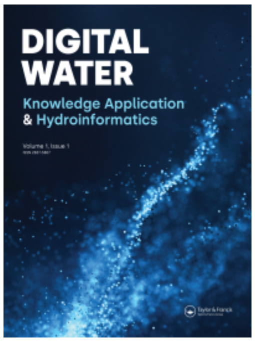
Taylor & Francis
Taylor & Francis Group



Digital Water
Knowledge Application & Hydroinformatics

ISSN: 2837-5807 (Online) Journal homepage: www.tandfonline.com/journals/tdwa20

Developing a graph-based, object-oriented urban water cycle simulation engine to explore circular water options

Dimitrios Bouziotas , Andreas Efstratiadis & Christos Makropoulos

To cite this article: Dimitrios Bouziotas , Andreas Efstratiadis & Christos Makropoulos (2026) Developing a graph-based, object-oriented urban water cycle simulation engine to explore circular water options, Digital Water, 4:1, 2725222, DOI: 10.1080/28375807.2026.2725222

To link to this article: https://doi.org/10.1080/28375807.2026.2725222

Published online: 02 Sep 2026.

DIGITAL WATER
2026, VOL. 4, NO. 1, 2725222
https://doi.org/10.1080/28375807.2026.2725222

Routledge
Taylor & Francis Group

RESEARCH ARTICLE

OPEN ACCESS Check for updates

Developing a graph-based, object-oriented urban water cycle simulation engine to explore circular water options

Dimitrios Bouziotas [a,b], Andreas Efstratiadis [a] and Christos Makropoulos[a]

[a]Department of Water Resources and Environmental Engineering, Urban Water Hydroinformatics Group, National Technical University of Athens, Athens, Greece; [b]Previsico, Advanced Technology Innovation Centre (ATIC), Loughborough, UK

ABSTRACT
This study demonstrates the development and use of a graph-based, object-oriented urban water cycle simulation engine able to model: (a) arbitrary information flows through graph edges and (b) arbitrary circular water technologies, either at the supply or the demand side, through graph nodes. The concept, built in Python, is demonstrated for the information flows of water quantity and (conservative) quality in a real-world application at a neighbourhood spatial scale in the Netherlands, where it delivers quantitative insights that agree with previous research works, while significantly reducing the complexity of earlier procedural modelling structures. Structural aspects and limitations are discussed, including slower runtimes but drastically improved code coherence and readability due to the use of OOP and high-level graph network management libraries. The application shows the potential of this engine to be used as a *modus universalis* for designing circular water options and understanding their role within the complex socio-technical urban water system.

ARTICLE HISTORY
Received 30 July 2026
Accepted 7 August 2026

1. Introduction

Modelling the complete urban water cycle is essential to understand how evolving urban regions interact with natural hydrological processes. Urbanisation directly affects multiple aspects of the hydrological cycle, as it increases impervious surfaces, alters natural flow channels with drainage networks of preset design standards, and requires drinking water demand to be covered by central infrastructure and water transfers across large distances, leading to a complex socio-hydrological system. Moreover, this system is vulnerable to future risks related to both socioeconomic drivers (e.g., population influx or demographic changes) (Bouziotas *et al.* 2015), as well as climate variability and change (Short *et al.* 2012), which results in increased stress from a possibly wetter (e.g., flash flooding) or drier (e.g., heat stress and dry spells) new normal. To study the interactions and vulnerabilities of such a system in an integrated manner, simulation frameworks can be used to quantify changes either at the demand side (i.e. clean water demands, uptake of circular water technologies) or the supply side (i.e., changes in central or distributed water infrastructure, climate-driven impacts to sources). These models also enable efficient scenario testing and act as decision-support systems for stakeholders to evaluate their vision of a circular water future (Bouziotas *et al.* 2023), or to quantify climate-change impacts and the effectiveness of climate-adaptation measures such as blue-green infrastructure (Almaaitah *et al.* 2021). In past studies, many of these models have been sectoral, focusing on limited interactions between different subsystems such as drainage, urban water treatment, or water consumption and demand (Bazrkar *et al.* 2016), while integrated models that incorporate all components of the urban water cycle have been developed in recent times (Peña-Guzmán *et al.* 2017). For integrated modelling applications, it is imperative to develop architectures that simulate multiple flows within the urban water cycle (including, but not necessarily limited to, drinking water (DW), wastewater (WW) and stormwater-runoff (SW)), while simultaneously flexibly allowing the multitude of water-aware, circular technologies and their manner of altering these flows to be factored in. This combination of multiple flows vis-à-vis multiple technologies enables the study of how different interventions, across different urban water streams,

CONTACT Dimitrios Bouziotas bouziot@mail.ntua.com National Technical University of Athens, Iroon Politechniou 5, Athens 15780, Greece

can be applied to shape robust and resilient urban water systems (Nikolopoulos *et al.* 2019, Bouziotas *et al.* 2023).

