” (like Porteau). In EPANET on contrary, valves and pumps are special cases of links. To equip a link with a valve, the link has to be broken in two parts, two extra nodes have to be added and a very short valve link is added. The “Equipment” approach leads to a more concise mathematical representation that improves system conditioning by grouping the resistances of links and valves so that the condition number of the iteration matrix in the solving method (e.g., GGA) is smaller or reduced.

Links (used for hydraulic simulations) are collections of a group of individual physical pipes, which facilitates interfaces to GIS and asset management software. The hydraulic properties of links (e.g., priors for roughness values) are derived from the physical properties of the underlying pipe objects (material, diameter, age, ...).

The novel implementation of demands and water losses (see Figure 2) allows more sophisticated simulation approaches, ranging from background leakages [6] to coupling hydraulic simulations and stochastic demand models [7][8].

Furthermore, prototypes of demand driven-, pressure dependent- and dynamic models are implemented to solve the hydraulics, including graph theoretic algorithms through an interface to Python’s networkx package. Data and metadata are stored using the open-standard file format JSON (JavaScript Object Notation), a lightweight human-readable data-interchange format.

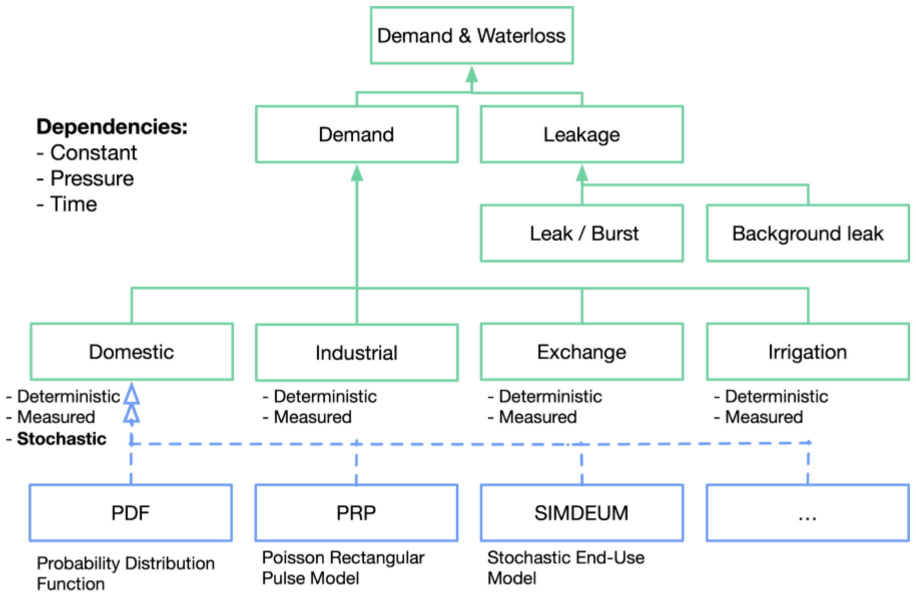

Figure 2. Class diagram for Demands & Water Losses.

References

[1] Peters T., 2004, PEP 20 -- The Zen of Python. Index of Python Enhancement Proposals (PEPs), 301–302. Retrieved from https://www.python.org/dev/peps/pep-0020/.

[2] Carbonnelle P., 2019, PYPL PopularitY of Programming Language index. De PyDatalog, 10–12. Retrieved from http://pypl.github.io/PYPL.html.

[3] IEEE Spectrum., 2019, The Top Programming Languages 2019. Retrieved October 30, 2019, from https://spectrum.ieee.org/computing/software/the-top-programming-languages-2019.

[4] Klise K.A., Murray R., Haxton T., 2018, An overview of the Water Network Tool for Resilience (WNTR), *In Proceedings of the 1st International WDSA/CCWI Joint Conference*, Kingston, Ontario, Canada, July 23-25, 075, 8p.

[5] Steffelbauer D.B. and Fuchs-Hanusch D., 2015, OOPNET: An object-oriented EPANET in Python. *Procedia Engineering*, 119, 710–718. https://doi.org/10.1016/j.proeng.2015.08.924.

[6] Chambon C., Piller O., and Mortazavi I., 2022, A new slow transient pressure-dependent model to simulate background leakages and inertia in water distribution networks. *Waterloss 2022; 2022 19 - 22 June 2022; Prague, Czech Republic: IWA*, 4p.

[7] Steffelbauer D., Blokker M., Knobbe A., and Abraham E, 2020, DASH of Water – water distribution system modelling in the age of smart water meters, *EGU General Assembly 2020*, Online, 4–8 May 2020, EGU2020-13439, https://doi.org/10.5194/egusphere-egu2020-13439, 2020.

[8] Creaco E., Blokker M., and Buchberger S., 2017, Models for Generating Household Water Demand Pulses: Literature Review and Comparison. *Journal of Water Resources Planning and Management*, **143**(6), 04017013.