An automated tool for Urban Building Energy Modelling: from sparse datasets to CityJSON

1st Pietro Rando Mazzarino
Politecnico di Torino
Turin, Italy
Email: pietro.randomazzarino@polito.it

2nd Salvatore Finocchiaro
Politecnico di Torino
Turin, Italy
Email: salvatore.finocchiaro@polito.it

3rd Luca Barbierato
Politecnico di Torino
Turin, Italy
Email: luca.barbierato@polito.it

4th Daniele Salvatore Schiera
Politecnico di Torino
Turin, Italy
Email: daniele.schiera@polito.it

5th Lorenzo Bottaccioli
Politecnico di Torino
Turin, Italy
Email: lorenzo.bottaccioli@polito.it

6th Edoardo Patti
Politecnico di Torino
Turin, Italy
Email: edoardo.patti@polito.it

***Abstract*—Urban Energy System Models (UESMs) development has become necessary to conduct quantitative performance analysis and support decision-making for adopting local energy-saving strategies. UESMs require a large amount of data from different sources. Designers and planners have to deal with several challenges related to data accessibility, availability, and standardisation. Often when modeling Urban Energy System (UES) for simulation purposes, lack of data, data incompleteness and data heterogeneity are major obstacles. In addition to that, available datasets or City Information Model (CIM) usually follow different standardisation and formats, making reproducibility on different simulation engines difficult. This work aims at developing a modular tool for creating standard CIMs with very few input constraints. The purpose is to create a flexible and configurable workflow to transform sparse data into a complete information model for urban energy simulations. The framework has been tested on an Italian study case, relying on a few georeferenced information, census data, and a power grid case file. The tool is designed to build the information model and to configure it based on the data availability of the users, in case some information is missing the tool performs data gathering and data filling through theoretical and statistical assumptions. This workflow has been designed to prepare information model for energy simulation in urban contexts thus tackling some specific aspects (e.g. building envelopes, utility networks, building systems, etc.), but its modularity allows further extensions and easy model integration.**

***Index Terms*—Urban Energy System, Urban Energy System Modelling, Urban Building Energy Modelling**

I. INTRODUCTION

The increasing awareness of urban areas' responsibilities in mitigating global greenhouse gas emissions and pollution reduction pushes the need for energy-saving strategies planning and adoption. As a result, modeling and simulating Urban Energy Systems (UESs) become crucial for carrying out analyses, demand forecasting, and management optimization to support decision-making. The discipline of modeling UES is commonly referred to as Urban Energy System Model (UESM), although in literature, another widely-used term is Urban Building Energy Modeling, which emphasizes building description. Throughout this article, we will utilize the term UESM to encompass the broader concept, considering that it allows for the inclusion of detailed building descriptions as well.

Authors in [1] classified the different approaches adopted for UESM, individuating two main groups: top-down and bottom-up approaches. In particular, the bottom-up approach can be further divided into data-driven and simulation methods. Focusing on the simulation methods, the general approach can be depicted in Figure 1. Chosen a real-world UES, all the needed information must be gathered and processed to be merged in a complete information model (e.g. CIM), which will be the starting point for instantiating and performing simulations. An extensive literature review exists for this field [1]–[5] and the vast majority of analysed tools tackle all the steps presented in Figure 1. This leads to a variety of independent tools in which the design of the data management step (see Figure 1) is strictly tailored to each tool with its specific macro objectives and purposes. Thus, standardisation in describing information models is missing, which, in our view, is a strong limitation, as information cannot be shared among different tools. For this reason in this work, we will mostly focus on the step in between the real-world case study (from a single building to an entire district with utility networks) and its representation as an information model. From this perspective, common challenges and gaps have been highlighted in all the previously mentioned reviews. The first real challenge is related to the difficulty of finding quality data, with proper granularity and consistency. This holds in particular when dealing with a high complexity system such as UES, which encompasses different fields of expertise [6] and is described by a high number of heterogeneous variables [4]. Furtherly, the second challenge is related to the lack of centralized standards, for both input data collection and output model information. This makes it difficult to replicate case studies when changing or testing different simulation engines. Strictly related to this last challenge it is possible to highlight a third missing aspects, the lack of automated tools for simulation engine agnostic CIM generation.

