SOFTWARE METAPAPER

A Software Framework for the Integration of Infrastructure Simulation Models

Will Usher[1] and Tom Russell[2]

[1] Infrastructure System Modeller, University of Oxford, GB
[2] Research Software Engineer, University of Oxford, GB
Corresponding author: Will Usher (william.usher@ouce.ox.ac.uk)

Infrastructure systems, such as those that generate and transmit energy, process waste water and enable the transportation of people and goods, provide fundamental services that underpin modern society. The simulation model integration framework (smif) is an open source Python package which supports the coupling and running of infrastructure simulation models as a system-of-systems. smif connects individual simulation models – written in different programming languages and using different methods of data input and output – allowing users to compose a system-of-systems with explicit configuration of the connections and data transformations between models. Users can then run system-of-systems models using various data sources in order to explore the effects of uncertainty in external factors and model parameterisations, and to test alternative approaches to long-term decision-making and planning across the connected models.

(1) Overview

Introduction

Infrastructure systems, such as energy supply, water supply, transport and digital communications provide crucial services to society and underpin economic growth and sustainable development [18]. National infrastructure systems are complex adaptive systems that exhibit key properties including diversity and adaptivity, dynamics, irreversibility and emergence across three hierarchical levels of agents, networks and systems [13]. Consequently, the use of simulation models is well suited to examine these systems, their linkages and their behaviour over the long-term [11]. In contrast to other approaches that have embedded an infrastructure system-of-systems approach into a common mathematical framework [4], the emphasis within the MISTRAL research programme, under which the simulation modelling integration framework (smif) was developed, is to couple new and existing infrastructure simulation models of the United Kingdom to support strategic long-term planning of national infrastructure.

The simulation model integration framework (smif) is a software tool that supports the configuration, coupling, running and orchestration of infrastructure simulation models as a system-of-systems. smif has been developed to support the coupling of detailed high-resolution engineering models of infrastructure systems at the national scale to explore infrastructure performance under alternative planning strategies and future scenarios. Models of transport, energy, solid waste, water supply and digital communications have been coupled using smif within the broader framing of the Infrastructure Transitions Research Consortium's Assessment Process to build the second-generation National Infrastructure Systems Model (NISMOD 2). This builds on the work from the first ITRC programme which is described in [6].

Infrastructure simulation models, such as the Combined Gas and Electricity Network model – CGEN++ [1], the high-resolution energy demand model HIRE [2], the road transport model [9, 10] and the Cambridge Communications Assessment Model [12, 14] have been developed within the MISTRAL programme and used to assess the performance of their respective infrastructure systems over long time scales. The availability of infrastructure data differs across the sectors, and the spatial and temporal resolution at which the models are resolved is a function of both data availability and the properties of the system under examination. For example, the electricity supply system must meet demand continuously and is modelled at an hourly resolution with a representation of the United Kingdom's transmission network as 29 busbars. In contrast, the digital communications model is concerned with the availability of bandwidth in a particular

year and yet considers network upgrades at the building level. The road transport model simulates road congestion and trip time by assigning trip demand over the national road network for each hour in one representative working day. The mismatch between time and space is illustrated in **Figure 1** and presents a set of distinct requirements that are not available in existing software solutions, such as the need to perform frequent spatial and temporal data transformations [19] within a model run.

Given the long timescales, large capital expenditure and path-dependency associated with developing infrastructure systems, each of the models is simulated over annual timesteps, allowing an exploration of the effect of different planning strategies upon the performance of the systems within each timestep. Combining the high spatial and temporal resolution simulation modelling with the coarser planning timesteps provides insights into the interconnected performance of these infrastructure systems into an uncertain future. To characterise future uncertainty, high resolution scenarios of the evolution of national population and demographics [15, 16], economic productivity, building stock [3] and environment [5] are propagated through the infrastructure simulation models.