The introduction of circular water options with varying granularity and spatial context within the city is a complex management task, as these options can be introduced in a combined manner at multiple spatial scales (Bacchin *et al.* 2014), including, for instance, large recreational areas as well as smaller spatial units such as neighbourhoods or groups of households. Regardless of scale, the introduction of multiple circular water technologies leads to cascading effects on multiple, interconnected urban water-cycle domains that are commonly not evaluated or managed together. For instance, water-saving measures target the reduction of drinking water (DW) demands from the centralised supply system while ensuring (partial) coverage through local freshwater sources. As local and decentralised freshwater treatment can be challenging, many of these measures separate the quality characteristics of various consumptions and target lower-impact uses, such as the so-called grey-water use (Allen *et al.* 2010; Vuppaladadiyam *et al.* 2019) or irrigation. To supplement these demand-management measures, urban planners might opt for design options such as the introduction of rainwater harvesting (RWH) schemes to both enhance local freshwater security and increase local stormwater retention, leading to an altered (slower and lower) runoff response at the outlet, especially when RWH is combined with Sustainable Urban Drainage (SUDS) (Ferrans *et al.* 2022). Along with RWH and stormwater management, planners may also target locally produced wastewater (WW) from housing units or other uses, which can be locally sourced, treated and reused, thus increasing local water security for grey-water uses while simultaneously reducing the quantity of wastewater propagated downstream. WW treatment and reuse can be implemented with a variety of interventions, including grey-water-recycling (GWR) schemes for groups of households (Pidou *et al.* 2007), but also circular-economy options at the regional scale, such as sewer mining for the irrigation of urban green spaces (Makropoulos *et al.* 2018).

All of these multi-stream effects need to be part of an integrated performance assessment to allow for a multi-faceted evaluation of the system across different quantities of interest to all stakeholders. The evaluation typically focuses on multiple criteria important to the stakeholders, including the effectiveness of each solution for overall water management, as well as its energy footprint, initial investment and operational costs (Lai *et al.* 2008). Moreover, integration needs to reach beyond present-day urban water cycle settings and explore external driving factors, such as climate change, behavioural client shifts, demographic changes, capital and operational costs, etc., in order to account for various deep uncertainties and variability factors that might affect the urban water system. It is, indeed, common for stakeholder groups to want to project how their proposed interventions secure a better future in terms of water security, apart from alleviating present-day pressures (Bouziotas *et al.* 2023). To implement integrated assessment tools and methodologies, research questions therefore arise related to: (a) the capacity of said methods to factor in the effects of multiple technologies across multiple urban water-cycle flows and domains; (b) the capability of said methods to account for deep uncertainties, either socioeconomic or climate-driven; (c) the flexibility of these methods to capture, study, and integrate emerging and novel technologies in a dynamic, ever-changing context that expands to sustainability contexts such as the circular economy (Geissdoerfer *et al.* 2017) and nature-based solutions (Seddon *et al.* 2020).

2. Materials and methods

1. Urban water cycle simulation and the precursor model of UWOT

Among the various integrated assessment tools and methods, the urban water cycle model simulation (Sokolowski and Banks 2009) stands out as a promising, physically consistent and conceptually explainable modelling technique to mimic the response of whole urban water systems ex ante. Following advances in the logic and depth of integration in water systems, urban water simulation models have started from sectoral applications focusing on specific parts of the water cycle, expanding in more recent years to account for integrated applications targeting multiple water cycle streams and even water-food-energy nexus applications (Peña-Guzmán *et al.* 2017, Zhang *et al.* 2020, Evans *et al.* 2023). At a generic level, these simulation models provide insights that assist water system management and planning by reproducing source–demand interactions over time and space, as well as allowing the modeller to set planning rule modifications and simulate their impact (Sulis and Sechi 2013). Simulation models also form building blocks for more elaborate studies, such as analyses of systemic (deep) uncertainties or of optimal planning policies, by being embedded within so-called Parameterisation-Simulation-Optimisation (PSO) frameworks (Koutsoyiannis and Economou 2003), which connect a simulation model with optimisation algorithms and (typically parsimonious) parameter sets to reduce the dimensionality of optimal search in planning strategies.

For integrated applications in the urban water cycle, an interesting family of simulation techniques for this study is the so-called *metabolism* modelling type (Behzadian and Kapelan 2015), which relies on metabolic relations between units and mass-balance equations of (primarily) urban water streams, as well as contaminants, materials, energy flows, and other urban-system quantities, to quantify the complex interactions of Urban Water Systems (UWS). Multiple models of the metabolism type have been developed in recent decades for a range of UWS (Mitchell *et al.* 2001; Behzadian and Kapelan 2015; Renouf *et al.* 2018), primarily relying on water mass balance to study the combined effect of multiple urban water interventions. A precursor of this modelling concept is the Urban Water Optioneering Tool (UWOT) (Makropoulos *et al.* 2008), an urban water-cycle model that has been the modelling testbed of several previous studies and projects for the Urban Water Management and Hydroinformatics (UWMH) group of the National Technical University of Athens (NTUA) over the course of more than 18 years. UWOT employs the so-called "signal model language" (Makropoulos *et al.* 2008) to model urban water streams (DW, WW, SW) and include different interventions that target all or any of them. The initial development of UWOT started from household and neighbourhood-level studies for specific decentralised technologies such as RWH and GWR (Rozos *et al.* 2010), and then, over the course of more than a decade, expanded to more complex cases including neighbourhood-scale blue-green area design (Rozos *et al.* 2013), whole city cycle modelling (Nikolopoulos *et al.* 2019, Bouziotas *et al.* 2023) and analyses of distributed neighbourhood options under scenarios of urban growth (Bouziotas *et al.* 2015). UWOT is able to include individual water uses and technologies/options for managing them, starting from the household level and aggregating to coarser spatial units, also performing a (loose) matching with supply resources at the district-metered area level (Rozos and Makropoulos 2013).