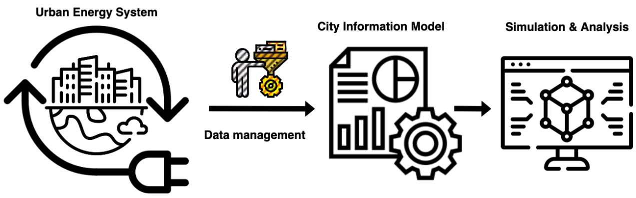

Fig. 1: Urban energy modeling generic workflow

Therefore, in this work, we propose a flexible and modular tool addressing the data management and creation of CIMs for complex UESs (see Figure 1). The proposed tool aims at creating standardized CIMs for the chosen case study area, independently of the kind of input quality given by the user (e.g. researchers, administrators, etc.). Indeed, the proposed solution does not only convert inputs into a standard CIM format but it is also able to retrieve missing data through automatic downloading, theoretical assumptions, and statistical estimations.

The structure of this paper is the following: Section II reports a review of the state of the art; Section III explains in detail the tool workflow and the methodology implemented by the proposed solution; Section IV reports a case-study application and an output description and finally Section V presents our final considerations.

II. Related works

As already mentioned in the introduction of this manuscript, it is possible to highlight three main challenges or gaps in the UESM field:

- Lack of automated tools for CIM generation that must be decoupled from specific simulation engines;
- Lack of Data availability, often it is impossible to simulate energy case studies for lack of data, which can range from complete absence to partially complete datasets;
- Lack of standards when dealing with information models for the description of real-world case studies.

Most of the reviewed solutions in the literature cover the full workflow presented in Figure 1, thus, even if they present automated solutions for data management, the information models they use or simply the methods used for storing data do not allow reproducibility. Indeed, in the vast majority of literature solutions, the data management aspects are designed for the specific needs of the simulation engine. This led to a huge variety of non-standardized data structures that cannot be used elsewhere and which usually do not contain all the information needed for describing a proper UES.

Authors in [7] introduced *"City Building Energy Saver (CityBES)"*, a web-based tool that automatically generates and simulates Urban Building Energy Models using EnergyPlus [8], relying on city building datasets and user-selected Energy Conservation Measures. It supports the CityGML standard and performs some data-filling, but it is only used for building geometries and does not consider utility networks such as the power grid. Takase et al. [9] developed a system for the automatic generation of 3D city models using laser profiler data, 2D digital maps, and aerial images. However, this system is based on data inputs that are very detailed and difficult to retrieve. Ang et al. [10] introduced *"UBEM.io"*, a web-based framework to generate UBEMs in an automated and scalable fashion rapidly. UBEM.io is designed to help any city or municipality conduct low-cost and fast energy and carbon emissions case studies. However, it requires city spatial data management by the urban planner or the administration and the technical know-how to update and organize the Geographic Information System (GIS) files to develop a compliant UBEM pipeline. Ferrando et al. [5], [11] provided a practical overview of the different methods and approaches involved in the UBEM field. Their work compares the available and most relevant UBEM tools, assessing their potential and limitations. As a result, they found out that the main challenge for users in choosing an appropriate UBEM tool remains decision-making for balancing the level of complexity, accuracy, usability, and computing needs. Fonseca et al. [12] present a simulation software *City Energy Analyst*, which provides a series of helpers and a pipeline for data integration that are clear and user-friendly, especially thanks to a very detailed graphic interface, but the data structure does not rely on any standard and it is custom made.

Focusing on the third highlighted gap, i.e. lack of standards, the representation of knowledge is a key aspect when modeling and simulating UESs. Storing information is usually custom made but as highlighted in [13]: Representation and storage of 3D city information must comply with standards that provide common languages for information exchange and sharing purposes. In general, the most used information models in the urban context are Building Information Model (BIM) and CIM. The first refers to both indoor and envelope representations of buildings, including geometric, quantitative, and qualitative information. The latter refers, instead, to urban environments with related features and objects. Indeed, BIMs are mostly used for describing very detailed individual buildings while with CIMs the individual building details are often sacrificed for the broader urban perspective. Among all the open source solutions for information models in the urban context, as stated in [14], CityGML [15] represents a very attractive solution that combines 3D information and semantic information in a single data model. In addition, the possibility to extend the standard information model with Application Domain Extensions (ADEs) [16] allows this standard to be used not only for building level details but also as a proper CIM integrating information related to utilities and energy. In particular, the Energy ADE [17] and the Utility Network ADE [18] allow us to model, respectively, features and properties for energy simulations and utility networks inside the 3D city model. Besides being one of the most widespread standards for these kinds of applications, CityGML also offers other features that strongly enhance its attractiveness and make us choose it as a standard for our work. Firstly, it provides a conceptual structure for sharing, representing, and storing 3D city models [19]. Secondly, there exists a JSON-based version of CityGML called CityJSON [20], [21], which eases the visualization, modification, and exchange of this data format.

