IOP Conference Series: Earth and Environmental Science

# Nested Phoenix: a bottom-up Python model for the life cycle environmental performance of urban built stocks

To cite this article: A Stephan *et al* 2022 *IOP Conf. Ser.: Earth Environ. Sci.* **1122** 012028

# Nested Phoenix: a bottom-up Python model for the life cycle environmental performance of urban built stocks

**A Stephan, G Mulders, K Slavkovic**

Faculty of Architecture, Architectural Engineering and Urban Planning, Université Catholique de Louvain, B-1348, Louvain-la-Neuve, Belgium

Andre.Stephan@uclouvain.be

**Abstract**. Buildings and infrastructure assets in cities represent the dominant majority of the anthropogenic material stock and with the expected population growth this is set to double by 2100. It is therefore critical to quantify the life cycle environmental performance of built stocks, existing and forthcoming, to better manage them, modify their designs and mitigate climate change and resource depletion. Yet existing models fail to provide the required spatial and temporal resolution, are not comprehensive enough and often do not capture shifts in environmental effects. This paper presents Nested Phoenix, a bottom-up Python model that addresses these gaps and provides one of the most sophisticated models for built stocks to date. We present the scope of the model, its functionalities and development solutions before describing the different Python packages used, the overall approach and the database and model architecture. Nested Phoenix enables quantifying material stocks and flows and life cycle embodied, operational and transport environmental flows, alongside carbon sequestration in green infrastructure and biogenic carbon. This is coupled with a dynamic modelling approach that enables the investigation of myriad scenarios over time. This capacity, coupled with spatialization using geographic information systems, represents the breadth of Nested Phoenix.

## 1. Introduction

Cities are and will be critical systems to manage in order to mitigate climate change and resource depletion. In fact, cities alone are responsible for most resource use, greenhouse gas emissions and pollution emissions [1]. Their built stocks represent by far the single most significant accumulation of materials in human society [2]. In addition, cities are complex socio-economic systems with myriad variables driving resource use and their accumulation within them. Understanding the drivers of environmental performance in cities is therefore essential.

However, existing research on urban metabolism, material stocks and flows modelling, and other urban sciences tend to be siloed [3], lack the required consistency in terms of approach and representation and tend to focus on single case studies rather than on re-usable models. With urban data becoming increasingly available, it is easier than ever before to access granular data at a building level in order to model material composition, energy use, water, greenhouse gas emissions and other variables.

For these reasons, it is critical to provide the means to better quantify, represent and visualise resource use, accumulation, and wastage in cities. This will help improve our understanding of urban flows and enable us to improve environmental performance.

The aim of this paper is to present Nested Phoenix, a bottom-up Python-based model to quantify material stocks and flows as well as the life cycle environmental performance of urban built stocks.

This paper focuses solely on the characteristics (including specifications, database implementation and model architecture) of the Python model behind Nested Phoenix and does not include any actual code nor case study application, due to space restrictions. Readers are referred to Stephan, Crawford, Bunster, Warren-Myers and Moosavi [4] to read more about the scientific approach of the model and a pilot case study. Nested Phoenix is under development and more information is available at: http://www.nestedphoenix.com.

Section 2 touches on current approaches in modelling cities for resource efficiency, Section 3 succinctly describes the scope of Nested Phoenix and its functionalities. Section 4 presents the specifications, database implementation and model architecture of the Nested Phoenix Python model before discussing and concluding in Section 5.

## 2. Current trends in modelling cities for resource efficiency

Urban metabolism [5] is the most usual framework used to model resource use in cities. However, multiple accounting techniques can be used within an urban metabolism framework to account for environmental and materials flows, as well as resource use, greenhouse gas emissions, the emissions of pollutants and the indirect use of resources. These flows can also be correlated with socio-economic indicators [6] to obtain a more comprehensive understanding of the variables affecting resource efficiency. Two main accounting approaches are discussed here, namely material flows and stocks analysis (MFA) and life cycle assessment (LCA).

