STRUCTURAL ENGINEERING TRANSFORMATION WITH ETABS API DESIGN AUTOMATION

GANGADHARI ANIL KUMAR[1], Dr. M. ANITHA[2], Dr. B. SHARATH CHANDRA[3]
[1]M.Tech Student, [2]Associate Professor, [3]HOD & Assistant Professor
Department of Civil Engineering, Ellenki College of Engineering and Technology, Patelguda, Hyderabad, Telangana.

Abstract - Due to its many advantages, including increased uniformity in design, less room for human mistake, and streamlined modeling efficiency, automation has quickly become an integral part of contemporary structural engineering. An automated system for structural modeling and engineering documentation was created and validated in this work utilizing the ETABS Open Application Programming Interface (API). The suggested application was built in C# with the help of the Microsoft.NET Framework. It automates the production of structural models and the compilation of engineering reports by integrating the ETABS COM automation with the Microsoft Word COM automation. Input characteristics such as structural grids, columns, roof rafters, intermediate members, and bracing systems are used by the created program to automatically build warehouse structural geometry. Simultaneously, the software produces standardized engineering documentation. We compared the built system to ETABS models that were manually created and used three warehouse layouts with varying dimensions to verify it. Verification of geometry, connection of members, correctness of coordinates, modeling time, and documentation quality were all part of the validation process. While cutting modeling time by about 97-99.6 percent, the automated models showed full geometric agreement with reference models made by hand. Additionally, without any human involvement, the produced engineering reports had consistent formatting. Using ETABS API-based automation, the research shows that repeated structural modelling applications may be reliably and efficiently handled, leading to significant increases in engineering productivity, modeling accuracy, and documentation quality.

Important Terms - The following terms are used to describe various aspects of software development: ETABS API, structural automation, building information modeling (BIM), C#, the Microsoft.NET framework, COM automation, structural modeling, engineering documentation, warehouse structures, and software.

I. INTRODUCTION

Computational techniques for structural analysis, modeling, and design have become more popular due to the rising complexity of contemporary building projects. Building Information Modeling (BIM), finite element analysis software, and bespoke engineering apps are becoming more important tools for structural engineering companies looking to boost efficiency without sacrificing accuracy. The extensive features of ETABS for structural engineering software have led to its widespread use as a platform for modeling, assessing, and designing building structures. Despite ETABS's effective graphical modeling environment, manual processes are still heavily relied upon for the production of structural models for recurring tasks. Engineering documentation, structural grid definitions, joints, frame member generation, section property assignment, and so on are all tasks that engineers must do regularly. These processes need a lot of engineering work for projects that include repeating warehouse or industrial buildings, and they can lead to coordination mistakes, missing structural parts, and inconsistent paperwork. Therefore, one of the main goals of structural engineering automation is to decrease the amount of repetitious manual labor.

To get around these restrictions, the ETABS analysis engine may be directly accessed by third-party software programs using the ETABS Open Application Programming Interface (API). As an alternative to manual graphical operations, the API allows for the automated generation of structural models utilizing engineering techniques. The API provides a full digital workflow from structural modeling to report creation when integrated with Microsoft Office COM Automation. It also allows automatic compilation of engineering documentation immediately after model development. With the help of the C# programming language and the Microsoft. NET Framework, this research builds a desktop application that can automatically generate warehouse structural models and engineering documentation. The program communicates with ETABS via its COM-based API, uses user-defined geometric parameters to automatically produce structural components, and uses Microsoft Word automation to make structured engineering reports. We check the built software for errors by comparing it to ETABS models that were created by hand. We also look at how fast it runs, how reliable it is, and how good the documentation is. This research shows that engineering automation is useful and shows how the ETABS API may help make structural modeling more efficient in repeated engineering tasks.