Despite the persistence of the precursor UWOT model across multiple projects and applications of varying complexity, the tool has multiple limitations that stem from its segmented development over almost two decades. It relies on a complex architecture that includes major parts in legacy code: its solver engine is built in C, with extensive wrappers in old versions of Python (2.X), a graphical user interface built using PyQt, and a limited API focused on static methods mainly wrapped on PyQt elements rather than directly on the C solver. More importantly, the compiled engine is procedural and not built using OOP, despite the affinity of the "signal language" and relevant components with classes and objects. As a result, the UWOT solver relies on a legacy codebase that exceeds 15,000 lines of code and over 100 files with multiple dependencies, leading to poor maintainability and high technical debt. Another important design omission for the old model is the lack of stepwise solving; UWOT performs the solve step of an individual simulation scenario using large matrices in C. While this is computationally efficient (especially for older hardware systems), it does not allow any low-level flexibility coming from, for instance, the introduction of dynamic feedback mechanisms to (conditionally) morph and update individual components or change component parameters in the middle of a simulation. Despite these drawbacks, the use of UWOT offers a solid testbed for inter-model comparisons, in order to validate the outcomes of the solver presented in this study, as well as compare runtimes and performance. Moreover, the new model is built to offer a high percentage of backward compatibility with UWOT, in order to enable a smoother transition for the modeller user base of UWMH that relies on the old model's GUI and included interventions.

2. Graph-based and object-oriented simulation: the OPTIONAUT model

With the three research questions discussed in Section 1 in mind, this study presents a novel modelling approach based on graph theory and object-oriented programming (OOP) to simulate urban water flows and their interaction through various objects (components) that generically represent collection, treatment, distribution, or water-demand generation within the UWS. By employing demand-based, signal-driven logic that connects these objects in a directed graph, from household demand to distributed and then centralised supply, this model permits the study of interconnected water-processing units as a whole represented through a network structure. The model is named OPTIONAUT, i.e. OPTIONing Associative Urban water Testbed. The main idea behind the OPTIONAUT object ontology is to establish (and simulate based on) "associative" relationships between different demand and supply components. Initially, each demand or supply component is represented as a nodal object within a directed graph (DiGraph) network (Figure 1). These components act as generic generators or transformers of water signals belonging to different urban water-cycle streams (DW, WW, SW), with each containing the distinct information characterising that stream, i.e., the requested or generated water quantity Q_n in L/dt, as well as the water quality of an abstract contaminant Q_l in mg/L. It then becomes apparent that any generated or requested signal, regardless of the water-cycle stream it belongs to, is fully defined through the associative array $\{Q_{n,}\ Q_l\}$. For initial model development, we further distinguish between three major categories of urban water-cycle streams (even though any new categories can be added as long as $\{Q_{n,}\ Q_l\}$ is defined): (a) drinking-water (DW) request signals, which, for most contaminants, have strict quality requirements (so that $Q_l \rightarrow 0$), (b) wastewater (WW) generation signals, (c) stormwater/runoff signals (SW). The demand and supply components of Figure 1 can then be defined to produce DW, WW, and SW signals and output them (through the associative array {out}) or expect them as inputs (through respective associative arrays, {in1}, {in2}, etc.).

Network edges are then used as "associations" to represent the (instantaneous within the same time step dt) flow of "water signals" from one component to the other(s). Water quantity Q_n (in L/dt) and (conservative) water quality Q_l (in mg/L) are transmitted through each signal and delivered to components according to the specified graph topology, using the assumption that any component can alter one, the other, or both of these attributes within a simulation time step, dt. The associative nature of the engine is extended in the information flow as well; by using associative arrays as the preferred data structure, the "water signal" between two connected components represents arbitrary information that is valuable to urban water cycle simulation. Parameterisation of each component follows a similar approach, requiring an associative array of parameters {params} for each component that can be flexibly altered at any time step (along with their units in a separate associative array, {units}). Both scalar and time-series parameter quantities are accepted, with the latter required to conform to the time steps of the simulation. Finally, some parameters have an internal associative array that stores their simulation states {states}, as it is important to keep a memory of previous states for some transformation mechanics. Examples in the context of urban water include runoff units, which require antecedent soil-moisture states, and storage units, which record and recall their stored water volumes $\{V_0, V_1, V_2, \ldots, V_{t-1}, V_t\}$ to be able to calculate outputs and future stored water volumes according to their water-balance equation, which belongs in the simplified form of:

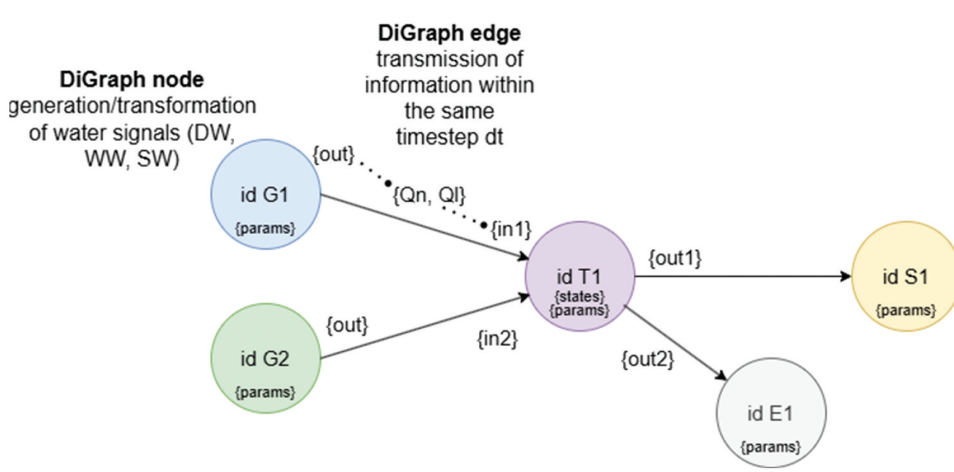


Figure 1. Conceptual schema of the nodes and edges of the associative solver.