MFA consists of quantifying the existing amount of material(s) in a given system which constitutes the ‘stock’ of material at a given time (e.g. the amount of steel in buildings in Manhattan, New York, USA, as of 2022). MFA also quantifies the material inputs and outputs occurring through time, these are referred to as ‘inflows’ and ‘outflows’, respectively. MFA can also quantify material flows within the system, by reattributing a part of the outflows as inflows. MFA can be coupled with spatial information to geo-reference where these stocks exist and where and when the flows occur, but this spatialisation is not systematic (as discussed below). MFA is usually applied at the larger scales of the built environment (i.e. cities, regions, countries, continents, world).

LCA is a method to quantify the environmental performance of a product or process throughout its life cycle stages, by quantifying its inputs (e.g. energy) and outputs (e.g. greenhouse gas emissions) and converting these to environmental effects (e.g. global warming). LCA is implemented in ISO 14040 [7] as a generic framework, in EN15978 [8] specifically for buildings, and in EN15840 + Annex 2 [9] for construction products. LCA is often conducted at a material/product scale or at the building scale, but less so at the neighbourhood, city, or larger scales.

A range of review papers have investigated the current science on material stocks and flow analysis of built stocks [10, 11], life cycle assessment of buildings and neighbourhoods [12, 13], dynamic life cycle assessment [14-16], and their combinations. Recent review papers agree on the need to develop consistent, bottom-up, spatially-explicit models that combine material flow analysis and life cycle assessment, to produce a more comprehensive and integrated environmental assessment of built stocks. This paper provides a first overview of such a model that covers existing gapes (See Figure 1).

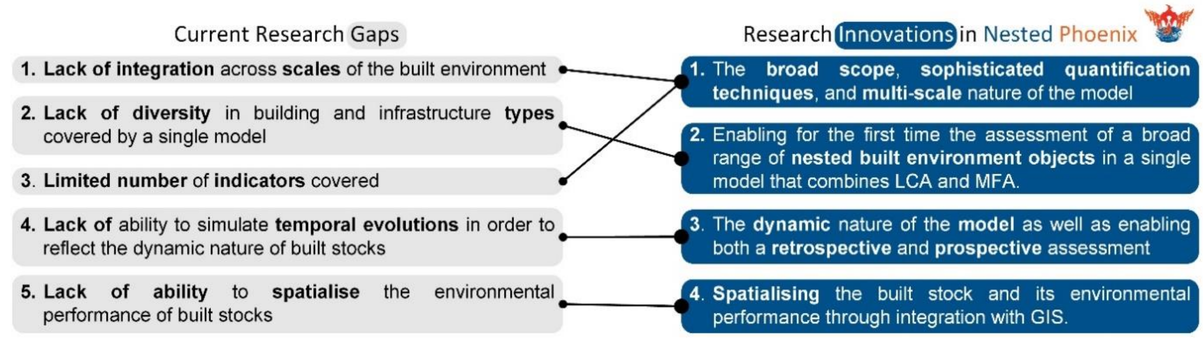

**Figure 1.** Research gaps and associated innovations in the Nested Phoenix model.

## 3. Scope, functionalities and development approach of the Nested Phoenix model

### *3.1. Scope of the Nested Phoenix model*

Nested Phoenix is an integrated Python model to conduct a multi-scale material stocks and flow analysis and life cycle assessment of built assets, across multiple environmental flows. The proposed model considers the existing material stock and expected material replacements over time and associated initial and recurrent embodied environmental flows associated with construction materials in buildings and infrastructure assets. In addition, it considers operational environmental flows linked to the use of buildings (e.g. heating, cooling, lighting, domestic hot water, appliances, etc.) and infrastructure (e.g. lighting). Moreover, the model quantifies mobility-related environmental flows of the occupants of residential buildings, including direct (e.g. using electricity to move a car) and indirect flows (e.g. manufacturing an electric car). This model is applied to a suite of nested built objects, from construction materials to neighbourhoods and cities.