II. REVIEW OF LITERATURE

Computational technologies have greatly enhanced the efficiency and reliability of structural analysis and design in structural engineering. Engineers may now create more accurate and computationally efficient complicated analytical models because to the shift from traditional manual drawing to computer-aided structural modeling. Programs sold commercially include ETABS, SAP2000, and STAAD.Because of its compatibility with international design standards and capacity to do sophisticated finite element analysis, Pro have become the go-to tools for structural analysis. Even with all these improvements, structural model production for recurring tasks is still heavily reliant on manual modeling, which is both laborious and prone to human mistake. By creating a unified digital environment for geometric modeling, structural analysis, and project documentation, the fast-growing Building Information Modeling (BIM) has further revolutionized the structural engineering business. Improved project coordination, less design discrepancies, and more collaboration are all outcomes of BIM-based processes for the engineering, architectural, and construction industries. Application Programming Interfaces (APIs) are becoming more common in modern engineering software. This opens the door for specialized apps to automate mundane engineering tasks and enhance the features of pre-existing commercial software. By use of the Component Object Model (COM) technology, the ETABS Open Application Programming Interface (API) provides direct access to the core modeling operations of ETABS. Through the API, third-party apps may automate structural geometry generation, material property definition, section property assignment, frame object creation, structural analysis execution, and analytical result extraction. These features have made the ETABS API a desirable platform for creating bespoke engineering apps that lessen the burden of manual modeling without sacrificing consistency or accuracy.

Recent studies have shown that engineering software development is best accomplished in an environment that makes use of programming languages like C# and the Microsoft.NET Framework. Their object-oriented design allows for reusable software components, modular program design, and smooth interface with engineering programs that run on Windows. Complete automation of structural modeling and engineering documentation is made possible by COM compatibility, which allows C# programs to connect quickly with ETABS and Microsoft Office products. Workflow efficiency is greatly enhanced and repetitive engineering jobs are drastically reduced by this integration. For structural design projects, engineering documentation is a must. Even once structural modeling is finished, there is always a lot of human labor required to prepare design documentation, calculation summaries, member schedules, and technical reports. With the help of Microsoft Word COM Automation, programs may automatically create headers, tables, paragraphs, and engineering data within Microsoft Word, resulting in properly designed engineering reports. Automated documentation does more than only boost efficiency; it also guarantees uniform report layout and cuts down on editing mistakes. Thorough validation processes are crucial for ensuring the dependability of structural modeling systems that use automated methods. Research has shown that in order to ensure the correctness of automated models, it is necessary to compare them to reference models that were created manually. This comparison should include aspects such as structural topology, member connectivity, geometric accuracy, and coordinate generation. Furthermore, comparisons of modeling times, testing for robustness, and error analysis are typical ways to evaluate software performance. These validation techniques ensure that automation algorithms may be used in real-world engineering projects without sacrificing structural dependability or the correctness of the models. Despite the significant advancements made in structural engineering automation, the majority of the existing research focuses on specific areas such as API programming or structural analysis. The topic of combining technical documentation with automated structural model development in one software application has received little attention in the literature. On top of that, there has been no effort to systematically validate automated warehouse models using quantitative performance measures. In order to fill these gaps, this research developed an automated framework that is based on the ETABS API. This framework can generate engineering reports and full warehouse structural models with high geometric correctness and efficiency.

III. STUDY DESIGN

A. The Design of an Automated System

Built in a Microsoft Visual Studio environment, the two software components that make up the automation framework are built using the C# programming language. One part of ETABS, called CreateWarehouseGeometryInETABS, uses the ETABS Open Application Programming Interface (OAPI) to generate the steel warehouse structural model automatically. The second part, named MicrosoftWordsExample01, uses the Microsoft Office Word Interop API to automate structural documentation procedures. Stage one of the process involves gathering user-defined geometric parameters. Stage two involves automatically generating structural geometry inside ETABS. Stage three involves validating the resulting model objects and connections. Lastly, stage four involves

automatically preparing engineering documentation. The architecture is designed to work with common ETABS modeling techniques while minimizing manual intervention. The ETABS automation module is able to connect with an active ETABS instance using the OAPI interface, which is based on COM. via the cOAPI interface, the program has access to the ETABS object model, and via the cSapModel interface, it controls the structural model. You may access the functions needed to create grids, stories, joints, and frame objects using these interfaces.

