# D6.4 Plug-in for tailor-based resilience design

MULTICARE

Funded by
the European Union

# D6.4 Plug-in for tailor-based resilience design

| | |
|---|---|
| Dissemination Level: | PU - Public |
| Lead Partner: | UNIROMA1 |
| Due date: | 30 June 2025 |
| Actual submission date: | 30 June 2025 |

## PUBLISHED IN THE FRAMEWORK OF

MULTICARE (Horizon Europe grant 101123467)

## AUTHORS

Giada Formichetti, UNIROMA1
Kyujin Kim, TU Delft
Anna Maria Koniari, TU Delft
Michele Matteoni, UNIROMA1

## REVISION AND HISTORY CHART

| VERSION | DATE | EDITORS | COMMENT |
|---|---|---|---|
| V0.1 | 9 June 2025 | Giada Formichetti (UNIROMA1)<br>Kyujin Kim (TUD)<br>Anna Maria Koniari (TUD)<br>Michele Matteoni (UNIROMA1) | Draft |
| V0.2 | 12 June 2025 | Divyae Mittal (OMRT) | Review |
| V1 | 12 June 2025 | Giada Formichetti (UNIROMA1) | Final version |

## DISCLAIMER

Funded by the European Union

This project has received funding from the European Union under the Horizon Europe Research & Innovation Programme (Grant Agreement no. 101123467 MultiCare). Views and opinions expressed are however those of author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

# Table of contents

## Contents

## LIST OF FIGURES

This project has received funding from the European Union under the Horizon Europe Research & Innovation Programme (Grant Agreement no. 101123467 MultiCare). Views and opinions expressed are however those of author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

## LIST OF TABLES

Funded by the European Union

This project has received funding from the European Union under the Horizon Europe Research & Innovation Programme (Grant Agreement no. 101123467 MultiCare). Views and opinions expressed are however those of author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

# 1. Introduction

## 1.1. MULTICARE project

The built environment is ill-prepared for more frequent and increasingly intense climate-related extreme events. The current building stock is particularly vulnerable because it has limited or no capacity to adapt and recover from extreme events thereby leading to building failures that cause severe socio-economic losses and adversely affecting the health and wellbeing of people. Recent scientific and technological advances in the construction industry provide timely solutions for improving the resilience for specific single hazards (e.g. flood hazard or seismic hazard), but they are often not cost effective, rarely eco-friendly and nearly never address the multiple hazards present in many locations. This is hardly surprising because there is neither a clearly defined framework for quantifying the whole-life socio-economic-environmental impacts of extreme natural events nor tools for assessing the holistic climate resilience of buildings. Consequently, it is currently very challenging to develop/select optimal solutions for real-world multi-hazard scenarios.

MULTICARE will address this challenge directly by developing new multi-criteria decision-support frameworks and providing plug & play technological and digital solutions for improving the resilience of the built environment in a cost-effective, reliable and sustainable manner. The technological solutions consist of multi-functional low-carbon resilient technologies embedded in modular and prefabricated construction for the next generation of high performance and smart buildings, characterized by enhanced safety, energy efficiency, environmental-sustainability, improved quality of life, circularity, and scalability for a broad range of natural events and end-user. The plug & play technologies will be applied to either new multi-story buildings or existing structures by means of low-invasive external interventions. The digital solutions consist of a suite of multi-disciplinary digital services and tools for performing multi-hazard resilience assessment, design, operation and management across multiple scales (material, component, building, neighborhood/city). The new digital tools will enable stakeholders to make informed decisions in the selection of materials/solutions, including for heritage buildings, and support resilient supply chains. The effectiveness of the MULTICARE solutions will be demonstrated through large-scale pilots (3 buildings, 4 neighborhoods/district) in three different European countries carefully selected for their diverse local environmental, social and economic conditions (Italy, Netherlands, Romania). Banks and institutional investors will be engaged to better understand the financial risk reduction value of resilience and update existing and future “green finance” mechanisms that will help to leverage the project results. A user-center, inclusive and participatory approach will be consistently implemented throughout the project to engage citizens and extend the durability of MULTICARE impact.

To achieve these ambitious goals, MULTICARE brings together a unique interdisciplinary Consortium of 21 partners (**Table 1**. Consortium) from 6 different EU countries with strong R&D and practical expertise, who are either established leaders in their sector or agile SMEs

Funded by the European Union

This project has received funding from the European Union under the Horizon Europe Research & Innovation Programme (Grant Agreement no. 101123467 MultiCare). Views and opinions expressed are however those of author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

in emerging fields. Altogether the Consortium members span across the whole technical and value chain required for developing and implementing solutions in terms of design, digitization, manufacturing, construction and monitoring of resilient and sustainable buildings. The Consortium also includes partners with experience in social sciences, user engagement, and training to ensure the success and widespread application of new technologies in local communities. The Consortium will also support clustering activities with other relevant research projects to share knowledge and raise public awareness of building resilience. An international outreach and cooperation strategy will also be implemented to tackle the project challenges.

*Table 1. Consortium*

| Number | Role | Short Name | Legal Name | Country |
|---|---|---|---|---|
| 1 | CO | TU Delft | TECHNISCHE UNIVERSITEIT DELFT | NL |
| 2 | BEN | PFE | PRIEDEMANN FASSADENBERATUNG GMBH | DE |
| 3 | BEN | IES R&D | IES R&D | IE |
| 4 | BEN | INCDFP | INSTITUTUL NATIONAL DE CERCETARE-DEZVOLTARE PENTRU FIZICA PAMANTULUI | RO |
| 5 | BEN | UNIROMA1 | UNIVERSITA DEGLI STUDI DI ROMA LA SAPIENZA | IT |
| 6 | BEN | XLD | X-LAM DOLOMITI SRL | IT |
| 7 | BEN | STRESS | SVILUPPO TECNOLOGIE E RICERCA PER L'EDILIZIA SISMICAMENTE SICURA ED ECOSOSTENIBILE SCARL | IT |
| 7.1 | AE | UNINA | UNIVERSITA DEGLI STUDI DI NAPOLI FEDERICO II | IT |
| 8 | BEN | AMS Institute | STICHTING AMSTERDAM INSTITUTE FORADVANCED METROPOLITAN SOLUTIONS(AMS) | NL |
| 9 | BEN | PMB | MUNICIPIUL BUCURESTI | RO |
| 10 | BEN | ASM | ASM - CENTRUM BADAN I ANALIZ RYNKUSPOLKA Z OGRANICZONA ODPOWIEDZIALNOSCIA | PL |
| 11 | BEN | RoGBC | ASOCIATIA ROMANIA GREEN BUILDING COUNCIL | RO |
| 12 | BEN | RINA-C | RINA CONSULTING SPA | IT |
| 13 | BEN | UTBV | UNIVERSITATEA TRANSILVANIA DIN BRASOV | RO |
| 14 | BEN | ACER | AGENZIA CAMPANA PER L EDILIZIA RESIDENZIALE | IT |
| 15 | BEN | Boom | BOOM BUILDS B.V. | NL |
| 16 | BEN | OMRT | OMRT BV | NL |
| 17 | BEN | ROTHO BLAAS SRL | ROTHO BLAAS SRL | IT |
| 18 | BEN | ARUP | ARUP BV | NL |
| 19 | BEN | Tecuci | MUNICIPIUL TECUCI | RO |
| 20 | BEN | Hölscher | DIPL.-ING. HPLSCHER GMBH & CO.KG | DE |

Funded by the European Union

This project has received funding from the European Union under the Horizon Europe Research & Innovation Programme (Grant Agreement no. 101123467 MultiCare). Views and opinions expressed are however those of author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

## 1.2. D6.4 tasks

D6.4 aims to develop a plug-in/open-source workflow for CAD-based (Rhinoceros) application, to support holistic building design. The main objectives of this deliverable include:

- Developing a user-friendly workflow that assists designers and decision-makers during the early design stages of a building project.
- Integrating various aspects of building performance—such as structural and energy factors—into a unified design tool to promote resilient and sustainable building systems.
- Establishing an automated parametric framework that concurrently evaluates multiple performance metrics and their interdependencies.
- Delivering a scalable and extensible framework that supports the design of resilient new buildings.

## 1.3. Deliverable

As per the Grant Agreement, deliverable for 6.4. is a report. This document includes the link to download the source-code for the workflow developed.