### *3.2. Functionalities*

Table 1 presents a selection of the functionalities of the Nested Phoenix model and provides the main objective for each. The multi-scale nature of the model, its prospective and retrospective aspects, its integration with GIS, its reliance on hybrid life cycle assessment and its integration of multiple built assets provide a high level of sophistication. This enables a dynamic spatio-temporal capacity and comprehensive approach to the environmental assessment of built stocks.

**Table 1.** Proposed objectives and functionalities of the Nested Phoenix model integrating material flow analysis and life cycle assessment of urban built stocks

| Objective | Selected core functionality |
|---|---|
| Assessing the environmental performance of built stocks using a single model | Integrated material stocks and flow analysis and life cycle assessment |
| Assessing embodied environmental flows comprehensively and avoiding the truncation error of process analysis | Hybrid life cycle assessment |
| Enabling actors of the built environment to model different building types and infrastructure assets. | Encompassing multiple built asset types |
| Providing actors of the built environment with a powerful means to understand environmental performance across space and time | Multi-scale assessment, including slicing results using multiple spatio-temporal dimensions |
| Enabling rapid analyses and uptake of the model | Quick and efficient algorithms |
| Enabling testing multiple scenarios and evolution of parameters, predefined values and/or user-defined | Flexible data input, including for dynamic modelling aspects |
| Spatialising built stocks, their material stocks and flows and their environmental performance | Integration with geographic information systems |
| Enable other researchers to build upon the work, use it and expand on it | Transparency and modularity of the model |

### *3.3. Proposed implementation solutions*

Nested Phoenix is coded using Python [17], the high-level programming language. With its readability, simplicity, speed of development, and maintainability, Python is a highly suitable programming language for research projects that are open-source, because it enables a higher level of understanding of the code by the public and also facilitates its use and uptake by fellow researchers.

Python enables the use of a larger set of powerful packages that can be used for data science and integration with other tools, such as GIS. As such, Nested Phoenix capitalises on one of the most widely used and accessible programming languages as well as some powerful packages to provide a state-of-the-art model for quantifying environmental performance in the built environment. In conjunction with Python, Nested Phoenix relies on a relational database engine that is managed directly by the Django package used (see Section 4.3. ).

Nested Phoenix relies on object-oriented programming [18] to enable a modularisation of its architecture. We purposely choose to implement Python objects that reflect the reality of the built environment and its actors to make the code more understandable and intuitive (see Section 4.2. ).

## 4. Specifications of the Nested Phoenix Python model

This Section focuses on the specifications of Nested Phoenix and its Python implementation.

### *4.1. Core Python packages*

Nested Phoenix relies on a range of packages that enable vectorised calculations at scale. It uses Pandas [19] and NumPy [20] for the calculation of large matrices, Django [21] for managing the database through its Object-Relational Mapper, for web interfacing purposes and for interfacing with GIS data.

These packages, among others, enable Nested Phoenix to capitalise on their data structures, e.g. DataFrames in Pandas, Arrays in NumPy, Models in Django, to perform algorithmic operations at scale in optimised computational complexity. We follow best practice in terms of implementation to limit the use of resources and memory.

### *4.2. Overall design approach and philosophy*

Nested Phoenix is pedagogical in its design approach and philosophy and intentionally mimics the real world in the way it names objects, data structures, and agents, therefore making the code intuitive and logical.

Nested Phoenix relies on data extracted from the complex network of tables in its database. The database implementation follows the same principles in terms of representing reality, with a table focusing on each component of the built environment, in addition to other more generic tables. The database structure is detailed in Section 4.3.