$$out1 = \min(V_{\max} - \max(0, V_{t-1} + in1 - EV_t - G_t), 0) \quad (1)$$

where out_1, in_1 are the respective outflow and inflow signals, V_{t-1} is the previous state of storage, and $\{EV_t, G_t, V_{max}\}$ is the parameters associative array that includes two timeseries (evapotranspiration and losses to groundwater) and one scalar (the maximum design volume V_{max}).

Naturally, the use of nodal objects as components suggests the use of principles of object-oriented programming (OOP) to simplify coding patterns and ensure high maintainability. The use of OOP rules to design all components allows an inheritance structure to be followed (Figure 2), where each node inherits properties from a parent main Node class (*Comp* superclass) that provides the standards needed for simulation. The most important standard needed across all components is the solve method, which follows the pattern:

$$\{\{outs\}\} = Comp.solve(dt, \{\{ins\}\}, \{params\}) \quad (2)$$

and allows for a stepwise solving of the component at timestep dt, i.e. the processing of incoming information association {{ins}}={in1, in2, . . .} with the use of a specific parameterisation {params}, to create the outgoing information signals {{outs}}={out1, out2, . . .}. Other standards defined by the *Comp* main node class relate to methods to alter and update parameters and their units, as well as checks about whether the parameter data types are correct prior to solving. As long as these standards, such as the stepwise solve method, are provided for each component class, any component that processes (water) signals can be designed. In the initial development of OPTIONAUT, we have aimed for consistency and reverse compatibility with the precursor model and have thus included subclasses for: (a) household-type generators that generate DW requests and WW signals, (b) runoff (SW) generators, (c) water treatment units (DW/SW/WW), (d) water storage units, (e) generic signal modulators, which alter the water signal or split it according to logical conditions, (f) conveyance units, a subclass of modulators that consider DW network losses to the water signal, (g) definitions for end or inline nodes that primarily perform logging or summation.

To keep the stack modern, OPTIONAUT is designed and tested in Python 3.x with a minimal stack using *datetime*, *numpy*, *pandas*, and *networkx* to handle graph properties, associations and solve actions, as well as *matplotlib/seaborn* to visualise outcomes. The high-level python graph library of *networkx* (Hagberg and Conway 2020) is employed to design, manage and manipulate the resulting UWS graph, from tap to source. Moreover, the native "batteries-included" Python OOP features are used to set up and manage the class-inheritance structure without additional dependencies. Besides the *solver* module, and to ensure compatibility with its precursor, OPTIONAUT includes the following separate modules that work in sequence: (a) the *parser* module, which utilises the same PyQt GUI as UWOT to read in user-defined

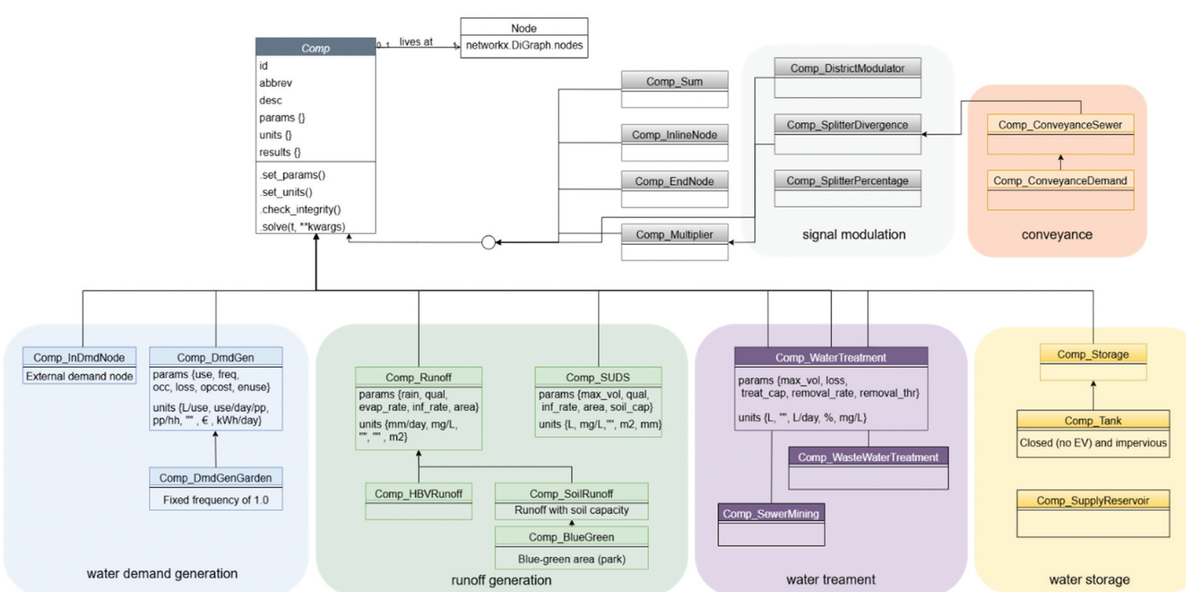


Figure 2. Class hierarchy and inheritance structure for OPTIONAUT.

topologies and component attributes; (b) the *graph_designer* module, which is used to design the OOP graph network from scratch (or convert it from the parsed structure of (a)); (c) the *utils* module, which includes helper functions to prepare hydrological component parameters such as evapotranspiration (EV).