In conclusion, we propose an automated and independent tool that allows the automatic creation of a standardized CIM starting from sparse information. It allows the user to input partially complete data as well as minimum information about the case study area and to process it to return a complete data information model for energy simulation in an urban context. In addition to that, the high modularity of this tool allows easy integration of new and more complex data filling methods.

III. Methodology

The proposed solution implements several modules to meet the need for flexibility to model UESs. It offers a predefined workflow that, depending on the user inputs, can be started at any point enabling both automatic and user defined characterisation of the final information model. Our solution enables the integration of heterogeneous data inputs returning, as an output, a complete city information model that encompasses low availability of data. The general idea has been to carry on the flow independently on whether data are available or not. For this purpose, a *bottom-up* methodology has been mixed with statistical assignments and theoretical assumptions to fill in missing information (e.g. building heights, envelope types, etc.) and achieve as much consistency as possible with reality. The architecture, as shown in Figure 2, is divided into two main layers: i) *Data Source layer*; ii) *Model Extraction layer*. The tool provides a specific workflow that follows from raw data inputs to a complete output, i.e. CIM in CityJSON. Nevertheless, the modularity of the framework enables interaction with the intermediary steps of the workflow and this allows running the pipeline from any point. This means that in case the user has already some detailed information it is possible to bypass the related data-filling modules and run the rest of them. Therefore, the whole process may be executed in a single long run or by exploiting the functionalities of each block individually, providing input at any level, with no need to ensure their completeness. This modular execution also enables proper inspection of intermediary outputs. The rest of this section describes each layer of the proposed solution.

A. Data Source Layer

In the Data Source Layer, thanks to configuration parameters, it is possible to set up the starting condition of both raw inputs and setup for the desired final CIM output. By referring to Figure 2, we can observe the diverse nature of the data handled by the proposed solution. Besides the semantic difference of these data it is also important to notice that some of these can be uploaded as raw inputs (e.g *Utility Network Data*, *Shapefile Data*), others are automatically retrieved by external web data sources (e.g. *OSM Data*), some of them are strictly related to the specific case study (e.g. *DTM/DSM*) others can be used for multiple case studies having a Regional

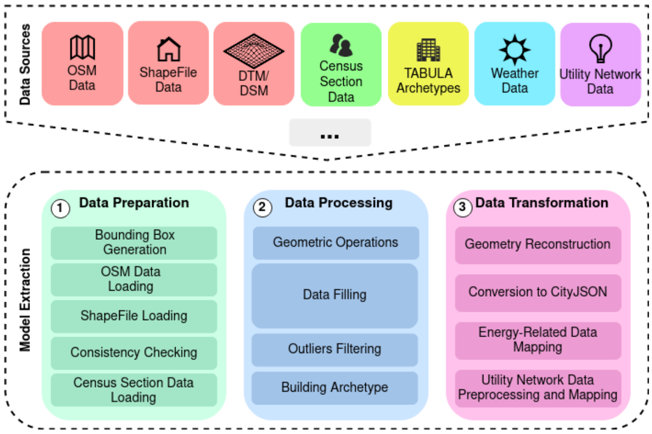

Fig. 2: Architecture Overview

or National validity (e.g. *Census Section Data* and *TABULA archetypes*).

OSM Data are geo-referenced polygonal information, retrievable from Open Street Map (OSM), representing the buildings in a defined study area. They usually come with attached information that can be used during the workflow. *ShapeFile Data* are provided by the user and contain geo-referenced information for building descriptions. *DTM/DSM* data represent indeed the Digital Terrain Model (DTM) and the Digital Surface Model (DSM) which are georeferenced models containing the elevation of the ground above the sea level and the elevation of surface elements above the ground. These files can be automatically retrieved and are useful for the building's height estimation. *Census Section Data* provides urban-level statistics at high spatial resolution. This is a powerful source of information, providing census section level information on population, families, and buildings as well as economics and social records. *TABULA Archetypes* [22] provides building national typologies according to sharing features (envelope layering, common HVAC systems, etc.). The proposed solution relies on those archetypes for classifying buildings accordingly, characterizing each of them with energy-related information (e.g. Transmittance values, envelope thermal resistance, etc.). *Weather Data* can be provided or gathered through *Meteostat* API [23], which allows for data retrieval according to the specified resolution and time interval. *Utility Network Data* can be uploaded, supporting the *PandaPower* [24] data-format. This format consists of a series of data frames describing all the network objects from lines to transformers, with this kind of file any LV-MV power grid can be represented. In the case of District Heating, water, or gas network the supported format is *PandaPipes* [25].