Nested Phoenix implements different actors of the built environment and even common documents as so-called ‘classes’, an implementation that groups data and algorithms. For example, Nested Phoenix has a *Quantity Surveyor* class that generates a *Bill Of Quantities* which is provided to the *ESD Consultant* to quantify embodied environmental flows. More details on the model architecture are provided in Section 4.4.

### *4.3. Database implementation*

The database is implemented using Django [22] that interfaces with a Relational Database Management System [23], e.g. MySQL, PostgreSQL, etc. Relational databases offer multiple advantages for data that are consistently structured and interlinked as data are not replicated, are stored once in the relevant table, and are optimised for size and memory because they are consistent. Data in Nested Phoenix are a good candidate for a relational database because they are highly interlinked through nesting (e.g. materials in elements in assemblies) and are consistent (e.g. a material always has a name, category, type, functional unit, and density).

Figure 2 depicts the current implementation of the Nested Phoenix database simplified. It consists of a series of interlinked tables, from the micro scale of the built environment (e.g. Materials (A3)) to entire built areas at the macro scale (e.g. Areas (D3)). It is important to flag that archetypes differ from buildings and infrastructure assets by notably not including geometric information. This means that the same building archetype ‘e.g. suburban detached house 1970 single storey’ can be applied to buildings with different geometries but all sharing the same assemblies for their construction. This decoupling enables significant flexibility in the model and enables a building-by-building modelling of geometry.