The architecture of smif provides a consistent interface to each of the models by wrapping them in a Python class. While this requires some effort to configure and write a wrapper, the wrapped model is then an interoperable unit which can be easily included in one or more system-of-systems. Similarly, smif enables the development of Adaptors which perform additional data transformations between models which may require data in different units, categories or resolutions.

There are existing tools that provide some of the functionality required to simulate an infrastructure system-of-systems. These have provided inspiration during the development of smif, but also demonstrate the value that smif adds for the infrastructure systems community. For example, the OpenMI standard [20] provides the specifications for an interoperable modelling interface, is very technical and closely linked to the Java language and water modelling community. The Basic Model Interface [17] defines a set of functions to query, modify and run models using the Python Modelling Toolkit [7] written for the surface dynamics community. cylc [8] is a general workflow tool which manages the orchestration of dependent tasks. The focus of smif is to provide a layer above the orchestration of model simulation, so that infrastructure modellers can engage with the system-of-systems concepts that are relevant to them with smif handling the technical detail. For example, we observe that a system-of-systems model run unpacks into a directed-acyclic graph of processes as a function of the dependencies between models. This graph can subsequently be passed to an existing workflow tool or scheduler for orchestration. smif does provide a serial scheduler which will run models one at a time in an order that satisfies their dependencies.

smif is an open-source Python package, available to install through PyPI or conda-forge. Source code is available at https://github.com/nismod/smif/ and full documentation is at https://smif.readthedocs.io/en/latest/. As of publication, smif is in active use as the "glue" to support the integrated modelling work of a consortium of around 20 infrastructure systems researchers.