*Table 2. Key deliverable information. Source: MULTICARE Grant Agreement.*

| Deliverable number and name | Work Package nº | Lead Beneficiary | Type |
|---|---|---|---|
| D6.4 – Plug-in for tailor-based resilience design | WP6 | 5 – UNIROMA1 | R – Document, report |

# 2. The Grasshopper-based framework

## 2.1 Introduction to parametric design in Grasshopper

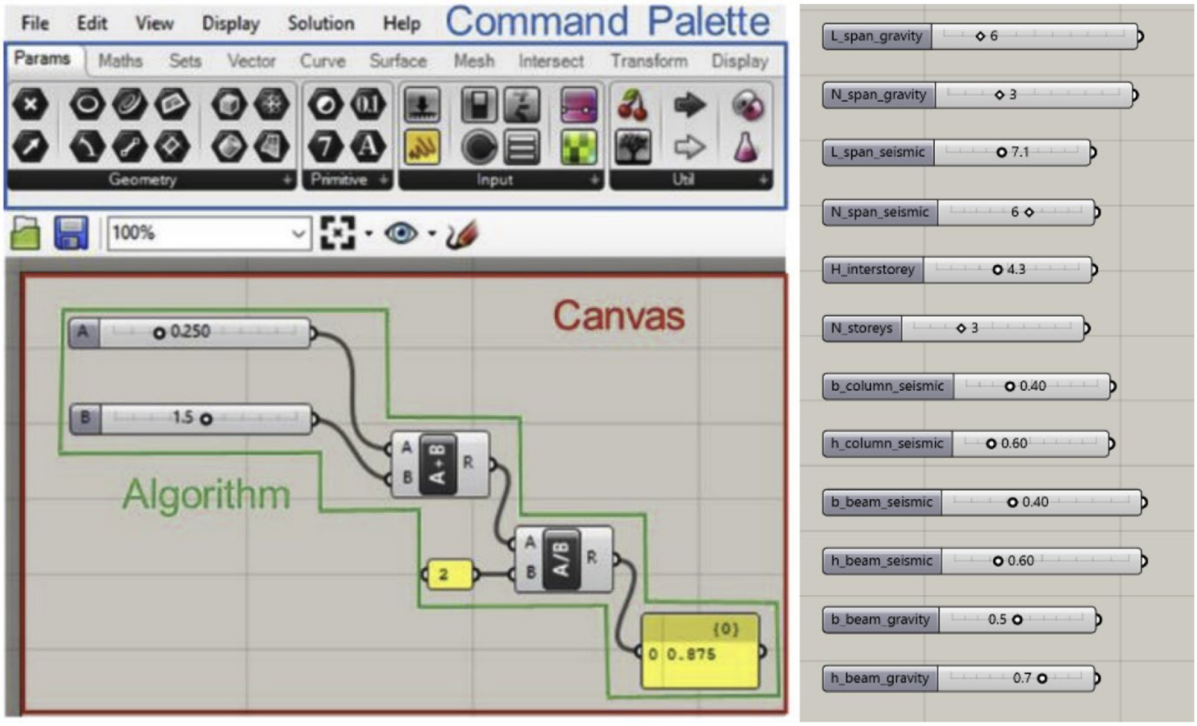
The development of digital tools, relying on automation, has significantly speed up the design process, which is intrinsically iterative. Through the so-called APIs (Application Programming Interfaces), applications can communicate with each other through a shared programming language. This approach allows users to automate manual tasks, typically performed via the software's front-end interface, by transforming them into specific algorithms that directly interact with the software's back-end through its API. Through these algorithms, a sequence of activities can be structured into a flowchart to accomplish a particular goal. In essence, a workflow represents a series of tasks that must be completed to achieve a desired result, with each task being an automated step that processes a set of inputs to produce corresponding outputs.

To date, two main visual programming software are mainly used in the AEC (Architecture Engineering and Construction) industry: Grasshopper (McNeel, 2010) and Dynamo, the first working together with Rhino, while the second working together with Revit. More specifically, Grasshopper is geometry-driven and computes quicker than Dynamo, which is object-driven. Within Grasshopper, algorithms are represented by a series of boxes dragged into the software's canvas and linked to each other by wires (**Figure 1**). By implementing a set of algorithms and manipulating parameters in a dynamic, intuitive interface, the geometric model can be defined, and several analyses can be performed, also leveraging the multitude of plug-ins available for different scopes in building modelling, design, and analysis. The input parameters are represented by sliders which can be easily modified by the user. Otherwise, if fixed parameters are required as input, the user can use a "panel" to write specific data. Instead of modeling geometry directly, users construct a

Funded by the European Union

This project has received funding from the European Union under the Horizon Europe Research & Innovation Programme (Grant Agreement no. 101123467 MultiCare). Views and opinions expressed are however those of author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

flow of data and operations that generate and modify geometry in real time. This enables rapid iteration, responsiveness to changing inputs, and exploration of a wide range of design possibilities—all while maintaining full control over the underlying logic and structure of the model.



A 0.250

B 1.5

0 0.875

L_span_gravity 6

N_span_gravity 3

L_span_seismic 7.1

N_span_seismic 6

H_interstorey 4.3

N_storeys 3

b_column_seismic 0.40

h_column_seismic 0.60

b_beam_seismic 0.40

h_beam_seismic 0.60

b_beam_gravity 0.5

h_beam_gravity 0.7

*Figure 1. Grasshopper user interface with boxes and wires to create an algorithm, on the left, and an example of sliders defining the input parameters of the model, on the right.*

## 2.2 Overview of the framework

Deliverable D6.4 aims at developing a comprehensive parametric framework within the Rhino-Grasshopper environment, also using Python for scripting and automation. The workflow consists of independently developed packages and Python-based modules all inside the same parametric model, which enables to analyze a wide range of solutions just varying the input parameters in a user-friendly manner. The Grasshopper framework is then automatically linked to external Python scripts to perform in-depth multi-performance analyses to evaluate the resilience of the considered solution.

In line with the main objectives of the MULTICARE project, the framework herein presented has been developed for a timber low-damage office/public building, consisting of post-tensioned timber frames as the lateral load-resisting system (LLRS) in one direction and post-tensioned timber walls as the LLRS in the other direction. Timber gravity frames run in the walls' direction to bear the gravity loads coming from the floor. The latter consists of a timber-concrete composite (TCC) flooring system, which can achieve long spans consistently with the use of a post-tensioned system. A representative model of the overall structure is shown in **Figure 2**. The parametric nature of the framework, however, allows to explore a wide range of building configurations by changing the input parameters, listed in the next paragraph, and materials properties.

Funded by the European Union

This project has received funding from the European Union under the Horizon Europe Research & Innovation Programme (Grant Agreement no. 101123467 MultiCare). Views and opinions expressed are however those of author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

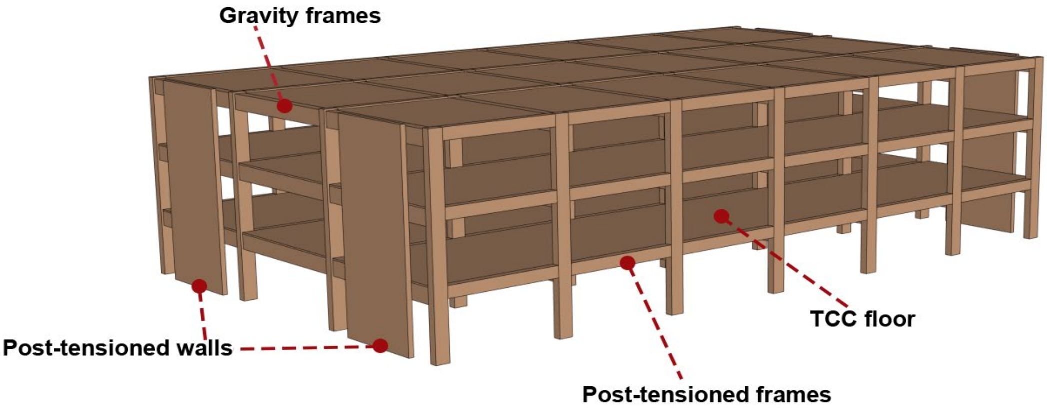


*Figure 2. Three-dimensional parametric model of a timber low-damage building designed by the Grasshopper framework.*

The building considered herein shows two facade systems in the two main directions, specifically, glass curtain walls in the direction of the post-tensioned walls and timber-based cladding walls in the direction of the post-tensioned frames. Similarly to what concerns the structure, also the non-structural elements can be easily modified by changing the input parameters related to facade thickness and its material properties.
An overview of the integrated framework is shown in **Figure 3**, and the main plug-in and software used for the analyses are listed in **Table 3**.