Data consistency and configurability are two of the main focus of this solution involving in particular input data. For all the data that are not automatically retrieved by this solution a major problem is to ensure the same nomenclature and structure and thus dedicated modules and configuration files allow to explicitly provide meta description of raw inputs enabling the tool to correctly assess them. In conclusion, the flexibility and the configurability of the platform allow for different inputs to be integrated in the future.

B. Model Extraction - Data Preparation

As shown in Figure 2, *Data Preparation* includes all the modules that are performed at the beginning of the workflow.

The first module, *Bounding Box Generation* in Figure 2, is responsible for delimiting the study area around a specified address. The bounding box can be either a square area centered in the address location or a circular area, in the first case the desired distance fixes the length of the square side while in the second one the radius of the circle.

OSM Data Loading module executes some operations involving, first, a connection to the OSM Application Programming Interface (API) to get raw data referred to the desired study area inside the bounding box. Using the package *OSMnx* [26] requested data are downloaded and then arranged in a GeoDataFrame with a predefined structure.

ShapeFile Loading module crops the input ShapeFile data (if provided) within the previously generated bounding box and stores geometries and associated tabular information into a GeoDataFrame.

Consistency Checking module ensures consistency among the two datasets (the one from OSM and the one from the user input) for later joining. The majority of joining procedures are based on spatial knowledge, thus the control performed by this block focuses on indexing, Coordinate Reference System (CRS), and geometries. In addition to that based on configuration files, it ensures the correct nomenclature for each type of data in both the two GeoDataFrames.

Lastly, *Census Section Data Loading* module allows the user to upload tabular census data. These can provide demographical and economic statistics related to each census zone, they must be georeferenced but they could differ from country to country. To keep consistency a configuration file allows to specify and link the nomenclature for the specific features needed in the characterisation.

To summarise, OSM building data, ShapeFile building data, and census section data are now available as GeoDataFrames. These are the preliminary outputs of *Data Preparation* that will be fed to the core process of this workflow: *Data Processing*.

C. Model Extraction - Data Processing

Data Processing, step number 2 in the workflow in Figure 2, includes all the modules that perform geometric operations and data filling for missing information. The output will be a complete dataset with the desired buildings information and city-wide characterisation, ready to be finally converted into a standard CIM in CityJSON. Firstly, *Geometric Operations* module joins in a single dataset the previously loaded data sources. This step reconstructs a complete georeferenced dataframe with all the buildings in the study area and performs basic geometry calculations such as the gross and net floor area. The coupling of shapefile and OSM data allows us to check for any missing geometry. At the same time, each building is assigned the census section code they belong to.

After that, *Data Filling* module addresses the lack of the following specific information for each individual building in the case-study area: i) *Year of construction*, ii) *Height*; iii) *Number of floors*; iv) *Use destination*; v) *Infiltration rate*; vi) *Desired Demographic information*.

This module parses the geodataframe from the previous module and when some data is missing, it calls the specific helper method to fill it. This process is automated and mostly configurable, especially for the fixed theoretical assumptions (e.g. average infiltration rates). Data filling operation replicability is ensured as this module can be configured to specify statistic data sources and corresponding nomenclature, with the assumption that these new data sources will provide the same information. The data filling process follows different approaches dependending on which data is addressing. For example geometric calculations can be used for assigning the building heights, statistical approach can be used for estimating use destination and theoretical assumptions can fix general average values such as the avergae floor height for the specific case study.