3. Results

The new simulation testbed is tested against a real-world UWS, using different scenarios of circular water technology implementation and uptake. More specifically, the benchmark of a newly developed circular water neighbourhood in Limburg, the Netherlands, is used, where an array of circular water interventions is employed, including (a) different household types (houses and apartments) with water-saving appliances, (b) a neighbourhood-scale RWH scheme, combined with a 'water square' central infiltration basin for excess rainwater management, (c) a neighbourhood-scale GWR scheme that purifies selected water uses such as showers and handbasins. The specifics of this case are described in detail in a previous study (Bouziotas *et al.* 2019) that distinguished among three different design scenarios (BAU, SCENA, SCENB) focusing on different aspects of integration between the aforementioned circular water interventions (see Table 1 for more details). Aspects of the integrated UWS have been extensively studied using the precursor model (Bouziotas *et al.* 2019); in the context of this study, it is useful to compare previous results with the ones being generated by the new solver, both in terms of quantities but also in terms of topological characteristics, setup and runtime.

To test the case study, a total of 24 circular water components are developed, tested, and simulated using OPTIONAUT, belonging to the inheritance structure of Figure 2. Three network topologies are designed using these components, i.e. one for each design scenario (Figure 3). Compared to the 49 different components found in its precursor, we observe a significant reduction in modelling units by 51%, primarily because different UWOT units had thematic overlap and performed similar signal transformation functions, which have now been grouped into the same component-subclass. Moreover, a number of UWOT units addressed aspects of supply or water-energy concepts in a limited fashion and will be conceptually revisited before being added as an OPTIONAUT class; none of these affected the case study or limited the output of the new solver. With regard to code size and complexity, the difference is even more significant; OPTIONAUT, in total, features 2715 lines of code divided across its four module Python files, effectively reducing the code footprint by 81.2% and required repo files by 89.4%. As it is Python-based, the resulting code structure is readable, lean and significantly more maintainable than the previous legacy codebase.

With regards to modelled quantities, we observe full agreement between modelled information outputs between the two models, accounting for rounding errors that were, on average, <0.1%. The only exception that has deviations in modelled quantities across models in the range of 5%–10% was the blue-green component (*Comp_BlueGreen*), which is attributed to a revision in the way evapotranspiration (EV) rates are calculated per time step; the precursor method relied on the Hargreaves equation (Rozos *et al.* 2013) and the estimation of solar/astronomical quantities using hard-coded latitudes typical of the Mediterranean basin. These assumptions limited the applicability and validity of the component results. In OPTIONAUT, EV can now be estimated using three different methods (Hargreaves, Tegos, and Penman-Monteith), depending on available hydrometeorological data (Howell and Evett 2004, Tegos *et al.* 2015), and across a large range of latitudes to provide near-global coverage. With regards to simulation runtime, we observe generally slower runtimes that range in the order of 8–35 s for simulation of 10–30 years at a daily time step (Table 2), while the precursor remains fast at under 10 *s*. On average, OPTIONAUT is 5.3–6.5 times slower than its precursor. This is expected compared to the procedural C engine, as the

Table 1. Overview of considered scenarios.

Abbreviation	Short Description	Description
BAU	BAU	Business As Usual (BAU) case. Linear and centralised water management for SW and WW, no decentralised technologies.
SCENA	RWH|GWR	Separate (distinct) RWH and GWR systems, targeting different water uses. Water-saving appliances.
SCENB	RWH+GWR	Combined RWH and GWR systems using the same storage unit, targeting most water uses. Water-saving appliances.