B. An Analytical Depiction of Warehouse Layout

This research focuses on a two-story steel frame warehouse with a rectangular design. A small number of input parameters reflect the geometry, rather than specifying each structural part separately. The application that has been designed may take the following main variables:

Parameter	Description
Number of spans in X direction	Number of longitudinal bays
Number of spans in Y direction	Number of transverse bays
Spacing X	Longitudinal bay spacing
Spacing Y	Transverse bay spacing
Bottom story height	Height of lower structural level
Typical story height	Height of upper structural level

All of the main geometrical features of the warehouse structure are defined by these characteristics. Mathematical equations based on grid indices and spacing values automatically determine the coordinate coordinates of frame parts and joints. The model that is produced includes:

- Columns for the ground and higher stories
- Rivers for the door frame
- Columns that connect the front and back of the building
- Members for bracing the roof
- Members that brace the wall longitudinally and transversely

Because of this parametric form, the same algorithm may produce warehouse models of varying sizes without requiring any changes to the code.

C. Generating Models via ETABS and OAPI

The first step in creating geometry is connecting to ETABS. Using the COM running-object protocol, the program first tries to connect to an existing ETABS session. A new instance of ETABS is automatically constructed and shown to the user if a current session cannot be maintained. The cSapModel object is used to initialize a new structural model when communication is established. After that, the user-provided parameters are matched with a consistent metric system that uses meters and kilonewtons to provide the working units. The span and spacing values you choose will determine the structural grid system that is created. Both the X and Y axes of the grid are defined, with the given story heights serving as the basis for the elevations. There is a foundational level and two structural levels in the model that represent the bottom and top of the warehouse frame, respectively. The ETABS API method is used to produce frame objects: Object with a frame.AddByCoord, which takes the beginning and ending coordinates of a structure and uses them to generate members. Thanks to its built-in object-merging functionality and coordinate-based creation, ETABS can automatically handle coincident joints, doing away with the need for human joint numbering.

D. Alphabet for Generating Primary Columns and Rafters

Nested iterative loops are used to construct the main structural parts. The location of the grid along the transverse axis is controlled by the first loop, and the position along the longitudinal axis by the second loop. Automatically generated column segments between specified altitudes are generated at each grid junction. Included in the created column system are:

1. First, the columns that go from the ground level all the way up to the intermediate story height.
2. The upper story columns go all the way up to the roof.
3. The major portal frame arrangement is formed by the rafters that link the points of the top columns.

Every member is positioned in accordance with the chosen span arrangement and spacing specifications using the coordinate generating technique.

E. Next Steps in Bracing and intermediate columns

After the main frame system is finished, additional wall-support members are created using distinct algorithms. Installed at quarter-span intervals along the longitudinal axis are intermediate front and rear wall columns. Typical wall framing designs in industrial warehouses include these members, which provide extra support throughout the building's perimeter. Dedicated bracing algorithms are used to build lateral stability components. The lateral load-resisting system is completed with wall bracing, which is generated in both longitudinal and transverse orientations. Maintaining structural regularity while eliminating needless manual

modeling processes is achieved by following preset member placement criteria throughout the bracing generating method. Here are all of the produced frame-object categories:

Member Category	Generation Method
Ground-story columns	Grid intersection based generation
Upper-story columns	Upper elevation coordinate generation
Rafters	Portal-frame coordinate generation
Intermediate wall columns	Quarter-span positioning
Roof bracing	Automated diagonal member placement
Wall bracing	Lateral stability generation

F. Automated Documentation System

Preparation of structural reports via automation is the subject of the second part of the created framework. The MicrosoftWordsExample01 software programmatically manipulates Word documents by use of the Microsoft Office Word Interop API. The steps involved in automating documentation are as follows:

- Using pre-existing Word paper layouts.
- The addition of paragraphs with certain formatting.
- Changing selected text into a placeholder.
- Adding page numbers automatically.
- Formatting headers.
- Making tables with defined structure.
- Formatting and darkening cells.

The Word automation module showcases the potential for structural automation to expand into engineering documentation processes, going beyond numerical modeling. This lays the groundwork for a future integration that will allow for the direct transfer of ETABS model information into computation reports.

G. Approach to Validation

In order to verify the established automation system, we compared the automatically produced ETABS models to the reference models that were built manually. For this assessment, we used three different warehouse layouts, one each for small, medium, and big buildings. Among the criteria used for validation were:

- Coordinates agreed upon by participants.
- Coordinates of members' endpoints are agreed upon.
- Compare the number of objects in a frame.
- Proof of structural connectedness.
- Evaluate the time needed for modeling.