If information about the *Year of construction* is missing, the average age of construction of the census zone, to which the building belongs, is assigned. The value *Height* can be estimated following two approaches. The first relies on average heights per census zone if the data is available. Whilst the second exploits DTM and DSM downloaded by [27]. In the second approach, the difference between DSM and DTM is used to create height points that are assigned and averaged to each building after a spatial join. Those two methods strongly depend on the granularity of the given data sources, which indeed affects the reliability of this outcome. The modularity of the pipeline allows the integration of more complex height estimation procedures. The *Number of floors* is simply assigned by choosing an average floor height and then dividing the total building height by this value. The *Use destination*, usually, is already present in the OSM data, in its absence, it is filled by default values that can be configured. The *Infiltration rate*, which is the air rate that by default enters the building, can be set by the user. The *Demographic information* is statistically derived for each building by analyzing the distribution within each census zone. Among this data, the most crucial factors include the number of families and the types of families residing in each building. In addition, it is possible to specify which extra information to be included in the final output.

The *Outliers Filtering* module allows to clean up the whole dataset. It performs two operations: i) removing entries with wrong or unacceptable values (e.g. buildings too low or too high); ii) selecting buildings with specific features, e.g. residential, or commercial buildings or buildings built before a certain date.

The *Building archetype* module assigns a specific construction archetype to each building based on the *Year of construction* and the number of families. In this way, buildings are classified according to TABULA Physics Library [22]. This classification allows to retrieve from the TABULA dataset information about the envelope, the heating, and the cooling system, as well as possible retrofits over the years.

D. Model Extraction - Data Transformation

Data Transformation step involves the final data transformation into a 3D CIM. CityJSON is the transposition in JSON of CityGML, which exploits XML. Both CityJSON and CityGML are standard data formats to describe city information models that we augmented with additional information about weather and utility distribution networks. We chose JSON because it is lightweight and faster in coding and decoding w.r.t. XML. This step includes four modules as shown in Figure 2. *Geometry Reconstruction* module reconstructs the geometries to be compliant with CityJSON. This format registers each surface as an object of a building object, thus it is needed to compute the exact position of each building vertex and correct surface orientations. This creates a 3D model with a Level of Detail (LoD) of 1.2, meaning that buildings are represented as extruded boxes depending on their height.

Once the geometries are properly reconstructed, *Conversion to CityJSON* module, creates the proper file and populates it with the basic buildings objects. This file is the baseline for our complete CIM and the information belonging to the two ADEs will be mapped on top of it.

Energy-Related Data Mapping module relies on the energy extension schema in [21], which has been developed to comply with the corresponding CityGML ADE specifications requiring that building thermal behavior is modeled.

In addition to energy-related data on the building level, information on utility distribution networks has been integrated at the city level to enable simulation and plan energy-saving strategies accordingly. *Utility Network Data Preprocessing and Mapping* module enriches the model with data provided either by *PandaPower* [24] or *PandaPipes* [25] files. It maps the utility network in the study area and then associates each building to the nearest pod, bus, or injection/extraction node.

IV. Experimental set-up and results

The proposed solution has been tested on a bounding box of about 4 km^2 wide around a square in the city center of our city. In this area, 1533 buildings have been modeled and described as well as the distribution power grid which consists of 1076 buses, 987 lines, and 956 connected generators of different types. Figure 3 shows a small portion of the case study to visualize an example of the categorization of buildings based on use destination and the 3D visualization of this sample. The final CityJSON contains a complete set of information for both buildings and the utility network. More details on all the included features are presented in the schema of Figure 4, thus showing a complete overview of all kinds of data included in the final output. Figure 3 highlights the destination of use categorization, in our specific case we have only considered buildings with a residential purpose. Under this category falls the following Italian-specific categories: *"residential", "housing"* and the mixed-use categories *"residential & commercial", "residential & productive"*. Matching macro categories with a list of National-specific possibilities can be done during the configuration files and, at the filtering stage, the classification is modified accordingly.

Fig. 3: Portion of the case study used as example to show 2D categorization of Use destination and 3D visualization

We have tested our solution on our case study area knowing the real-world information for Building heights, number of floors, and use destinations and we use it as a benchmark. Thus, we did not provide this data in input to our solution as we want to assess the capabilities of our tool in reconstructing this information to build the final CIM. Table I shows the performance of our solution in estimating those values. For the height and the number of floors, we propose as performance metrics the Mean Square Error (MSE), the mean error among all the estimations, and the standard deviation of these errors, while for the use destination, being a categorical feature, we used the mean percentage error. Table I highlights that the height estimation performed well with very low MSE and mean error and a standard deviation that spans between ±4 meters. For the number of floor estimations, the MSE is still low and acceptable. This error strictly depends on approximation as the number of floors is represented as an integer value. By looking at the mean error we can notice that it is still acceptable and around 24.1%. Confirmation of this can be seen in the standard deviation value of ±1.35 floors, indeed in the large majority of cases, the wrong estimation of number of floors deviates from the real-world data of one floor at maximum. Finally, we can see that the percentage error in estimating the use destination is also acceptable and it is around 14%.