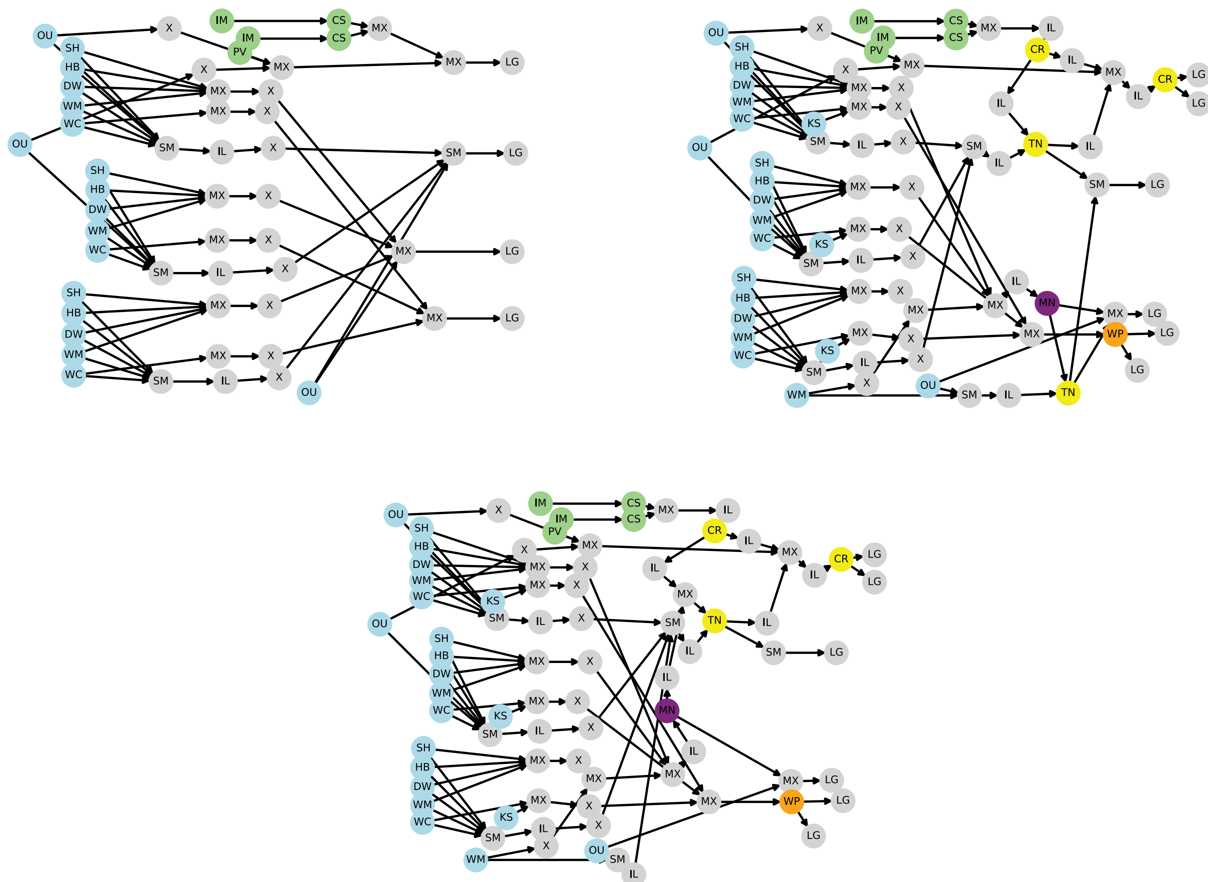


Figure 3. Topologies for the real-world case study for the design scenarios of BAU (upper left), SCENA (upper right) and SCENB (lower panel).

Table 2. Runtimes of OPTIONAUT and the precursor model for all scenarios at different simulation lengths.

		Length of simulation			
		10 yrs	30 yrs	50 yrs	100 yrs
scenario	Amount of daily timesteps dt	4018	11323	18628	36890
BAU	Runtime [s] - UWOT	1.31	3.55	7.28	12.76
SCENA	Runtime [s] - UWOT	2.20	5.91	9.65	20.49
SCENB	Runtime [s] - UWOT	2.08	5.85	9.44	18.98
BAU	Runtime [s] - OPTIONAUT	8.14	24.1	39.1	77.66
SCENA	Runtime [s] - OPTIONAUT	13.45	34.95	58.4	116.6
SCENB	Runtime [s] - OPTIONAUT	13.48	36.28	58.75	121.97

solver is now purely Python-based and attempts low-level solving at every component and every timestep, with previous authors reporting similar effects for native Python solvers (Steffelbauer and Fuchs-Hanusch 2015). While markedly slower than the C-based model, the runtime is not inhibitive and allows typical cases of up to 50 years at a daily time step to be simulated in under a minute, while simulation-optimisation applications will be completed in a few to several hours. Interestingly, both models scale well with increased data length; assuming a scaling factor *LF* defined as the ratio of the normalised increase of runtime r to the normalised increase of timesteps t and taking the simulation of 10 years as the basis of reference, we observe that *LF* is in the range of 0.93–1.25 for the precursor model and 0.88–1.08 for OPTIONAUT, meaning that both models exhibit linear complexity $O(n)$ when facing bigger datasets. While this is expected for a matrix-based simulator such as UWOT, it was interesting to note for OPTIONAUT, as it reveals that the use of associative data arrays is a solid design choice, offering general space complexity of $O(n)$ and $O(1)$ complexity for insertion, deletion and lookup of values (Mehlhorn and Sanders 2008). We did not observe system freeze, memory issues, divergence or instabilities during simulation runs for any of the considered topologies.

4. Discussion

Revisiting the three research questions of the Introduction section, we argue that OPTIONAUT demonstrates a modelling architecture where: (a) the effect of multiple technologies across multiple urban water cycle flows and domains can be modelled through relevant water signal associations $\{Q_n, Q_l\}$, (b) any component parameter can be changed prior to or even during simulation, allowing for the exploration of (deep) uncertainties in the UWS, (c) the proposed architecture, by design, provides the elasticity to include, with low effort, any component useful in UWS sustainability contexts. The authors believe that, since these three prerequisites are covered, OPTIONAUT has the intrinsic value to be the testbed for generic, arbitrary circular water solutions. High code readability and maintainability further enhance the generic nature and reusability of the new solver.