Implementation and architecture
smif has three main elements, as shown in **Figure 2**: a user interface to configure model runs; data storage for configuration and data exchange; and orchestration and glue code to run coupled models. The core of smif, including a command line interface, is written in Python 3, with a graphical user interface written in JavaScript using React.
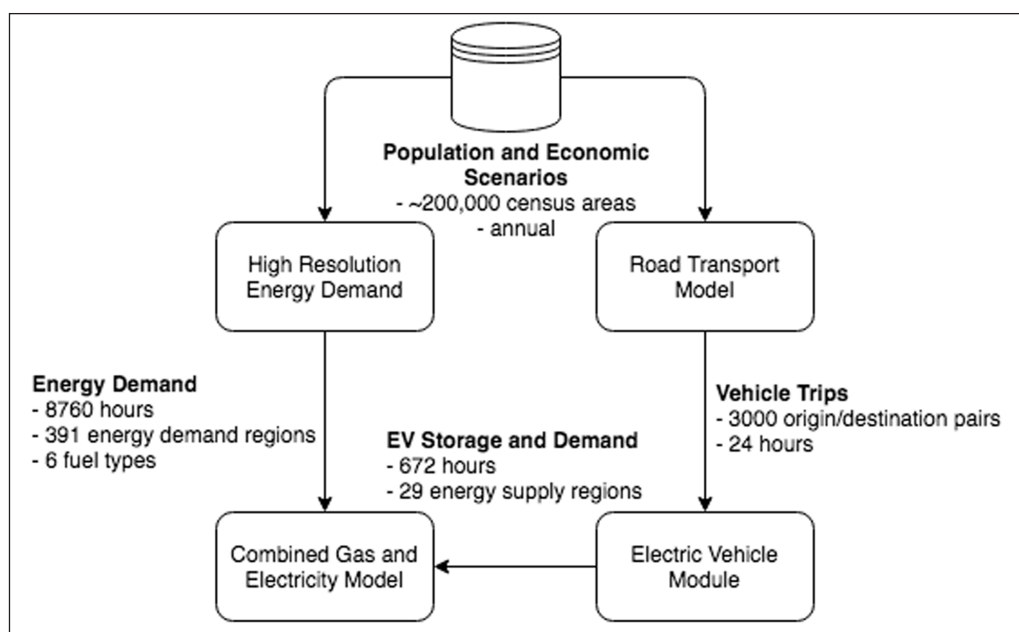
Figure 1: A sketch of the diverse data dimensions produced and required by energy and transport models – a subset of the infrastructure simulation models included in NISMOD 2.
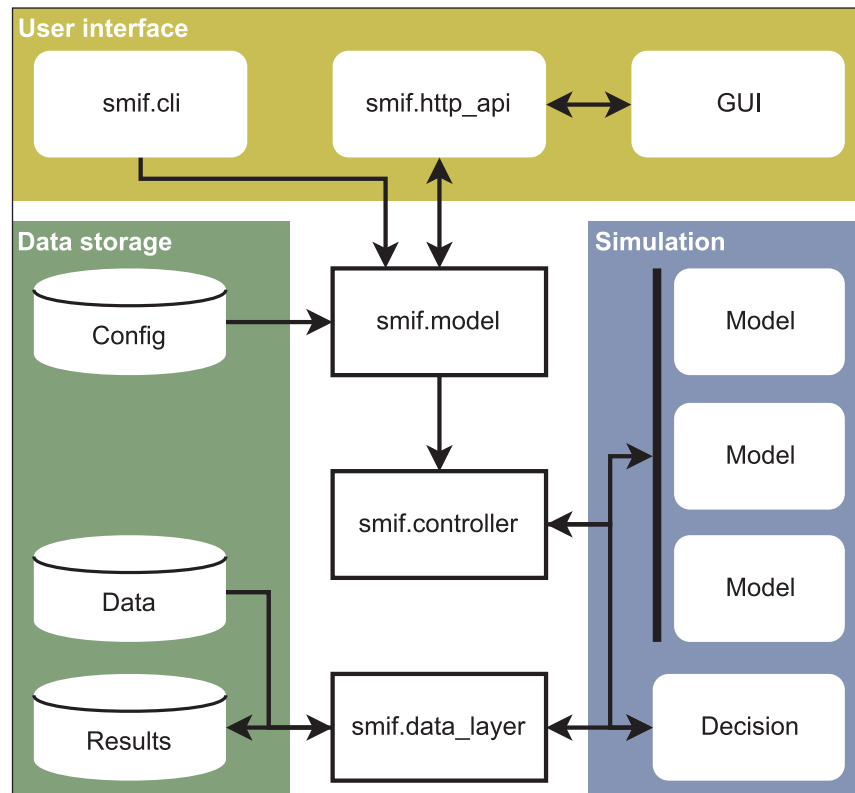
Figure 2: Shows a simplified representation of the smif architecture.

Data and configuration are stored in text files as YAML and CSV, with a data layer abstraction included in preparation for future work on database and binary file stores.

The classes in `smif.model` are concerned with configuring a system-of-systems. Several SectorModels compose a SosModel, and a fixed parameterisation of a SosModel comprises a repeatable ModelRun.

smif uses a plugin architecture to include simulation models. To add a new model to the smif model library, developer users write a Python wrapper class implementing SectorModel. These plugin wrappers are concerned with the mechanics of running each simulation model: passing in data and parameters; running the program or module; and parsing outputs to be passed back to smif.

The `smif.controller` module is concerned with running models: a ModelRunner alternates between requesting decisions from a decision module and requesting simulations from a scheduler, which resolves the model run order based on the data dependencies between models and calls out to run the models. Future extensions to smif could add extra schedulers or workflow libraries, including cylc as mentioned previously.

Quality control
Unit tests have been written for each Python module and the user interface. Continuous integration is set up to test both Python and JavaScript on Travis. Coverage statistics are recorded at codecov.io. Test coverage is over 90% for the Python code and around 70% for the user interface.

An example project is available to demonstrate features of smif. It can be generated on the command line in the current folder using the command `smif setup`. The command `smif --help` shows available commands; `smif list` shows available model runs and `smif run energy_central` runs a model run, with results saved to the results directory. `smif app` runs the user interface server, which is then accessible in a browser on http://localhost:5000/.