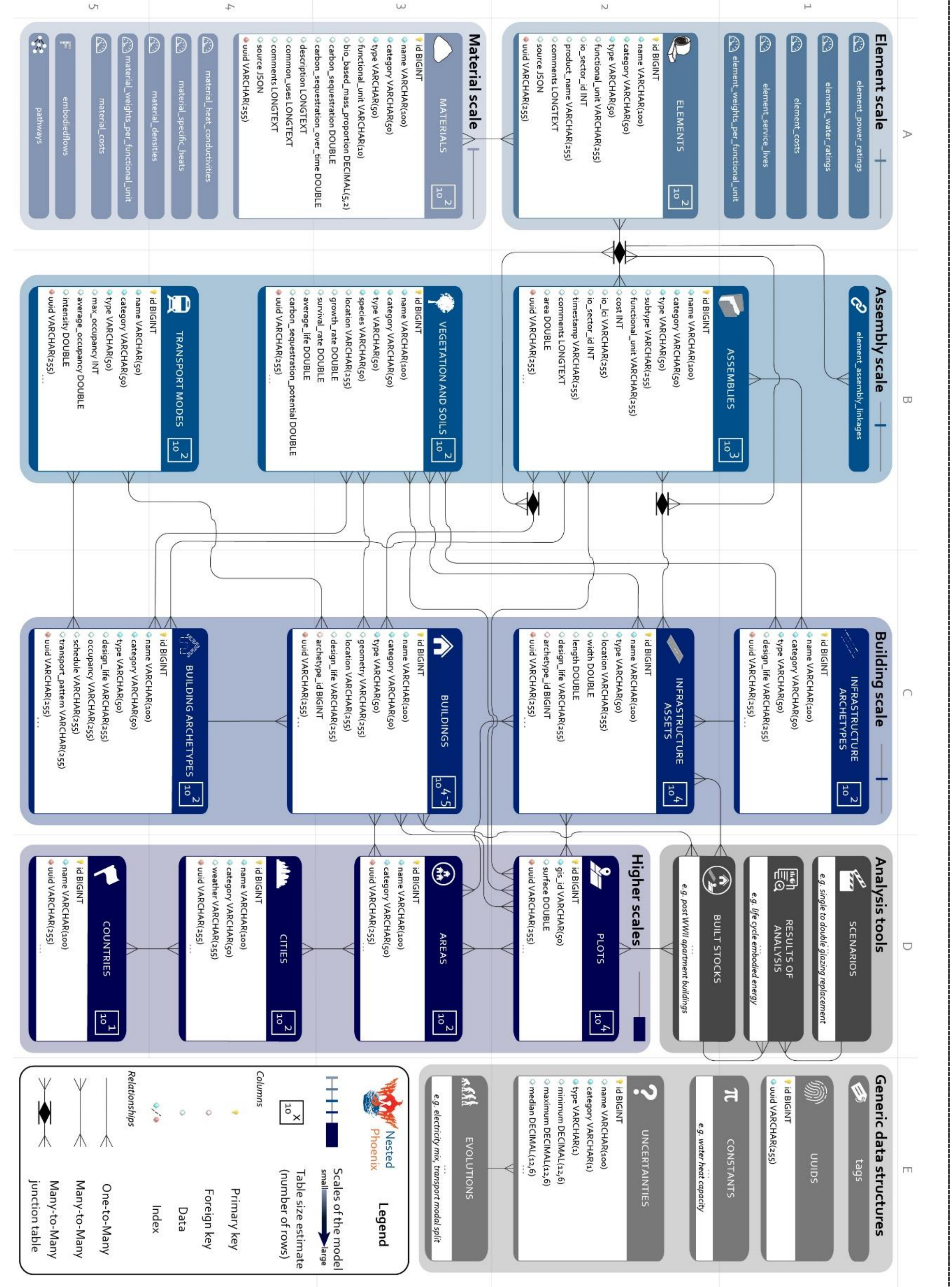

**Figure 2.** Database implementation for the Nested Phoenix model

Buildings (C3) and infrastructure assets (C2) are allocated to a plot of land (D2), which can also include vegetation and soils (B3) as well as assemblies (B2), such as benches, playgrounds, etc. Built or unbuilt plots (D2) are grouped into built areas (D3) which is the largest scale of parametrization of the model. The city (D4) and country (D5) scales, are used as static data containers that provide top-down inputs into the model. Tables in the Analysis Tools group (D1) are used to store assessment parameters and results. Generic data structures (E1-3) are used throughout the model, where necessary. It is important to note that we only display selected parameters for each table and that some tables have dozens of fields in them (e.g. the building table (C3)).

The nested architecture of the Nested Phoenix database not only replicates what occurs in reality but also enables the model to run sensitivity analyses to values by modifying them in one place, enables tracing the data source very easily and avoids duplicating data. For example, to calculate the life cycle embodied energy of the material ‘aluminium’ in an urban area, the code will flow from the results of analysis (D1) to the built stock (D1) to the composing buildings (C3) and infrastructure assets (C2), into their constituting assemblies (B2), into their composing elements (A2), into aluminium-based materials (e.g. aluminium bar, aluminium powder coated, etc.), into their embodied flows (A5). The diversity of data linkages demonstrates the level of sophistication and modelling enabled by Nested Phoenix.