The only observed downside, compared to its precursor, is a reduction in runtime speed, which is, however, not inhibitive, with most practical simulation cases at the climate-level timescales of 30 or 50 years finishing in a matter of seconds and under a minute. If speed optimisation becomes essential (for instance, in stochastic optimisation use cases), runtimes can be reduced with a smarter design of components to simplify topologies, the use of additional source-code translator libraries like Cython, or by exploring multiprocessing versions. This is aided by the fact that typical network topologies are dendritic (Figure 3) and include multiple components belonging to the same tree depth that can be simulated in parallel. Another practical limitation of the current version of OPTIONAUT is its lack of a graphical interface. To counteract this, the new solver is backwards compatible with most of the precursor functionalities so modellers can use the UWOT GUI to design graphs easily, before tweaking them with scripts (if needed) prior to running OPTIONAUT. However, the design of novel components that surpass UWOT will partly break this functionality, so the authors envision a web-based graphical interface to facilitate the generation of network topologies for UWS, along with the provision of OPTIONAUT capabilities to users through a web-browsing environment using the Software-as-a-Service (SaaS) model. Finally, the architecture has been demonstrated for demand-based UWS focusing on water signals. While this limits applications to generic urban water, the associative nature of the solver and its arbitrary graph-based information flows can be readily expanded to model abstract conservative or non-conservative quantities that are transported across nodes, including, for instance, energy signals for water-energy applications. This will increase modelling complexity and might require multiple inter-connected graphs to distinguish flow-information layers, for instance one for water and one for energy.

5. Conclusions

In this study, a graph-based urban water cycle simulation engine is developed, able to model information flows through graph edges, and circular water technologies, either at the supply or the demand side, through graph nodes. By employing object-oriented principles and a modern stack, the resulting framework offers high code readability and maintainability while providing the abstraction needed to model *any* information (including, for instance, energy or financial quantities) and *any* technology. The concept is demonstrated for a real-world neighbourhood design in the Netherlands, where it delivers quantitative insights that agree with previous findings while significantly reducing the complexity of the precursor model. With a drastically reduced code footprint (>80%) relying on OOP inheritance, the tool is expected to form a rigorous testbed for circular water policies that rely on emerging technologies (Frijns *et al.* 2024). Future developments will focus on further extending the library of novel circular water technologies, alleviating speed losses compared to C-based water cycle models, and introducing complex use cases with key flows such as energy consumption/generation, nutrients, or multiple urban water cycle streams across large spatial domains.

Author contributions

CRediT: **Dimitrios Bouziotas:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Software, Supervision, Validation, Visualization, Writing – original draft; **Andreas Efstratiadis:** Project administration, Writing – review & editing; **Christos Makropoulos:** Funding acquisition, Project administration, Writing – review & editing.

Disclosure statement

No potential conflict of interest was reported by the author(s).

ORCID

Dimitrios Bouziotas http://orcid.org/0000-0003-2172-350X
Andreas Efstratiadis http://orcid.org/0000-0002-2907-1633

References

Allen, L., Christian-Smith, J., and Palaniappan, M., 2010. Overview of greywater reuse: the potential of greywater systems to aid sustainable water management. *Pacific Institute*, 654 (1), 19–21.

Almaaitah, T., *et al.*, 2021. The potential of blue-green infrastructure as a climate change adaptation strategy: a systematic literature review. *Blue-green systems*, 3 (1), 223–248. doi:10.2166/bgs.2021.016.

Bacchin, T.K., *et al.*, 2014. Green-blue multifunctional infrastructure: an urban landscape system design new approach. In: 13th International Conference on Urban Drainage, Sep. Vol. 4. Sarawak, Malaysia, 1–8.

Bazrkar, M.H., Adamowski, J.F., and Eslamian, S., 2016. Water system modelling. *In*: J.N. Furze, K. Swing, A.K. Gupta, R.H. McClatchey, and D.M. Reynolds, eds. *Mathematical advances towards sustainable environmental systems*. Cham: Springer International Publishing, 61–88.

Behzadian, K. and Kapelan, Z., 2015. Modelling metabolism based performance of an urban water system using WaterMet2. *Resources, conservation & recycling*, 99, 84–99. doi:10.1016/j.resconrec.2015.03.015.

Bouziotas, D., *et al.*, 2019. Towards circular water neighborhoods: simulation-based decision support for integrated decentralized urban water systems. *Water*, 11 (6), 1227. doi:10.3390/w11061227.

Bouziotas, D., *et al.*, 2023. Assessing the resilience of circularity in water management: a modeling framework to redesign and stress-test regional systems under uncertainty. *Urban water Journal*, 20 (5), 532–549. doi:10.1080/1573062X.2023.2190030.

Bouziotas, D., Rozos, E., and Makropoulos, C., 2015. Water and the city: exploring links between urban growth and water demand management. *Journal of hydroinformatics*, 17 (2), 176–192. doi:10.2166/hydro.2014.053.

Evans, B., *et al.*, 2023. A modelling testbed to demonstrate the circular economy of water. *Journal of Cleaner production*, 405, 137018. doi:10.1016/j.jclepro.2023.137018.

Ferrans, P., *et al.*, 2022. Sustainable urban drainage system (SUDS) modeling supporting decision-making: a systematic quantitative review. *Science of the total environment*, 806, 150447. doi:10.1016/j.scitotenv.2021.150447.