(2) Availability
Operating system
Cross-platform 64-bit (currently tested on Windows Server 2012 R2 and Ubuntu Trusty 14.04, in use on Windows/Mac/Linux).

Programming language
Python 3

Additional system requirements
None

Dependencies
fiona>=1.7
Flask>=0.12
isodate>=0.6
networkx>=1.11
numpy>=1.11
Pandas>=0.23
Pint>=0.8
psycopg2>=2.7
pyarrow>=0.9
python-dateutil>=2.6
pywin32; sys_platform == 'win32'
Rtree>=0.7
ruamel.yaml==0.15.50

shapely>=1.3
Xarray>=0.10

List of contributors
Will Usher, University of Oxford
Tom Russell, University of Oxford
Roald Schoenmacher, University of Oxford
Craig Robson, University of Newcastle

(3) Reuse potential

The core focus of smif is the infrastructure system-of-systems community. The most obvious opportunities for reuse reside within this community and include incremental extensions to the existing uses. For example, adjacent to geography and systems engineering, various disciplines could interact with coupled infrastructure systems modelling:

- Health: couple a multi-modal transport simulation model to a local air-quality assessment model.
- Ecosystems biology: couple habitat and species distribution models to climate change scenarios and land use/urban development models.
- Earth sciences: couple a hydrological model of river flows, driven by climate variables, to a water supply model.
- Business and economics: couple firm location or retail site location models with a trip-based transport model in the context of broader changes to demographics and economy.
- Mathematics and computer science: develop algorithms to perform a multi-objective optimisation of a system-of-systems model.
- Political science: represent possible regulatory changes or modes of governance by exploring model parameterisations and developing decision heuristics which implement policies.

However, smif provides general functionality that enables any model that can be run from a Python subprocess to be configured, wrapped and coupled with one or more other configured and wrapped models. In its current form, the software can couple small- to medium-sized simulation models which run over aggregate timesteps, with data exchange occurring after each timestep, where data may consist of multiple variables, each with multiple dimensions. The framework includes features for models whose multi-dimensional input and output data may be linked to vector-based geometries, and where data may need to be transformed as it is transferred between coupled models. If users do not have a need for the modularity and interoperability that smif enables through wrapping models with a consistent interface, the cost of doing so may outweigh the benefit. In this case, it may be better to use a workflow tool.

There are a number of opportunities to extend the framework that would enable a greater number of modelling paradigms to make use of the framework. A full list of these are stored in the repository wiki, but highlights include: generalising the treatment of time to distinguish between aggregate "decision timesteps" and higher resolution "operational timesteps"; and implementing additional backing data stores to enable better I/O performance when running models.

The developers may be contacted through the repository on GitHub. We would welcome pull requests and additional development effort. Ad-hoc support is available through to mid-2020 via the GitHub issue tracker and will revert at the end of the ITRC-MISTRAL project to the open-source and researcher community.

Competing Interests
The authors have no competing interests to declare.

References
1. **Chaudry, M,** et al. 2014 Combined gas and electricity network expansion planning. *Applied Energy*, 113: 1171–1187. DOI: https://doi.org/10.1016/j.apenergy.2013.08.071
2. **Eggimann, S, Hall, J W** and **Eyre, N** 2019 A high-resolution spatio-temporal energy demand simulation to explore the potential of heating demand side management with large-scale heat pump diffusion. *Applied Energy*, 236(November 2018): 997–1010. DOI: https://doi.org/10.1016/j.apenergy.2018.12.052
3. **Ford, A,** et al. 2019 A multi-scale urban integrated assessment framework for climate change studies: A flooding application. *Computers, Environment and Urban Systems*, 75(January): 229–243. DOI: https://doi.org/10.1016/j.compenvurbsys.2019.02.005
4. **Grogan, P T** and **de Weck, O L** 2015 The ISoS Modeling Framework for Infrastructure Systems Simulation. *Systems Journal, IEEE*, 99: 1–12.
5. **Guillod, B P,** et al. 2018 A large set of potential past, present and future hydro-meteorological time series for the UK. *Hydrology and Earth System Sciences*, 22(1): 611–634. DOI: https://doi.org/10.5194/hess-22-611-2018