### *4.4. Model architecture*

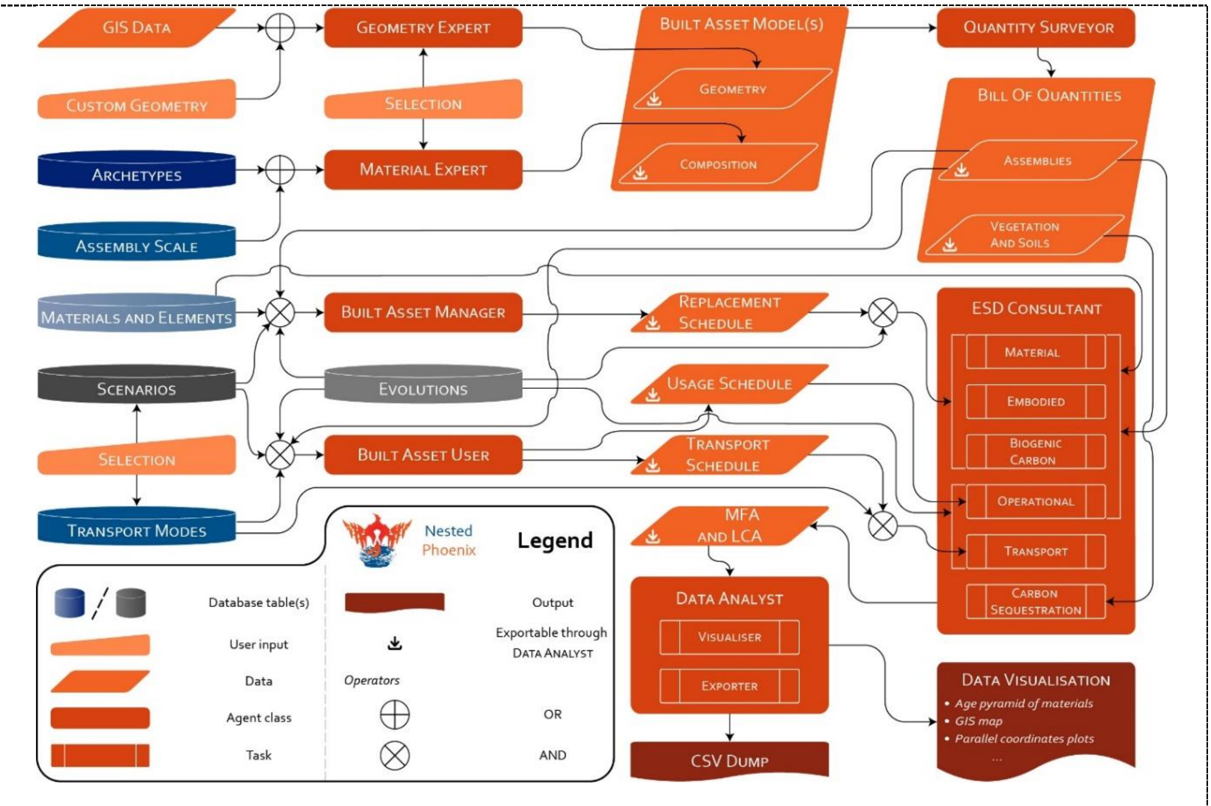

**Figure 3.** Simplified architecture of the Nested Phoenix model.

Figure 3 represents the simplified architecture of the Nested Phoenix model. Inputs flow from the database tables described in Section 4.3. and/or user selections. It is important to flag that user inputs are kept to a minimum where possible and that most data points flow from the databases. The user is able to override calculations at any step if more accurate data are available (e.g. post-occupancy energy use). Agent classes handle the input data and process them as further inputs for other agents.

A typical use case would be the quantification of the material stocks and flow and life cycle environmental performance of a neighbourhood. This use case is used to explain the architecture of the model. Firstly, the user selects the GIS data representing the neighbourhood, which includes the basic geometry of built assets (e.g. shoe box models). The user allocates buildings to predefined archetypes or defines new building or infrastructure archetypes. The *Geometry Expert* class processes the GIS inputs and generates a more detailed geometry of the building/infrastructure asset and the *Material Expert* provides the assembly composition of these built assets (e.g. windows are double-glazed 4:12:4 mm with timber frames and a steel lintel). The combination of the geometry and composition produces built assets models which are handed over to the *Quantity Surveyor* which generates two *Bill Of Quantities* (one for assemblies and the other for vegetation and soils). These *Bill Of Quantities* are provided to the *Built Asset Manager* class which combines that information with the service life of materials and elements coming from the database, potential user-defined scenarios of material makeups over time, and potential parameter evolutions (related to built assets, to the occupants or general) selected by the user to generate a *Replacement Schedule* of materials, elements and assemblies. In parallel, the *Bill Of Quantities* is provided to the *Built Asset User* which combines this information with *Scenarios*, *Evolutions* and *Transport Modes* to generate usage and/or transport *Schedules* of buildings and their occupants. The *Replacement Schedule*, *Evolutions*, *Usage and/or Transport Schedules* and *Bill Of Quantities* are provided to the main class, the *ESD Consultant*, which computes the material stocks and flows, life cycle embodied environmental flows, biogenic carbon in bio-based materials, operational flows, transport flows and the carbon sequestration in soils and vegetation. The MFA and LCA results are provided to the *Data Analyst* that enables visualising data and enable data exports across the model.