*Table 3. Plug-in and software used for the workflow definition.*

| Plug-in/Software | Tasks |
|---|---|
| Python built in Grasshopper | Structural seismic design |
| Ladybug | Weather definition |
| Honeybee | Energy modelling |
| EnergyPlus | Energy simulations |
| OpenStudio | Energy simulations |
| OpenSeesPy | Structural modelling and seismic assessment |
| PACT (FEMA p-58) | Loss assessment |

Through different algorithms developed within Grasshopper, the geometric model of the building is defined by assigning input parameters represented by sliders, which are better described in the next paragraph. By modifying those sliders, the model is automatically updated. Three main modules can be identified within the Grasshopper script:

- Geometry definition, where the structure and the non-structural components are parametrically modelled
- Structural seismic design, implementing the Displacement-Based Design procedure (Priestley 2002, Priestley et al. 2007) for the structural elements' section design and verification
- Weather definition and energy modelling through Ladybug-Honeybee plug-in (Roudsari and Pak, 2013), gathering the climate properties of the building's location and defining the energy model of the building consisting of parametrically defined thermal zones

Funded by the European Union

This project has received funding from the European Union under the Horizon Europe Research & Innovation Programme (Grant Agreement no. 101123467 MultiCare). Views and opinions expressed are however those of author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

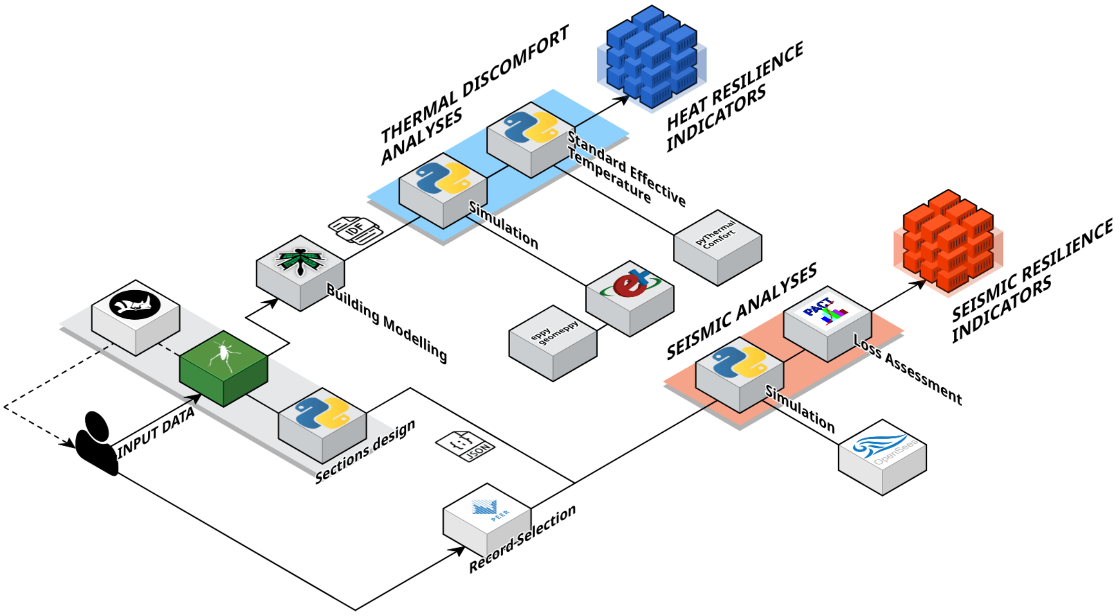


*Figure 3. Overview of the developed framework.*

The aforementioned Grasshopper modules are linked to external Python scripts by automatically exporting their outputs, which then serve as input parameters for the scripts. The developed Python scripts are the following:

- Structural modelling through OpenSees software in Python (McKenna, 2011, Zhu et al. 2018)
- Seismic analyses and earthquake resilience quantification through loss assessment
- Dynamic energy simulations and heat resilience quantification

All these modules and scripts will be explained in detail in the following paragraphs.

## 2.3 Input parameters

The main input parameters defining the building geometric model and influencing both its structural/seismic and energy performance are listed in **Table 4**. Different acceptable ranges with lower and upper bounds have been set for each parameter, determined by designer's choice and material's specifications.

*Table 4. Input parameters defining the building model and influencing both the seismic and energy performance with their lower and upper bounds.*

| Input | Range | Units |
|---|---|---|
| Seismic frame's span length | 5 ÷ 10 | m |
| Number of seismic spans | 2 ÷ 10 | - |
| Gravity frame's span length | 5 ÷ 10 | m |
| Number of gravity spans | 2 ÷ 10 | - |
| Inter-storey height | 2,5 ÷ 5 | m |

Funded by the European Union

This project has received funding from the European Union under the Horizon Europe Research & Innovation Programme (Grant Agreement no. 101123467 MultiCare). Views and opinions expressed are however those of author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

| | | |
|---|---|---|
| Number of storeys | 1 ÷ 10 | - |
| Shear walls thickness | 0,57 ÷ 0,297 | m |
| Shear walls length | 1,5 ÷ 3,5 | m |
| Facade thickness | 0,1 ÷ 0,4 | m |
| Structural materials density | - | kg/m$^3$ |
| Non-structural materials density | - | kg/m$^3$ |

## 2.4 Structural assessment module

Starting from the input parameters listed in the previous paragraph, the structural geometry of the building can be defined by using different Grasshopper algorithms as shown in **Figure 4**. Specifically, the structural grid, necessary to place the columns, is defined by the span number and length of the frames in both directions. Therefore, once the dimensions of the beams and columns cross sections are defined, the structure is automatically created by the algorithms. As far as the lateral shear walls are concern, their position is parametrically defined in order to have at least two shear walls on each side of the building, placed within the two external gravity frame's spans, as already illustrated in **Figure 2**.

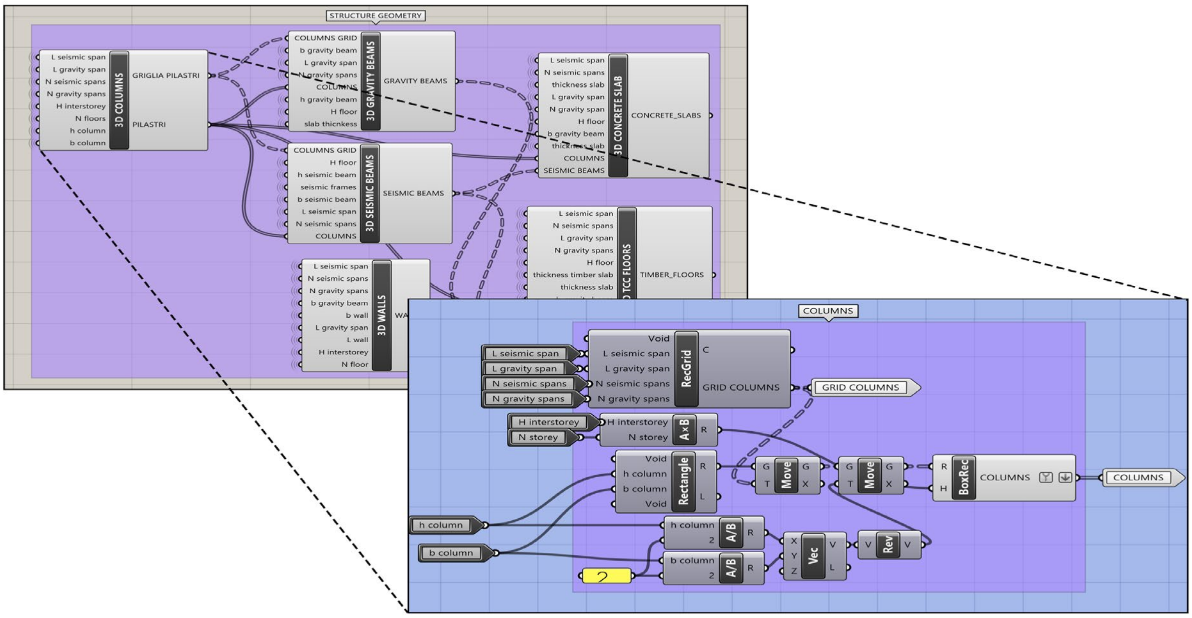


*Figure 4. Grasshopper clusters containing the algorithms defining the structure geometry.*

Once the structure geometry is defined, the building mass and the axial loads on the columns are automatically calculated through a series of algorithms from the weight of the different materials, as well as the variable loads. Then, the seismic design and verification is performed by a series of boxes by implementing the Direct Displacement-Based Design (DDBD) procedure (Priestley 2002, Priestley et al. 2007) through Python scripts. The input parameters related to the seismic structural design, to be considered together with the ones already illustrated in **Table 4**, are listed in **Table 5**, while the main DDBD steps and

Funded by the European Union

This project has received funding from the European Union under the Horizon Europe Research & Innovation Programme (Grant Agreement no. 101123467 MultiCare). Views and opinions expressed are however those of author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

their implementation in Python within Grasshopper are illustrated in **Figure 5a** and **Figure 5b**, respectively.