Besides the values above mentioned, our real-world dataset also reports the year of construction feature. The reconstruction of this feature follows a statistical approach based on the distribution, per census zone, of buildings constructed in specific time slots. Indeed exact estimation of this feature for each building, following this approach, is very hard. Nevertheless, the principal usage of the year of construction

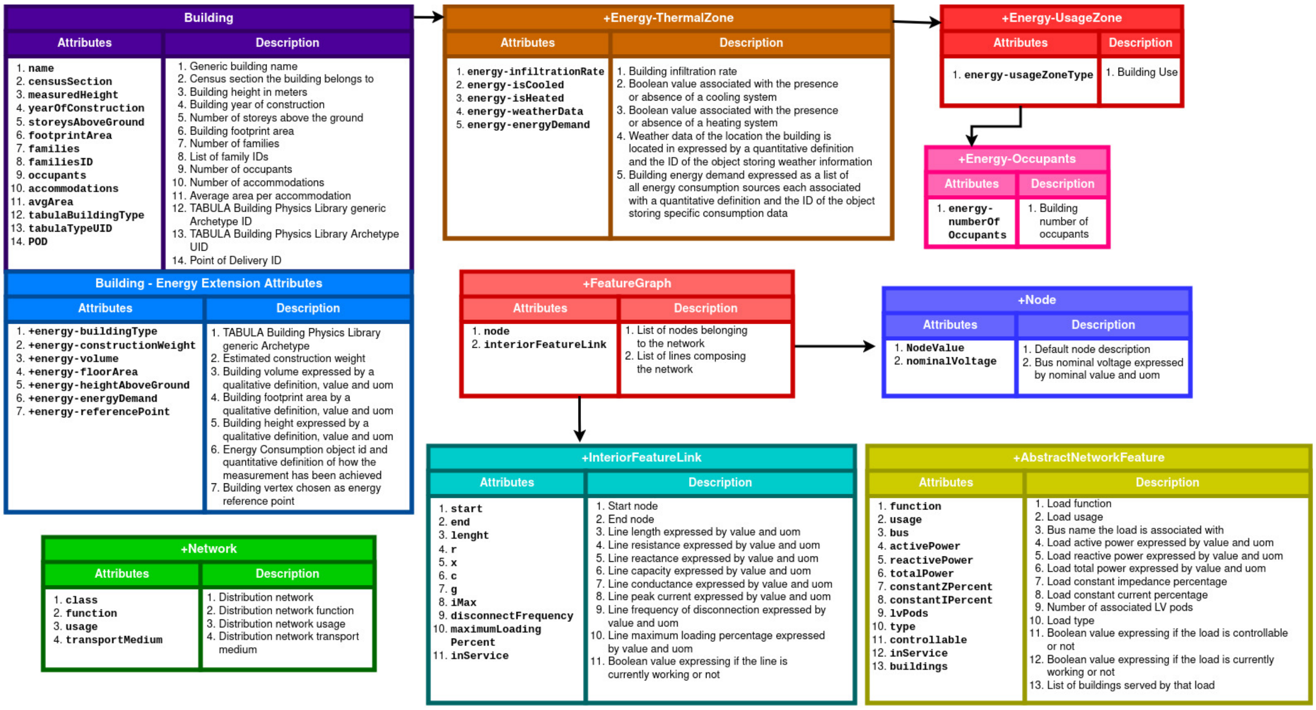

Fig. 4: Data-structure of the pipeline output in CityJSON

	MSE	mean error	standard deviation
Height	1.6e-7	6.6%	+- 4.01 [m]
N° floor	0.0016	24.1%	+- 1.35 [floor]
Use destination	-	14.3%	-

TABLE I: Performance results for Height, N° of floors and Use destination estimation

information is to be used as a matching feature for the TABULA archetypes. This match is performed using time slots rather than exact years, thus, in addition to the year of construction, each building is associated with a specific time slot out of the seven presented by TABULA, i.e. before 1918, 1919-1945, 1946-1961, 1962-1971, 1972-1981, 1982-1991, after 1992. By examining the timeslots, all buildings are appropriately assigned to the slot corresponding to their original year of construction. TABULA provides a collection of building archetypes categorized by year of construction timeslot and typology, with the following typologies being considered: *Apartment Block*, *Multi Family House*, *Terraced House*, and *Single Family House*. Each building is assigned a typology based on configurable rules governing the number of floors and families.