This modular architecture that replicates the real world is useful for further developing the model. For instance, a *Structural Engineer* agent could be consulted by the *Quantity Surveyor* to produce structural quantities that are dimensioned according to structural standards (e.g. Eurocode), instead of using an archetypal approach. Additional environmental flows or cost can easily be added in the databases and integrated to the model.

## 5. Discussion and Conclusion

This paper has presented the Nested Phoenix Python model to quantify material stocks and flows and the life cycle environmental performance of urban built stocks. The paper showcases its advanced functionalities, its sophisticated and comprehensive approach, and summarises its implementation.

Nested Phoenix addresses existing gaps (see Figure 1) in environmental performance models at the urban scale by implementing spatialisation, using hybrid life cycle assessment, implementing a bottom-up approach to quantifying and reporting on environmental performance using a dynamic approach, and including both buildings and infrastructure assets. These features enhance environmental modelling of built stocks, as compared to existing models such as [24], [25] or [26].

Any model is only a simplified representation of reality and as such it suffers from limitations. Nested Phoenix would benefit from a more detailed 3D representation of buildings and infrastructure assets, more detailed models for operational energy use, and a more detailed model for modelling mobility-related flows. The approximation error due to the use of archetypes to represent buildings may also be significant. Future research includes finalising the development, testing on case studies, deploying at scale and collecting usage feedback from users to further improve the model and implement new features. By developing models that can reliably yet comprehensively assess the life cycle environmental performance of built stocks, decisions that result in net environmental gains can be made to help mitigate the anthropogenic climate emergency and resource depletion.

## Acknowledgments

This work was supported by the Fonds de la Recherche Scientifique - FNRS under Grant n° F.4547.21, titled *Nested Phoenix*. More information about the project is available on www.nestedphoenix.com.

## Authors contributions

AS received funding for the research. AS, KS and GM conceptualised the research. GM and AS wrote the Python code. AS wrote the first draft of the paper. GM made the figures with the consultation of AS and KS. GM and KS reviewed the paper.

## References

[1] Grubler A, Bai X, Buettner T, Dhakal S, Fisk D J, Ichinose T, Keirstead J E, Sammer G, Satterthwaite D, Schulz N B, Shah N, Steinberger J and Weisz H 2012 *Global Energy Assessment - Toward a Sustainable Future,* (Cambridge University Press, Cambridge, UK and New York, NY, USA and the International Institute for Applied Systems Analysis, Laxenburg, Austria pp 1307-400

[2] Krausmann F, Wiedenhofer D, Lauk C, Haas W, Tanikawa H, Fishman T, Miatto A, Schandl H and Haberl H 2017 Global socioeconomic material stocks rise 23-fold over the 20th century and require half of annual resource use *Proceedings of the National Academy of Sciences* **114** 1880-5

[3] Acuto M, Parnell S and Seto K C 2018 Building a global urban science *Nature Sustainability* **1** 2-4

[4] Stephan A, Crawford R H, Bunster V, Warren-Myers G and Moosavi S 2022 Towards a multiscale framework for modeling and improving the life cycle environmental performance of built stocks *J Ind Ecol*

[5] Kennedy C, Pincetl S and Bunje P 2011 The study of urban metabolism and its applications to urban planning and design *Environmental Pollution* **159** 1965-73

[6] Athanassiadis A 2016 Towards more comprehensive urban environmental assessments: exploring the complex relationship between urban and metabolic profiles. In: *ULB-BATir - Building, Architecture and Town planning and UoM-ABP - Architecture, Building and Planning,* (Brussels: Université Libre de Bruxelles and The University of Melbourne)