*Table 5. Input parameters for structural seismic design and verification. The upper and lower bounds are defined by designer's choice, material specifications, as well as codes limitations.*

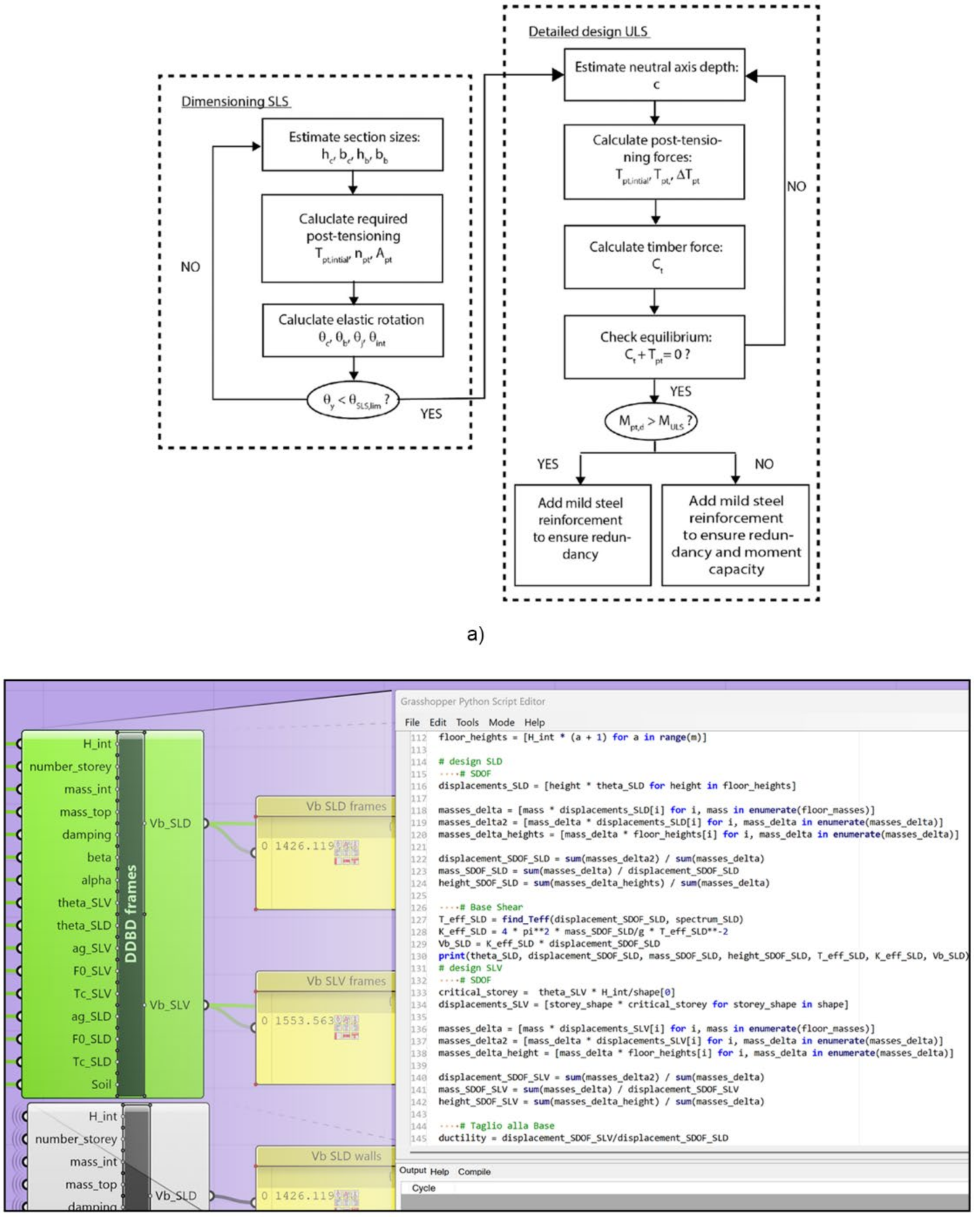
| **Input** | **Range** | **Units** |
|---|---|---|
| Number of seismic frames | 2 ÷ 11 | - |
| Column section width | 0,2 ÷ 0,8 | m |
| Column section height | 0,2 ÷ 1 | m |
| Beam section width | 0,2 ÷ 0,8 | m |
| Beam section height | 0,2 ÷ 1 | m |
| Floor thickness | 0,23 ÷ 0,27 | m |
| Variable loads | 5 | $kN/m^2$ |
| Elastic damping | 3 | % |
| Re-centering ratio for post-tensioned structure | 1,5 | - |
| Parameters defining the elastic acceleration-displacement responde spectra | - | - |
| Beam-column timber connection stiffness ratio | 0,55 ÷ 0,7 | - |
| Column-foundation connection stiffness ratio | 0,55 ÷ 0,7 | - |
| Timber elastic modulus parallel-to-grain | - | MPa |
| Timber elastic modulus perpendicular-to-grain | - | MPa |
| Timber shear modulus | - | MPa |
| Timber compression strength parallel-to-grain | - | MPa |
| Steel modulus of elasticity | - | MPa |
| Steel ultimate strain | - | MPa |
| Steel yielding strength | - | MPa |
| Steel ultimate strength | - | MPa |
| Post-tensioned tendons nominal area | - | MPa |

Through the DDBD methodology, the structure is converted into a single-degree-of-freedom (SDOF) system. By imposing a design drift, the effective (secant) period of the SDOF system can be evaluated from the displacement spectra, which is defined by different input parameters depending on the building location and then reduced by a factor as a function of the structural damping. In this way, the structure is characterized by an effective stiffness depending on the target design drift and on the structural damping, which is a combination of elastic and hysteretic damping. In the framework developed within this deliverable, some adjustments have been made to account for the use of timber in the post-tensioned structure. Indeed, the higher elastic deformability of the components

Funded by the European Union

This project has received funding from the European Union under the Horizon Europe Research & Innovation Programme (Grant Agreement no. 101123467 MultiCare). Views and opinions expressed are however those of author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

before the rocking motion activation should be considered in the case of Pres-Lam. For this technology, the Serviceability Limit State (SLS) generally governs the size of the components and the amount of post-tensioning. Consequently, the structural members are dimensioned to carry the force-demand at the SLS, and then designed in detail and verified at the Ultimate Limit State (ULS). Therefore, the output of the procedure implemented in the model are represented by the design base shears at SLS and ULS, which are distributed along the structure for both the seismic-resistant frames and walls. From the resulting forces acting on the structural elements it is then possible to design in detail the connections reinforcement (i.e., internal post-tensioned tendons and external dissipaters) and verify them at ULS.