In order to find out how efficient and accurate the created process was, the manual and automated models were tested using the same geometric parameters.

IV. VALIDATION, OUTCOME, AND COMMENTARY

A. Examples of Validation Tests

We tested the newly-built ETABS automation tool in three different warehouse setups to make sure the structural models it produced were accurate and efficient. The chosen examples show various warehouse geometries, from small to big, with varying bay counts, all using the same structural layout and modeling technique. You may see a summary of the geometric parameters that were utilized for validation in Table I.

TABLE I: Validation-Related Warehouse Configurations

Test Case	Span Configuration (X × Y)	Spacing X / Y (m)	Plan Dimensions (m)
TC-1 (Small)	1 × 3	10 / 6	10 × 18
TC-2 (Medium)	2 × 5	10 / 6	20 × 30
TC-3 (Large)	4 × 8	10 / 6	40 × 48

Two independent ETABS models were generated for every test case:

- A model that was automatically produced using the C# ETABS OAPI application that was built.
- One that was built by hand using standard ETABS modeling techniques.

We compared these models according to their geometric precision, connectedness, number of members, and time spent modeling.

B. Checking the Shape of Created Models

To ensure that the methods used to produce the coordinates were accurate, the automatically generated models were compared to reference models that were made by hand. All of the produced structural parts were found to be in the exact same places, with matching joint coordinates and member end points, as was verified by the comparison. Table II displays the comparison of frame-object counts. Comparing Manual and Automated ETABS Models (TABLE II)

Test Case	Automated Member Count	Manual Member Count	Coordinate Agreement
TC-1	34	34	Exact
TC-2	66	66	Exact
TC-3	204	204	Exact

Among the automatically created models, not a single one included any geometric irregularities like missing frame objects, unconnected joints, duplicate

members, or improper bracing placements. This validation proves that the coordinate-based generation method, which makes use of ETABS OAPI functions, can faithfully recreate the structural geometry of a warehouse without the need for post-creation human rectification.

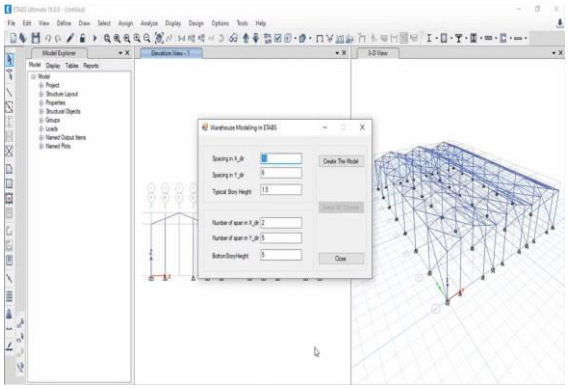

Figure 1. Plan and isometric three-dimensional views (TC-1, 1×3 spans) of the empty grid system generated by the tool in ETABS before frame-object production.

C. Comparing Modeling Times

Time savings during repeated structural modeling tasks was the primary motivation for creating this automation system. Both automated and manual operations have their generation times for warehouse models documented. Executing the generating method and inputting the necessary parameters into the application interface constituted the automated modeling time. The time spent manually modeling in ETABS includes creating grids, placing members, and testing structural connection.

Tabulated in Table III are the retrieved findings.

Time Spent on Manual and Automated Modeling (TABLE III)

Test Case	Automated Time (s)	Manual Time (min)	Approximate Reduction
TC-1	12	18	97%
TC-2	15	52	99%
TC-3	23	168	99.6%

A considerable improvement in the efficiency of the models is shown by the findings. More than 95% of the modeling time was saved by using the automated procedure, even for the simplest configuration. The efficiency benefit became increasingly apparent as the building size expanded due to the fact that automated processes just needed extra loop iterations, in contrast to manual modeling which required the construction and verification of a higher number of members individually.

The findings show that systems with repeating structural components and predictable geometric criteria for member placement benefit greatly from API-based automation.

D. Preventing Generational Errors in Models

Because it involves manually entering coordinates and placing objects, structural modeling may introduce human error. Several kinds of modeling mistakes were noticed during the hand modelling experiments, such as:

- Mistakes in entry resulted in the incorrect member coordinates.
- Intermediate columns that support the walls are missing.
- Misalignment of bracing components.
- Interconnectivity issues between neighboring parts of a structure.