[7] International Standard 14040 2006 Environmental management - life cycle assessment - principles and framework. (Geneva: International Organization for Standardization (ISO))

[8] European Standard 15978 2011 Sustainability of construction works -Assessment of environmental performance of buildings - Calculation method. (Brussels: European Committee for Standardization (CEN)) p 66

[9] European Standard 15804 + Annex 2 2019 Sustainability of construction works - Environmental product declarations - Core rules for the product category of construction products. (Brussels: European Committee for Standardization (CEN)) p 65

[10] Lanau M, Liu G, Kral U, Wiedenhofer D, Keijzer E, Yu C and Ehlert C 2019 Taking Stock of Built Environment Stock Studies: Progress and Prospects *Environ Sci Technol* **53** 8499-515

[11] Augiseau V and Barles S 2017 Studying construction materials flows and stock: A review *Resources, Conservation and Recycling* **123** 153-64

[12] Mastrucci A, Marvuglia A, Leopold U and Benetto E 2017 Life Cycle Assessment of building stocks from urban to transnational scales: A review *Renew Sust Energ Rev* **74** 316-32

[13] Röck M, Baldereschi E, Verellen E, Passer A, Sala S and Allacker K 2021 Environmental modelling of building stocks – An integrated review of life cycle-based assessment models to support EU policy making *Renew Sust Energ Rev* **151** 111550

[14] Beloin-Saint-Pierre D, Albers A, Hélias A, Tiruta-Barna L, Fantke P, Levasseur A, Benetto E, Benoist A and Collet P 2020 Addressing temporal considerations in life cycle assessment *Science of The Total Environment* **743** 140700

[15] Göswein V, Silvestre J D, Habert G and Freire F 2019 Dynamic Assessment of Construction Materials in Urban Building Stocks: A Critical Review *Environmental Science and Technology* **53** 9992-10006

[16] Arehart J H, Hart J, Pomponi F and D'Amico B 2021 Carbon sequestration and storage in the built environment *Sustainable Production and Consumption* **27** 1047-63

[17] Python Software Foundation 2022 Python programming language - official website.

[18] Phillips D 2010 *Python 3 Object-Oriented Programming* (Birmingham: UK: Packt Publishing)

[19] The pandas development team 2022 pandas-dev/pandas: Pandas. Zenodo)

[20] Harris C R, Millman K J, van der Walt S J, Gommers R, Virtanen P, Cournapeau D, Wieser E, Taylor J, Berg S, Smith N J, Kern R, Picus M, Hoyer S, van Kerkwijk M H, Brett M, Haldane A, del Río J F, Wiebe M, Peterson P, Gérard-Marchant P, Sheppard K, Reddy T, Weckesser W, Abbasi H, Gohlke C and Oliphant T E 2020 Array programming with NumPy *Nature* **585** 357-62

[21] Django Software Foundation 2022 Django. (Lawrence, Kansas: Django Software Foundation)

[22] Bayer M 2012 *The Architecture of Open Source Applications Volume II: Structure, Scale, and a Few More Fearless Hacks,* ed A Brown and G Wilson (Mountain View: aosabook.org)

[23] Harrington J L 2016 *Relational database design and implementation : clearly explained* (Cambridge, MA: Elsevier)

[24] Huang B, Xing K and Pullen S 2017 Carbon assessment for urban precincts: Integrated model and case studies *Energ Buildings* **153** 111-25

[25] Stephan A and Athanassiadis A 2017 Quantifying and mapping embodied environmental requirements of urban building stocks *Building and Environment* **114** 187-202

[26] Lausselet C, Ellingsen L A W, Strømman A H and Brattebø H 2019 A life-cycle assessment model for zero emission neighborhoods *J Ind Ecol* **24** 500-16