a)

```python
floor_heights = [H_int * (a + 1) for a in range(m)]

# design SLD
    # SDOF
displacements_SLD = [height * theta_SLD for height in floor_heights]

masses_delta = [mass * displacements_SLD[i] for i, mass in enumerate(floor_masses)]
masses_delta2 = [mass_delta * displacements_SLD[i] for i, mass_delta in enumerate(masses_delta)]
masses_delta_heights = [mass_delta * floor_heights[i] for i, mass_delta in enumerate(masses_delta)]

displacement_SDOF_SLD = sum(masses_delta2) / sum(masses_delta)
mass_SDOF_SLD = sum(masses_delta) / displacement_SDOF_SLD
height_SDOF_SLD = sum(masses_delta_heights) / sum(masses_delta)

    # Base Shear
T_eff_SLD = find_Teff(displacement_SDOF_SLD, spectrum_SLD)
K_eff_SLD = 4 * pi**2 * mass_SDOF_SLD/g * T_eff_SLD**-2
Vb_SLD = K_eff_SLD * displacement_SDOF_SLD
print(theta_SLD, displacement_SDOF_SLD, mass_SDOF_SLD, height_SDOF_SLD, T_eff_SLD, K_eff_SLD, Vb_SLD)
# design SLV
    # SDOF
critical_storey =  theta_SLV * H_int/shape[0]
displacements_SLV = [storey_shape * critical_storey for storey_shape in shape]

masses_delta = [mass * displacements_SLV[i] for i, mass in enumerate(floor_masses)]
masses_delta2 = [mass_delta * displacements_SLV[i] for i, mass_delta in enumerate(masses_delta)]
masses_delta_height = [mass_delta * floor_heights[i] for i, mass_delta in enumerate(masses_delta)]

displacement_SDOF_SLV = sum(masses_delta2) / sum(masses_delta)
mass_SDOF_SLV = sum(masses_delta) / displacement_SDOF_SLV
height_SDOF_SLV = sum(masses_delta_height) / sum(masses_delta)

    # Taglio alla Base
ductility = displacement_SDOF_SLV/displacement_SDOF_SLD
```

b)

*Figure 5. a) Flow chart of the modified Displacement-Based design procedure for post-tensioned timber structures; b) Grasshopper box containing the Python script for the implementation of the design procedure, returning the base shear of the structure as output.*

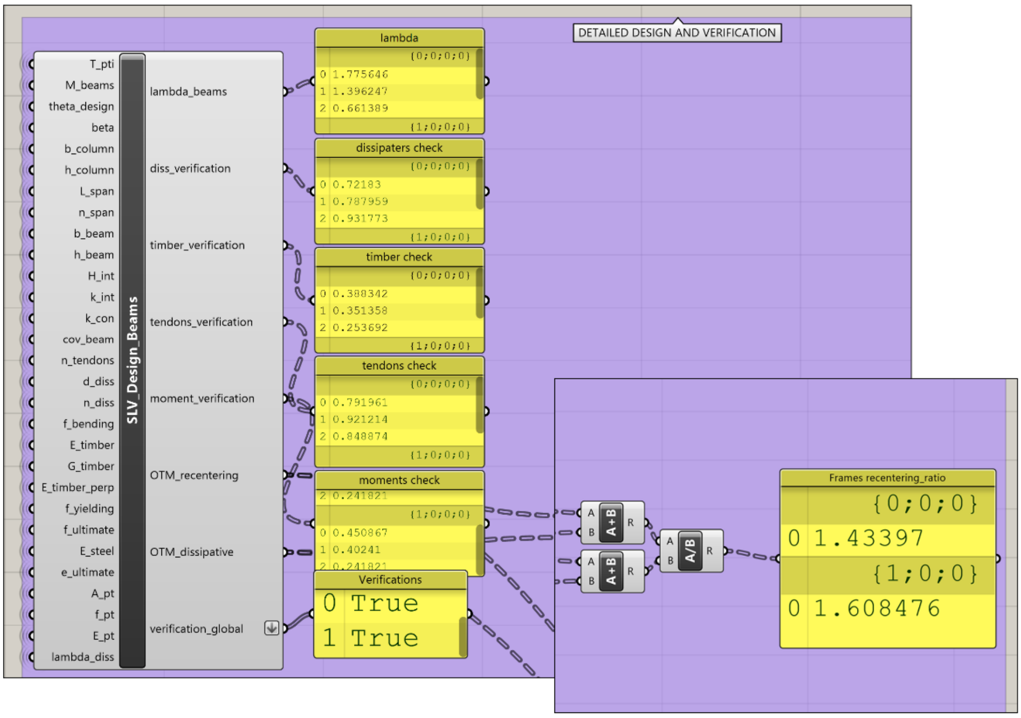
Funded by the European Union

This project has received funding from the European Union under the Horizon Europe Research & Innovation Programme (Grant Agreement no. 101123467 MultiCare). Views and opinions expressed are however those of author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

Therefore, the Grasshopper workflow related to the seismic design is mainly characterized by the following boxes, which contain different Python scripts:

- The box implementing the DDBD procedure and returning the design base shear of the structures at both SLS and ULS
- The box distributing the base shear along the structure following an equilibrium method, at both SLS and ULS
- The box where the rotation at SLS is verified with respect to the code-imposed limit. This box also returns the initial post-tensioning force for the internal unbonded tendons
- The box performing a quick design of the structural sections (i.e., beams, columns, walls) and returning the number and diameters of the external dissipaters and the number of internal post-tensioned tendons
- The box where verification at ULS are carried out for both beams, columns and walls

The outcome of the seismic design procedure is therefore represented by an overall Boolean value (**Figure 6**) saying if the structural elements are verified or if changes are necessary in the input parameters to design a code-compliant structure. Moreover, the value of the re-centering ratio is returned (i.e., the ratio between the re-centering capabilities and the energy dissipation of the structure, which might be around a value of 1.5 in order to guarantee the absence of residual displacement after an earthquake).
By implementing an automated loop, or by using an optimization algorithm within the same framework, the input parameters could be automatically changed by iteration when the Boolean value related to the verifications is False.



```
lambda
{0;0;0;0}
0 1.775646
1 1.396247
2 0.661389
{1;0;0;0}

dissipaters check
{0;0;0;0}
0 0.72183
1 0.787959
2 0.931773
{1;0;0;0}

timber check
{0;0;0;0}
0 0.388342
1 0.351358
2 0.253692
{1;0;0;0}

tendons check
{0;0;0;0}
0 0.791961
1 0.921214
2 0.848874
{1;0;0;0}

moments check
2 0.241821
{1;0;0;0}
0 0.450867
1 0.40241
2 0.241821

Verifications
0 True
1 True

Frames recentering_ratio
{0;0;0}
0 1.43397
{1;0;0}
0 1.608476
```

*Figure 6. Final Grasshopper seismic design box returning the verifications outcome of the structural systems.*

Funded by the European Union

This project has received funding from the European Union under the Horizon Europe Research & Innovation Programme (Grant Agreement no. 101123467 MultiCare). Views and opinions expressed are however those of author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

The main output resulting from the seismic design as well as the input parameters for materials and seismic zone definitions are then automatically exported as different json files within the external folder containing the python script for the seismic modelling, assessment and seismic resilience quantification. Specifically, four json file are necessary for this purpose (**Figure 7**):

- Json file containing the frame and walls geometry characteristics and the elements sections properties
- Json file containing the timber properties
- Json file containing the steel properties for the external dissipaters
- Json file containing the steel properties for the internal post-tensioning tendons