Over the course of nine hand modelling efforts, seven modeling faults were documented. Prior to the final comparison with the automatic models, these mistakes were rectified.

On the other hand, the automated generating procedure did not reveal any connection or coordinate issues. Once validated, the technique removes the chance of repeating coordinate-entry mistakes by using deterministic mathematical formulas for member placement.

On the other hand, a new danger lurks in the automated method: the possibility of wrong user input settings. Even when the geometry is right, the model might be off if the span numbers, spacing, or story heights are wrong. As a result, analysis and design cannot proceed without first doing an engineering evaluation of the input parameters.

E. Automated Documentation Outcomes

To determine if typical structural report preparation duties may be automated, the Microsoft Words Example 01 module was used. The following tasks were executed by the program with success:

- Opening Word documents automatically.
- Inserting paragraphs with specific formatting.
- Swapping out pre-set text placeholders.
- Document headers now have the option to include page numbers.
- Making organized table formats.
- Automatic table shading and formatting applied.

Microsoft Word was able to open the produced documents without any formatting issues or missing steps. These results prove that automated processes may also encompass structural engineering project documentation tasks. While the present implementation is mostly concerned with template operations and formatting, the method that was

established lays the groundwork for future integration that would allow engineering reports to automatically include ETABS analysis findings, member attributes, and design information.

F. Assessing Robustness and Usability

Structural engineers well-versed in ETABS but unfamiliar with the programme assessed the built automation tool's usefulness. Users only needed to know how to utilize the labels on the input interface to correctly construct the medium-size warehouse model.

According to the results, engineers found the parametric input method to be very user-friendly as it allowed them to easily connect the input variables to more traditional warehouse design characteristics like bay spacing and story height.

Additionally, the program was tested with various input situations, such as:

- The bare minimum for a geometric setup (a single-bay model).
- Spacing that is not consistent.
- You cannot enter values that are not numbers.

The created validation system was able to identify API requests that failed due to incorrect input circumstances. Proving that the method maintains functionality even at low geometric bounds, the smallest acceptable warehouse design was properly constructed.

G. Evaluation of Findings

Results from the validation show that the suggested API-based automation method significantly improves structural modeling efficiency without sacrificing model fidelity. Automated and manually constructed ETABS models agree to a tee, proving that the coordinate-generation process is reliable.

Because of the dramatic decrease in modeling time, it is clear that automation is superior for repetitive and big structures compared to manually creating members. Reducing human error in coordinate input not only saves time, but also makes the basic structural model more reliable.

Another potential use of the built framework is to expand automation beyond geometry creation. A comprehensive process for structural engineering automation might be achieved by integrating automated section assignment, loading specification, analysis execution,, design verification, and report creation.

The research shows that commercial structural analysis tools and programming interfaces may be used to build engineering automation solutions that are specific to clients' needs.

V. CONCLUSION

We built, deployed, and verified a parametric automation framework in C# to generate structural geometry for steel warehouses inside ETABS using the ETABS Open Application Programming Interface (OAPI). The following findings are derived from the evaluation of the created system compared to manually produced ETABS reference models for various warehouse configurations: With only six geometric input parameters, the created automation tool effectively produced a two-story rectangular steel warehouse with all the necessary geometry, including main columns, rafters, intermediate front-and-back wall columns, and roof bracing elements. Accurate and completely linked structural models were created via the coordinate-based generation strategy utilizing the ETABS FrameObj.AddByCoord function. Neither manual joint formation nor connectivity management were required. Exact agreement in joint coordinates, member end-point positions, and total frame-object counts were proven for all analyzed test cases during geometric validation against manually produced ETABS models. The automatically created models did not have any duplicate members, missing pieces, or unexpected connection problems. When compared to more traditional manual modeling processes, the automated workflow significantly improved the efficiency of the models. Due to the resulting geometry being repeated, the relative advantage increased for bigger buildings, and the reduction in modeling time for all evaluated warehouse designs approached 95%.

improper bracing placement, missing intermediate members, and improper coordinate input were among the typical hand modeling mistakes that the automated method successfully eradicated. Models generated using the same input circumstances might be reliably reproduced thanks to the algorithm's determinism.