Figure 5 illustrates that the vast majority of buildings in our case study area were constructed before 1919, with a discernible decline in the number of new buildings over time. This characterization aligns with expectations given that the area under investigation is a portion of the historic downtown of a relatively ancient city. Conversely, Figure 6 reveals that Apartment blocks constitute the overwhelming majority of buildings, with no presence of Terraced Houses, which is a fairly realistic representation of the case study area.

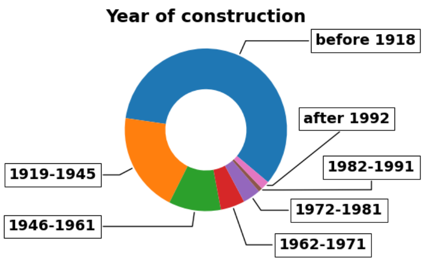

Fig. 5: distribution of buildings based on year of construction time-slot

For what concern the demographic aspects of the case study area that have been reconstructed, in the selected zone around 39789 people are living and they are arranged into around 20294 families (from 1 member up to 5).

In conclusion, the result of this solution offers a complete CityJSON for simulation purposes. The information involved takes into consideration, geometric and georeferenced aspects of buildings and utility networks, and connects them, allowing storage of additional information for the simulation of both buildings and utility networks, i.e. family occupancy of buildings, thermal characteristics, possible HVAC systems, making all of this easily accessible, configurable and modifiable.

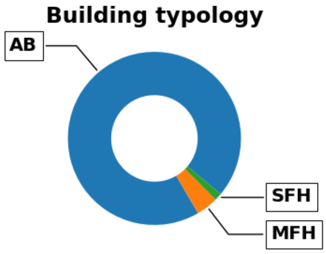

Fig. 6: distribution of buildings based typology: where AB is Apartment Block, MFH is Multi Family House, SFH is Single Family House

V. Conclusions

The proposed solution showcases an automated and modular workflow for urban energy modeling from sparse datasets to a standardized output. The main purpose of the proposed solution is to create a standardized CIM to model information for simulation purposes (e.g. potential analysis, planning for different installations, etc.). The tool modularity allows users to leverage individual components independently based on their specific needs and available inputs. The pipeline has been designed to enhance replicability across case studies and compatibility with downstream tools through its standardized CityJSON output extended with energy and utilities ADEs. This output conforms with established information modeling standards, easing interoperability. The workflow is highly configurable, making it adaptable to different data structures and regions through its user-defined configuration options. Overall, the presented pipeline addresses critical challenges in urban energy modeling related to data availability and standardization. As a next step, integrating more sophisticated modules within the architecture could improve its capabilities - for instance, incorporating a module for the creation of synthetic power grids could address the low availability of this kind of information. In addition, a future work that will be addressed is to enhance building descriptions to achieve a higher LoD and expand some of the modules for demographic data assignation. The modular design of the pipeline leaves room for augmenting its methods to further automate the modeling process from sparse, heterogeneous inputs to standardized scenarios.

Acknowledgment

This publication is part of the project NODES which has received funding from the MUR-M4C2 1.5 of PNRR founded by the European Union - NextGenerationEU (grant agreement no. ECS00000036).