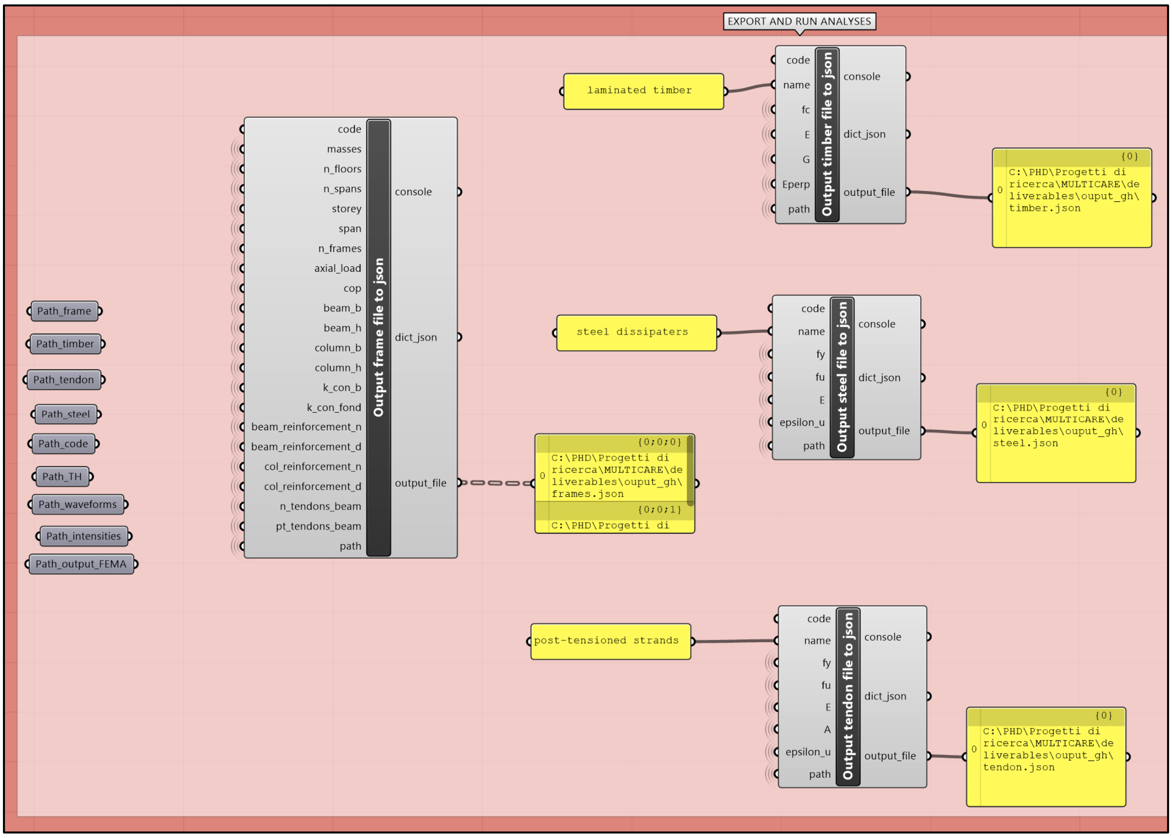
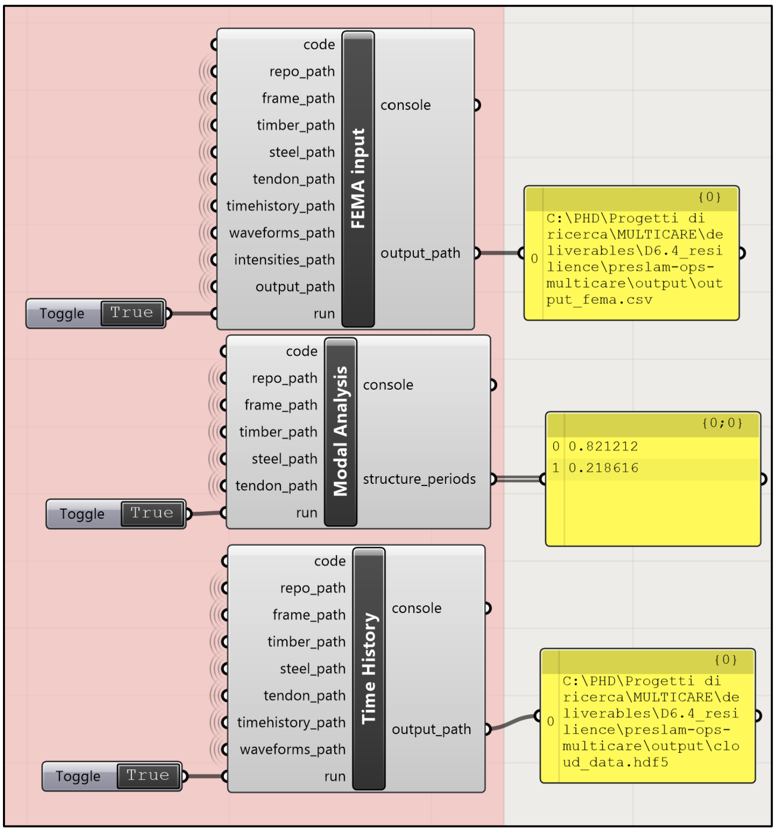


*Figure 7. Grasshopper boxes to export the .json file needed as input for the Python external scripts for seismic analyses.*

Once the json file representing the input parameters for the structural assessment are exported, seismic analyses are carried out to more accurately evaluate its performance. These analyses are performed using external software developed as part of the Multicare Project, which is openly accessible (Matteoni et al. 2024) and available also on the Multicare Github repository (https://github.com/multicareConsortium). The tool has been adapted to be invoked directly through the GH plugin. Specifically, through Grasshopper Boolean Toggle, the user can run seismic simulations and visualize the main results directly from the GH model in a user-friendly manner (**Figure 8**).

Funded by the European Union

This project has received funding from the European Union under the Horizon Europe Research & Innovation Programme (Grant Agreement no. 101123467 MultiCare). Views and opinions expressed are however those of author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.



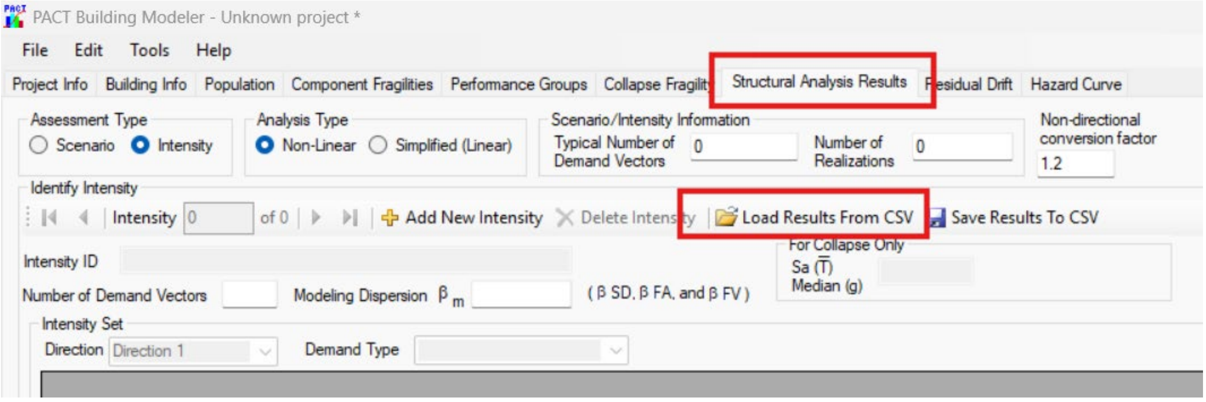
```
{0}
0 C:\PHD\Progetti di ricerca\MULTICARE\deliverables\D6.4_resilience\preslam-ops-multicare\output\output_fema.csv
```

```
{0;0}
0 0.821212
1 0.218616
```

```
{0}
0 C:\PHD\Progetti di ricerca\MULTICARE\deliverables\D6.4_resilience\preslam-ops-multicare\output\cloud_data.hdf5
```

*Figure 8. Grasshopper boxes to invoke the external scripts and run the structural analyses directly within Grasshopper, also visualizing the main output or the file path containing the output.*

The primary function of the structural simulation tool is to perform nonlinear time-history analyses, providing engineering demand parameters (EDPs) necessary for component-based loss assessments, in line with FEMA P-58 methodology. Specifically, the tool generates the input files required for structural characterization, using a format compatible with FEMA's PACT software. These results can be imported directly into PACT via the "Structural Analysis Results" tab, using the "Load Results From CSV" option (**Figure 9**).



Assessment Type ○ Scenario ● Intensity
Analysis Type ● Non-Linear ○ Simplified (Linear)
Scenario/Intensity Information
Typical Number of Demand Vectors 0
Number of Realizations 0
Non-directional conversion factor 1.2
Intensity 0 of 0
Intensity ID
Number of Demand Vectors
Modeling Dispersion $\beta_m$ ( β SD, β FA, and β FV )
For Collapse Only
Sa (T)
Median (g)
Direction Direction 1
Demand Type

*Figure 9. Procedure to import the results into PACT tool for loss assessment.*

Further details about the ouput resulting from the analyses will be discussed in the next chapter.

Funded by the European Union

This project has received funding from the European Union under the Horizon Europe Research & Innovation Programme (Grant Agreement no. 101123467 MultiCare). Views and opinions expressed are however those of author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

## 2.5 Thermal discomfort analyses module

To configure a parametric building model compatible with thermal discomfort simulations, a custom modeling logic was developed using the Honeybee plugin in Grasshopper. The model was exported as an IDF file for simulation in EnergyPlus. The Dragonfly plugin was also used alongside Honeybee to facilitate the structured creation of multi-zone energy models.