Frijns, J., Smith, H.M., and Makropoulos, C., 2024. Enabling the uptake of circular water solutions. *Water policy*, 26 (1), 94–110. doi:10.2166/wp.2024.167.

Geissdoerfer, M., *et al.*, 2017. The circular economy–A new sustainability paradigm? *Journal of Cleaner production*, 143, 757–768. doi:10.1016/j.jclepro.2016.12.048.

Hagberg, A. and Conway, D., 2020. Networkx: network analysis with python. URL. 1031. Available from: https://networkx.github.io.

Howell, T.A. and Evett, S.R., 2004. *The penman-monteith method*. Washington, DC: USDA-Agricultural Research Service, Conservation & Production Research Laboratory, 14.

Koutsoyiannis, D. and Economou, A., 2003. Evaluation of the parameterization-simulation-optimization approach for the control of reservoir systems. *Water Resources research*, 39 (6). doi:10.1029/2003WR002148.

Lai, E., Lundie, S., and Ashbolt, N.J., 2008. Review of multi-criteria decision aid for integrated sustainability assessment of urban water systems. *Urban water Journal*, 5 (4), 315–327. doi:10.1080/15730620802041038.

Makropoulos, C., *et al.*, 2018. Sewer-mining: a water reuse option supporting circular economy, public service provision and entrepreneurship. *Journal of environmental management*, 216, 285–298. doi:10.1016/j.jenvman.2017.07.026.

Makropoulos, C.K., *et al.*, 2008. Decision support for sustainable option selection in integrated urban water management. *Environmental modelling & Software*, 23 (12), 1448–1460. doi:10.1016/j.envsoft.2008.04.010.

Mehlhorn, K. and Sanders, P., 2008. *Algorithms and data structures: the basic toolbox*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Mitchell, V.G., Mein, R.G., and McMahon, T.A., 2001. Modelling the urban water cycle. *Environmental modelling & Software*, 16 (7), 615–629. doi:10.1016/S1364-8152(01)00029-9.
Nikolopoulos, D., *et al.*, 2019. Tackling the “new normal”: a resilience assessment method applied to real-world urban water systems. *Water*, 11 (2), 330. doi:10.3390/w11020330.
Peña-Guzmán, C.A., *et al.*, 2017. Urban water cycle simulation/management models: a review. *Water*, 9 (4), 285. doi:10.3390/w9040285.
Pidou, M., *et al.*, 2007. Greywater recycling: treatment options and applications. *In:* Proceedings of the Institution of Civil Engineers-Engineering Sustainability, Sep. Vol. 160. Thomas Telford Ltd, 119–131. https://www.emerald.com/jensu/article-abstract/160/3/119/421378/Greywater- recycling-treatment-options-and .
Renouf, M.A., *et al.*, 2018. Understanding urban water performance at the city-region scale using an urban water metabolism evaluation framework. *Water research*, 137, 395–406. doi:10.1016/j.watres.2018.01.070.
Rozos, E. and Makropoulos, C., 2013. Source to tap urban water cycle modelling. *Environmental modelling & Software*, 41, 139–150. doi:10.1016/j.envsoft.2012.11.015.
Rozos, E., Makropoulos, C., and Butler, D., 2010. Design robustness of local water-recycling schemes. *Journal of water Resources planning and management*, 136 (5), 531–538. doi:10.1061/(ASCE)WR.1943-5452.0000067.
Rozos, E., Makropoulos, C., and Maksimović, Č., 2013. Rethinking urban areas: an example of an integrated blue-green approach. *Water science & technology: Water supply*, 13 (6), 1534–1542. doi:10.2166/ws.2013.140.
Seddon, N., *et al.*, 2020(1794). Understanding the value and limits of nature-based solutions to climate change and other global challenges. *Philosophical transactions of the royal society B: Biological sciences*, 375 (1794), 20190120. doi:10.1098/rstb.2019.0120.
Short, M.D., *et al.*, 2012. Managing adaptation of urban water systems in a changing climate. *Water resources management*, 26 (7), 1953–1981. doi:10.1007/s11269-012-0002-8.
Sokolowski, J.A. and Banks, C.M., 2009. *Principles of modeling and simulation*. Hoboken, NJ, USA: John Wiley & Sons, Inc.; ISBN 9780470403563.
Steffelbauer, D. and Fuchs-Hanusch, D., 2015. OOPNET: an object-oriented EPANET in Python. *Procedia engineering*, 119, 710–718. doi:10.1016/j.proeng.2015.08.924.
Sulis, A. and Sechi, G.M., 2013. Comparison of generic simulation models for water resource systems. *Environmental modelling & Software*, 40, 214–225. doi:10.1016/j.envsoft.2012.09.012.
Tegos, A., Malamos, N., and Koutsoyiannis, D., 2015. A parsimonious regional parametric evapotranspiration model based on a simplification of the Penman–Monteith formula. *Journal of hydrology*, 524, 708–717. doi:10.1016/j.jhydrol.2015.03.024.
Vuppaladadiyam, A.K., *et al.*, 2019. A review on greywater reuse: quality, risks, barriers and global scenarios. *Reviews in environmental science and Bio/technology*, 18 (1), 77–99. doi:10.1007/s11157-018-9487-9.
Zhang, T., *et al.*, 2020. Synergy assessment and optimization for water-energy-food nexus: modeling and application. *Renewable and sustainable energy reviews*, 134, 110059. doi:10.1016/j.rser.2020.110059.