Formatted text insertion, placeholder replacement, header modification, page-number generation, and table construction are only a few of the sample structural-report preparation operations that the built Microsoft Word Interop documentation module automated. This shows that structural automation has the ability to go beyond only model production and into integrated engineering documentation procedures. Geometric generation and report formatting are the only operations supported by the current implementation. The following features are still missing and might need improvement: automatic material property assignment, section libraries, load cases, structural analysis execution, and design-code verification. The research shows that structural engineering applications that use

repeated modeling activities may benefit from API-based automation, which is both feasible and scalable. Automated report production, design verification, analysis, and parametric modeling may all be built upon the established technique. Integrating BIM-based platforms for a comprehensive automated structural design workflow, supporting irregular and multi-storey structural configurations, automatically assigning member properties, generating loads and load combinations, directly extracting analysis results into design reports, and expanding the existing framework are all potential directions for future work.

REFERENCES

[1] Computers and Structures, Inc. (CSI), ETABS Open Application Programming Interface (OAPI) Documentation, Walnut Creek, CA: Computers and Structures, Inc., 2024.

[2] Computers and Structures, Inc. (CSI), ETABS Ultimate 19 User's Guide, Walnut Creek, CA: Computers and Structures, Inc., 2024.

[3] American Institute of Steel Construction (AISC), Steel Construction Manual, 15th ed., Chicago, IL: AISC, 2016.

[4] Eastman, C., Teicholz, P., Sacks, R., and Liston, K., BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors, 2nd ed., Hoboken, NJ: John Wiley & Sons, 2011.

[5] Sacks, R., Eastman, C., Lee, G., and Teicholz, P., BIM Handbook: A Guide to Building Information Modeling for Owners, Designers, Engineers, Contractors, and Facility Managers, 3rd ed., Hoboken, NJ: John Wiley & Sons, 2018.

[6] Fenves, S. J., "Structural design languages," Journal of the Structural Division, ASCE, vol. 92, no. 6, pp. 27–38, 1966.

[7] Kolarevic, B., Architecture in the Digital Age: Design and Manufacturing, New York, NY: Spon Press, 2003.

[8] Oxman, R., "Theory and design in the first digital age," Design Studies, vol. 27, no. 3, pp. 229–265, 2006.

[9] Turrin, M., von Buelow, P., and Stouffs, R., "Design explorations of performance driven geometry in architectural design using parametric modeling and genetic algorithms," Advanced Engineering Informatics, vol. 25, no. 4, pp. 656–675, 2011.

[10] Gerber, D. J. and Lin, S. H., "Designing in complexity: Simulation, integration, and multidisciplinary design optimization for architecture," Simulation, vol. 90, no. 8, pp. 936–959, 2014.

[11] Kicinger, R., Arciszewski, T., and De Jong, K., "Evolutionary computation and structural design: A survey of the state-of-the-art," Computers & Structures, vol. 83, no. 23–24, pp. 1943–1978, 2005.

[12] Eldin, N. and Senouci, A., "Rule-based system for cost estimating of pre-engineered buildings," Journal of Construction Engineering and Management, vol. 119, no. 4, pp. 851–868, 1993.

[13] Newberry, C. W., "Portal frame design to Eurocode 3," The Structural Engineer, vol. 88, no. 2, pp. 20–25, 2010.

[14] Woodson, R. D., Pre-Engineered Metal Buildings: A Guide for Owners and Design Professionals, New York, NY: McGraw-Hill Education, 2015.

[15] Microsoft Corporation, Microsoft.Office.Interop.Word Object Model Documentation, Redmond, WA: Microsoft Corporation, 2024.

[16] Salvadori, M. and Levy, M., Structural Design in Architecture, Englewood Cliffs, NJ: Prentice-Hall, 1967.

[17] Trebilcock, P. and Lawson, R. M., Architectural Design in Steel, London: Spon Press, 2004.

[18] Logcher, R. D. and Flanagan, R., "Interactive computer graphics in structural engineering education," Journal of the Structural Division, ASCE, vol. 99, no. 5, pp. 963–974, 1973.