References

[1] N. Abbasabadi and M. Ashayeri, "Urban energy use modeling methods and tools: A review and an outlook," *Building and Environment*, vol. 161, p. 106270, 2019.
[2] J. Keirstead, M. Jennings, and A. Sivakumar, "A review of urban energy system models: Approaches, challenges and opportunities," *Renewable and Sustainable Energy Reviews*, vol. 16, no. 6, pp. 3847–3866, 2012.
[3] T. Hong, Y. Chen, X. Luo, N. Luo, and S. H. Lee, "Ten questions on urban building energy modeling," *Building and Environment*, vol. 168, p. 106508, 2020.
[4] M. Yazdanie and K. Orehounig, "Advancing urban energy system planning and modeling approaches: Gaps and solutions in perspective," *Renewable and Sustainable Energy Reviews*, vol. 137, p. 110607, 2021.
[5] M. Ferrando, F. Causone *et al.*, "An overview of urban building energy modelling (ubem) tools," in *BUILDING SIMULATION CONFERENCE PROCEEDINGS*, vol. 16, 2020, pp. 3452–3459.
[6] P. R. Mazzarino, A. Macii, L. Bottaccioli, and E. Patti, "A multi-agent framework for smart grid simulations: Strategies for power-to-heat flexibility management in residential context," *Sustainable Energy, Grids and Networks*, vol. 34, p. 101072, 2023.
[7] Y. Chen, T. Hong, and M. A. Piette, "Automatic generation and simulation of urban building energy models based on city datasets for city-scale building retrofit analysis," *Applied Energy*, vol. 205, pp. 323–335, 2017.
[8] M. Gerber, "energyplus energy simulation software," 2014.
[9] Y. Takase, N. Sho, A. Sone, and K. Shimiya, "Automatic generation of 3d city models and related applications," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 34, no. 5, 2003.
[10] Y. Q. Ang, Z. M. Berzolla, S. Letellier-Duchesne, V. Jusiega, and C. Reinhart, "Ubem. io: A web-based framework to rapidly generate urban building energy models for carbon reduction technology pathways," *Sustainable Cities and Society*, vol. 77, p. 103534, 2022.
[11] M. Ferrando, F. Causone, T. Hong, and Y. Chen, "Urban building energy modeling (ubem) tools: A state-of-the-art review of bottom-up physics-based approaches," *Sustainable Cities and Society*, vol. 62, p. 102408, 2020.
[12] J. A. Fonseca, T.-A. Nguyen, A. Schlueter, and F. Marechal, "City energy analyst (cea): Integrated framework for analysis and optimization of building energy systems in neighborhoods and city districts," *Energy and Buildings*, vol. 113, pp. 202–226, 2016.
[13] I. Hijazi and A. Donaubauer, "Integration of building and urban information modeling–opportunities and integration approaches," *Geoinformationssysteme*, pp. 42–56, 2017.
[14] I. Prieto, J. L. Izkara, and F. J. Delgado del Hoyo, "Efficient visualization of the geometric information of citygml: application for the documentation of built heritage," in *Computational Science and Its Applications–ICCSA 2012: 12th International Conference, Salvador de Bahia, Brazil, June 18-21, 2012, Proceedings, Part I 12*. Springer, 2012, pp. 529–544.
[15] Citygml. [Online]. Available: https://www.ogc.org/standard/citygml
[16] O. CityGML. (2023) Citygml ades. [Online]. Available: https://www.citygmlwiki.org/index.php/CityGML-ADEs
[17] Citygml energy ade. [Online]. Available: https://github.com/cstb/citygml-energy
[18] Citygml utility network ade. [Online]. Available: https://github.com/TatjanaKutzner/CityGML-UtilityNetwork-ADE
[19] Z. Yao, C. Nagel, F. Kunde, G. Hudra, P. Willkomm, A. Donaubauer, T. Adolphi, and T. H. Kolbe, "3dcitydb-a 3d geodatabase solution for the management, analysis, and visualization of semantic 3d city models based on citygml," *Open Geospatial Data, Software and Standards*, vol. 3, no. 1, pp. 1–26, 2018.
[20] Cityjson specifications 1.1.3. [Online]. Available: https://www.cityjson.org/specs/1.1.3
[21] Ö. TUFAN, "Development and testing of the cityjson energy extension for space heating demand calculation," 2022.
[22] T. episcope project. (2001) Tabula building physics library. [Online]. Available: https://webtool.building-typology.eu
[23] Meteostat developers. [Online]. Available: https://dev.meteostat.net/python
[24] L. Thurner, A. Scheidler, F. Schäfer, J.-H. Menke, J. Dollichon, F. Meier, S. Meinecke, and M. Braun, "pandapower—an open-source python tool for convenient modeling, analysis, and optimization of electric power systems," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 6510–6521, 2018.
[25] D. Lohmeier, D. Cronbach, S. R. Drauz, M. Braun, and T. M. Kneiske, "Pandapipes: An open-source piping grid calculation package for multi-energy grid simulations," *Sustainability*, vol. 12, no. 23, p. 9899, 2020.
[26] G. Boeing, "Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks," *Computers, Environment and Urban Systems*, vol. 65, pp. 126–139, 2017.
[27] Open-elevation api. [Online]. Available: https://open-elevation.com