6. **Hall, J W,** et al. 2016 *The future of national infrastructure: a system-of-systems approach*, 1st ed. Hall, J W, et al. (eds.). Cambridge University Press. DOI: https://doi.org/10.1017/CBO9781107588745

7. **Hutton, E** and **Piper, M** 2019 Python Modelling Toolkit. Available at: https://github.com/csdms/pymt.

8. **Oliver, H J, Shin, M** and **Sanders, O** 2018 Cylc: A Workflow Engine for Cycling Systems. *Journal of Open Source Software*, 3(27): 737. DOI: https://doi.org/10.21105/joss.00737

9. **Lovric, M, Blainey, S** and **Preston, J** 2017 Designing a road traffic model for the cross-sectoral analysis of future national infrastructure. In: *International Symposia for Next Generation Infrastructure.*

10. **Lovrić, M, Blainey, S** and **Preston, J** 2017 A conceptual design for a national transport model with cross-sectoral interdependencies. *Transportation Research Procedia*, 27: 720–727. DOI: https://doi.org/10.1016/j.trpro.2017.12.150

11. **Otto, A,** et al. 2014 A Quantified System-of-Systems Modeling Framework for Robust National Infrastructure Planning. *IEEE Systems Journal*, 10(2): 385–396. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6939636. DOI: https://doi.org/10.1109/JSYST.2014.2361157

12. **Oughton, E,** et al. 2018 Towards 5G: Scenario-based assessment of the future supply and demand for mobile telecommunications infrastructure. *Technological Forecasting and Social Change*, 133(March): 141–155. DOI: https://doi.org/10.1016/j.techfore.2018.03.016

13. **Oughton, E J,** et al. 2018 Infrastructure as a Complex Adaptive System. *Complexity*, 2018: 1–11. Available at: https://www.hindawi.com/journals/complexity/2018/3427826/. DOI: https://doi.org/10.1155/2018/3427826

14. **Oughton, E J** and **Frias, Z** 2018 The cost, coverage and rollout implications of 5G infrastructure in Britain. *Telecommunications Policy*, 42(8): 636–652. DOI: https://doi.org/10.1016/j.telpol.2017.07.009

15. **Smith, A P** 2018 humanleague: a C++ microsynthesis package with R and python interfaces. *Journal of Open Source Software*, 3(25): 629. DOI: https://doi.org/10.21105/joss.00629

16. **Smith, P A** and **Russell, T** 2018 Ukpopulation: Unified National and Subnational Population Estimates and Projections, Including Variants. *Journal of Open Source Software*, 3(28): 803. DOI: https://doi.org/10.21105/joss.00803

17. **Peckham, S D, Hutton, E W H** and **Norris, B** 2013 A component-based approach to integrated modeling in the geosciences: The design of CSDMS. *Computers and Geosciences*, 53: 3–12. DOI: https://doi.org/10.1016/j.cageo.2012.04.002

18. **Thacker, S,** et al. 2019 Infrastructure for sustainable development. *Nature Sustainability*, 2(April). Available at: http://www.nature.com/articles/s41893-019-0256-8.

19. **Usher, W** and **Russell, T** 2017 Challenges in SpaceTime – Coupling models of next generation infrastructure. In: *ISNGI Conference Proceedings 2017*, 411–423. London, United Kingdom. Available at: http://isngi.org/wp-content/uploads/2017/10/ISNGI-Conference-Proceedings-v2.pdf.

20. **Vanecek, S** & **Roger, M** 2014 *OGC® Open Modelling Interface Interface Standard*. Available at: http://www.opengeospatial.org/standards/openmi.