



AUTOMATING THE CONVERSION OF REVIT MODELS TO OPENSEES USING REVIT API: AN INITIAL CASE STUDY

Burcu Güldür Erkal¹, Merve Bayraktar², and Tobias Maile¹

¹Technical University of Applied Sciences Augsburg, Augsburg, Germany

²Hacettepe University, Ankara, Turkey

Abstract

This study proposes a workflow for converting Revit models into OpenSees input files using the Revit API. The goal is to automate the transfer of structural data, reducing manual errors and increasing efficiency in BIM-to-FEM integration. A case study of a two-story masonry building demonstrates the approach. The method enables direct extraction of geometry and properties from Revit and structures them into OpenSees-compatible scripts for structural analysis.

Introduction

Efficient structural analysis benefits from the integration of building information model (BIM) and finite element analyses (FEA) tools. Revit (Autodesk, Inc. (2024)) supports detailed 3D modeling, while OpenSees (McKenna, F. et al. (2010)) enables simulation of structural responses under loads, particularly seismic events. However, interoperability challenges between these platforms often require manual data handling, which is time-consuming and error-prone. This study introduces an automated process using Revit's API to bridge this gap, enabling direct translation from Revit to OpenSees.

Methodology

The proposed workflow involves:

- Identifying structural elements (e.g., beams, columns, walls) in Revit.
- Extracting key properties like coordinates, dimensions, and materials via the Revit API.
- Structuring this data into OpenSees scripts to define nodes, elements, boundary conditions, and loads.

A case study Revit model of a two-story masonry building was developed. Using Python tools (e.g., RevitPythonShell (Donker, E. (2013))), the script extracts and organizes structural data for OpenSees. The model includes materials, cross-sectional properties, and connectivity information.

Revit API usage

Key API functions include:

- FilteredElementCollector for selecting elements.
- LookupParameter for retrieving geometry and material data.
- Transactions for runtime parameter modifications.

Automating this process minimizes human error and ensures consistent data transfer.

OpenSees model generation

Once extracted, data is formatted into an OpenSees input script. Nodes are defined using Revit-derived coordinates. Beam and column elements are represented using elasticBeamColumn, and appropriate load cases and boundary conditions are applied. Material properties such as elastic modulus and density are derived from Revit definitions. The resulting OpenSees model replicates the original structural layout. Figure 1 shows the workflow diagram illustrating the process from Revit to OpenSees.

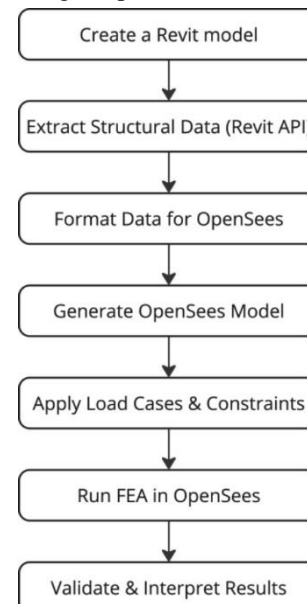


Figure 1: A workflow diagram illustrating the process from Revit to OpenSees

Challenges and solutions

Discrepancies in data structures between Revit and OpenSees pose challenges. For example:

- Complex geometries are simplified in OpenSees, sometimes losing detail.
- Not all Revit elements (e.g., slabs, braces) directly map to analytical types.
- Revit materials require conversion for OpenSees compatibility.

To mitigate this, preprocessing scripts and standardized mappings are recommended. Validation scripts are also used to check for connectivity errors and missing data before export.

Discussion

This automated workflow facilitates efficient model translation, reducing the need for manual scripting. However, care must be taken to align the structural logic between platforms. Geometry in Revit must be filtered and mapped to analytical equivalents in OpenSees to ensure accurate simulations.

Conclusion

The proposed API-based workflow significantly improves the transition from BIM to FEA by enabling the automated generation of OpenSees models from Revit. While challenges remain in aligning analytical representations, the approach offers a scalable and consistent method for integrating design and analysis processes in structural engineering.

Acknowledgments

This work was conducted at the Technical University of Applied Sciences Augsburg and is based on insights and experiences gained through the Master's course titled *CAX – Computer-Aided Engineering and Design*.

References

- McKenna, F., Fenves, G.L. & Scott, M.H. (2010) OpenSees: Open System for Earthquake Engineering Simulation. Pacific Earthquake Engineering Research Center, University of California, Berkeley. Available from: <https://opensees.berkeley.edu>
- Autodesk, Inc. (2024) Autodesk Revit API Developer's Guide. Available from: <https://help.autodesk.com/view/RVT/2024/ENU/>
- Donker, E. (2013) RevitPythonShell: Scripting Autodesk Revit with IronPython. Available from: <https://github.com/architecture-building-systems/revitpythonshell>