In EnergyPlus, all building surfaces are modeled as thin, zero-thickness planes. According to EnergyPlus documentation, exterior surfaces should use outside dimensions, while interior surfaces should use centerline dimensions. Based on this guidance, the parametric model was configured as follows (see **Figure 10**):

- Exterior walls: The outermost lines of the structural columns were offset by the defined façade thickness to form the exterior wall boundaries.
- Interior walls: X and Y grid lines from the columns were extended to the exterior walls, defining the interior partitions.
- Thermal zones: Walls, floors, and ceilings were assigned to enclose each space as an individual thermal zone.

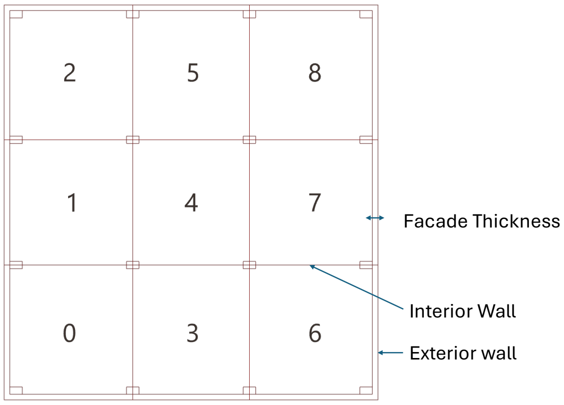


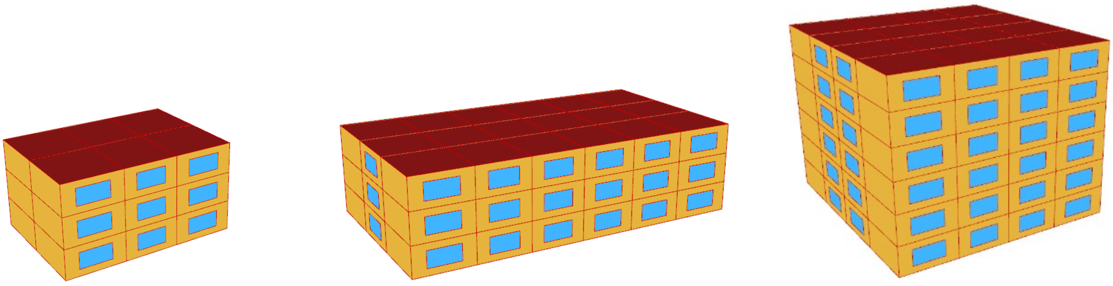
*Figure 10. Surface boundary and thermal zone definitions based on Energyplus modeling convention.*

The building features post-tensioned structural walls along its side edges. Regardless of the number of bays, two of these walls always appear at the outermost sides. As shown in **Figure 11**, thermal zones were assigned different window-to-wall ratios (WWR) based on the parametric configuration. For example, in a 2×3 bay configuration, only the side edge walls are solid. As the bay count increases (e.g., to 3×6), windows are added to the middle side walls.

To accommodate these variations, thermal zones were classified into the following types:

- Exterior-corner zones: Windows placed only on the north or south façade.
- Exterior-perimeter zones: Windows placed on exterior-facing walls.
- Interior zones: No windows assigned.

This project has received funding from the European Union under the Horizon Europe Research & Innovation Programme (Grant Agreement no. 101123467 MultiCare). Views and opinions expressed are however those of author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.



2x3x3 3x6x3 4x4x6

*Figure 11. Thermal zone classification and window-to-wall ratio configuration relative to structural wall and bay layout.*

Internal loads and occupancy data were sourced from the Medium Office program in the Honeybee database, which provides default values of 0.071 people/m² for occupancy, 6.7 W/m² for lighting, and 7.5 W/m² for equipment. The building was modeled as unconditioned, with no active HVAC systems. Natural ventilation was implemented using operable windows, which opened when indoor air temperature exceeded 20°C and outdoor temperature was above 15°C; otherwise, the windows remained closed.

After completing the modelling in Grasshopper, the building geometry and all related information are inserted in the python script in the form of an input data file (.idf). The building data are then used to perform the energy simulation connecting to the EnergyPlus (National Renewable Energy Laboratory, 2017) simulation engine using Eppy ("Eppy", 2013) and GeomEppy (Bull J. et al., 2016) python libraries (**Figure 12**).

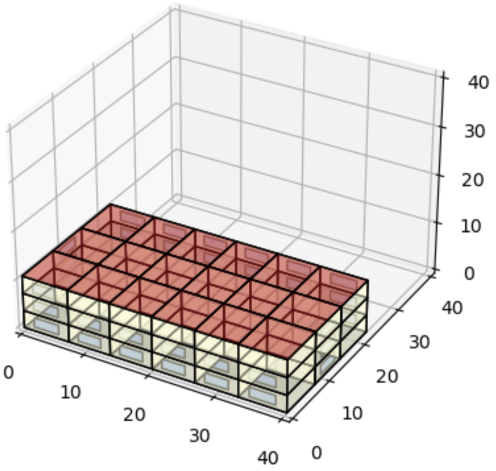


*Figure 12. Visualization of the model after inserting the building geometry in python using Eppy & Geomeppy python libraries.*

## 2.6 Output: Resilience indicators

### Heat resilience

The heat resilience indicators are computed based on the simulation output, which includes the following variables: (i) Zone Mean Radiant Temperature, (ii) Zone Mean Air

Funded by the European Union

This project has received funding from the European Union under the Horizon Europe Research & Innovation Programme (Grant Agreement no. 101123467 MultiCare). Views and opinions expressed are however those of author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

temperature and (iii) Zone Air Relative Humidity. All other output variables are removed from the idf file to reduce the total computational time needed. Particularly, in this case, given that the thermal resilience is evaluated considering a natural ventilation scenario, there are no output variables related to energy demand and/or consumption.

The results are the used to calculate the Standard Effective Temperature (SET) using pythermalcomfort python library (Tartarini & Schiavon, 2020). Specifically, the cumulative SET for each zone along the duration of the heatwave are used. The results are shown in **Figure 13** where the lines are color-coded based on zone orientation and building floor. In this example, the worst performing zones are the east and south of the top floor while the best performing zones are the zones in the north side of the ground floor.

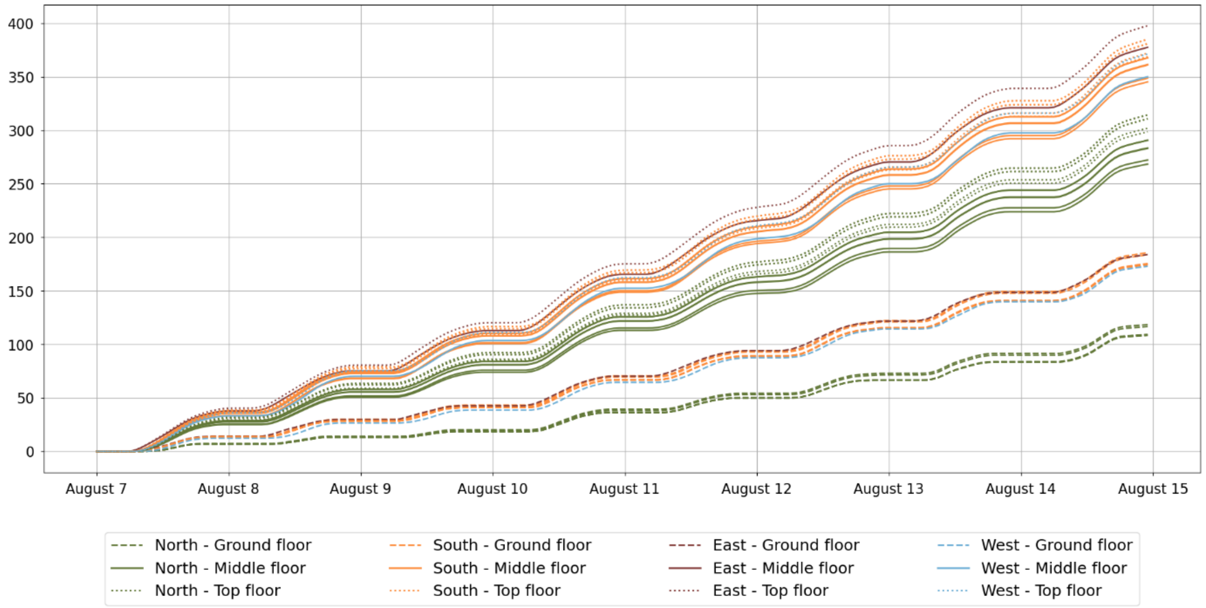


*Figure 13. Cumulative SET hours per thermal zone along the duration of the heatwave.*

Finally, the cumulative SET values are normalized to compute the heat resilience indicators, following the approach described in Deliverable D6.1. The two main indicators that are computed in this case are:

a. Normalized **response** indicator. Here the cumulative unmet SET zone hours during the heatwave are normalized with a reference value that corresponds to a SET temperature of 35 degrees for the same time period.
b. Normalized **recovery** indicator. This indicator is defined as the number of recovery hours normalized with a reference value corresponding to a duration of 14 days. The recovery hours are defined as the number of consecutive hours after the heatwave for which the Zone Mean Air temperature is maintained above 30 degrees.

Both results are computed individually for each thermal zone so as to assess individually the heat resilience for all different floors and orientations.

### Seismic Resilience

The plugin is designed to reduce the effort associated with structural characterization, typically the most demanding step in the loss assessment process. PACT supports the computation of several key performance indicators (KPIs) identified in Deliverable D6.1 (also listed below in **Table 6**), including Res3, Res4, Rec1, and Rec3. These KPIs can be evaluated either in a time-based format (e.g., normalized by annual losses) or specific hazard scenarios

Funded by the European Union

This project has received funding from the European Union under the Horizon Europe Research & Innovation Programme (Grant Agreement no. 101123467 MultiCare). Views and opinions expressed are however those of author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

(e.g., based on a given return period). Additionally, the structural simulation provides the residual drift ratio, Res2, which is a required input in the FEMA P-58 methodology.

*Table 6. Seismic Resilience Indicators considered in the present framework as per Deliverable D6.1.*

| **Resilience phase** | **Index** | **Domain** | **Indicator name** | **Normalization** |
|---|---|---|---|---|
| | | | Mean Annual Frequency of Exceedance of a Limit/Damage State (1/years) | It is normalized by setting a desirable target for the damage state. (Threshold) |
| Response | $I_{Res2}$ | Physical | Residual Drift Ratio | It is normalized by setting a desirable target for the residual drift ratio. (Threshold) |
| Response | $I_{Res3}$ | Social | Casualties | The number of casualties can be normalized by the number building occupants. |
| Response | $I_{Res4}$ | Economic | Expected Annual Loss (cost/m2/year or % Reconstruction Cost/year) | The building's cost is normalized as a percentage of the total building replacement cost. |
| Recovery | $I_{Rec1}$ | Economic | Downtime (months) | Downtime is normalized in relation to the life cycle of the structure. |
| Recovery | $I_{Rec3}$ | Environmental | Annualized Carbon Output (CO2gwp_eq/sqm/y) | The carbon output can be normalized considering the total building carbon footprint |

Lastly, the plugin includes an additional module to perform structural simulations for fragility assessment using the CLOUD methodology (Jalayer et al. 2017). Fragility curves are derived from the outputs of the structural simulation (Matteoni et al. 2024) and can be integrated to compute KPI Res1—i.e., the mean annual frequency of exceedance (MAFE) for a specified limit state—following the methodology proposed by Iervolino et al. (2015) and applied by Formichetti et al. (2023).

It is worth noting that, some additional information and additional input values need to be sourced outside the grasshopper environment itself to perform the whole calculation of the KPIs, such as record selection for the definition of the waveforms, or the definition of the inventory of components for the building.

# 3. Conclusions

Deliverable 6.4 focuses on the development of a user-friendly, open-source workflow—implemented as a plug-in for the CAD software Rhinoceros—to support holistic building design. This workflow assists designers and decision-makers in identifying optimal building solutions during the early design stages by integrating both seismic and energy performance considerations.

Funded by the European Union

This project has received funding from the European Union under the Horizon Europe Research & Innovation Programme (Grant Agreement no. 101123467 MultiCare). Views and opinions expressed are however those of author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

The framework is built within Grasshopper and leverages various tools and plug-ins, as well as native Python scripting, to enable advanced parametric modeling. To ensure a thorough evaluation of seismic and energy performance, external Python-based scripts were developed and seamlessly integrated into the Grasshopper environment. This integration allows users to execute complex analyses directly within the main interface, avoiding the need to switch between multiple software platforms.

The framework has been applied to a low-damage timber building, aligning with the objectives of the MULTICARE project. However, its parametric and flexible structure enables its application to a wide variety of building types, materials, and configurations, while accounting for the seismic and climatic conditions of the location.

The outputs of the workflow are expressed through Resilience Indicators, as defined in Deliverable D6.1, encompassing both seismic and thermal performance. As such, Deliverable 6.4 offers a valuable tool for automating and streamlining the design of resilient buildings, in line with the goals of Work Package 6, through user-friendly processes and visualization tools. In the future, the framework could also be extended to support the design of integrated, non-invasive rehabilitation strategies for existing buildings.

# 4. References

Bull J, et al. (2016). GeomEppy [Online]. Available: https://github.com/jamiebull1/geomeppy. [Accessed 12 03 2025]. "Eppy," 2013. [Online]. Available: https://github.com/santoshphilip/eppy. [Accessed 12 03 2025].

Federal Emergency Management Agency. (2012) Seismic performance assessment of buildings, volume 1—Methodology. Technical report FEMA-P-58-1, Department of Homeland Security, Washington, DC.

Formichetti, G., Matteoni, M., Pampanin, S. (2023) Development of a holistic parametric framework for multi-performance life-cycle evaluation of post-tensioned timber buildings. 9th International Conference on Computational Methods n Structural Dynamics and Earthquake Engineering, Athens, Greece.

Iervolino, I., Spillatura, A., & Bazzurro, P. (2018). Seismic Reliability of Code-Conforming Italian Buildings. Journal of Earthquake Engineering, 22(sup2), 5–27.

Jalayer, F., Ebrahimian, H., Miano, A., Manfredi, G., Sezen, H. (2017) Analytical fragility assessment using unscaled ground motion records. Earthquake Engineering & Structural Dynamics; 46(15): 2639-2663.

Matteoni, M., Ciurlanti, J., Bianchi, S., Pampanin, S. (2024). Fragility functions for low-damage post-tensioned timber frames. Earthquake Engineering & Structural Dynamics; 53(15): 4741–4762.

McKenna, F. (2011). OpenSees: A Framework for Earthquake Engineering Simulation. Computing in Science & Engineering, 13, 58–66.

McNeel, R. (2010). Grasshopper - Generative Modeling with Rhino. McNeel North America, Seattle, Washington.

Funded by the European Union

This project has received funding from the European Union under the Horizon Europe Research & Innovation Programme (Grant Agreement no. 101123467 MultiCare). Views and opinions expressed are however those of author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

National Renewable Energy Laboratory (NREL), *EnergyPlus™,* 2017.

Priestley, M.N. (2002). Direct displacement-based design of precast/prestressed concrete buildings. PCI Journal, 47(6), 66-79.

Priestley, M.N., Calvi, G.M., Kowalsky, M.J. (2007). Displacement-Based Seismic Design of Structures. Earthquake Spectra, 24(2), 555–557.

Roudsari, M.S., Pak, M. (2013). Ladybug: a parametric environmental plugin for grasshopper to help designers create an environmentally-conscious design. In Proceedings of the 13th IBPSA Conference, Lyon, France.

Tartarini, F., Schiavon, S., 2020. pythermalcomfort: A Python package for thermal comfort research. SoftwareX 12, 100578. https://doi.org/10.1016/j.softx.2020.100578

Zhu, M., McKenna, F., Scott, M.H. (2018). OpenSeesPy: Python library for the OpenSees finite element framework. SoftwareX, 7, 6–11.

Funded by
the European Union

This project has received funding from the European Union under the Horizon Europe Research & Innovation Programme (Grant Agreement no. 101123467 MultiCare). Views and opinions expressed are however those of author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.