UNIVERZA V LJUBLJANI

FGG Fakulteta za gradbeništvo in geodezijo

ARMIN MEMARI

A BIDIRECTIONAL FRAMEWORK FOR BIM AND AGENT-BASED MODELS IN BUILDING ENERGY SIMULATION

DVOSMERNA INTEGRACIJA AGENTNIH MODELOV Z BIM ZA SIMULACIJO ENERGIJSKE UČINKOVITOSTI STAVB

BIM A+ European Master in Building Information Modelling

Master thesis No.:

Supervisor:
Assist. Prof. Luka Pajek, Ph.D.

Co-supervisor:
Ilaria Pigliautile, Ph.D.

President of the Committee
Prof. Goran Turk, Ph.D.

Ljubljana, 2025

ERRATA

Page	Line	Error	Correction

BIBLIOGRAFSKO – DOKUMENTACIJSKA STRAN IN IZVLEČEK

UDK:	**004.414.23:620.92(043.2)**
Avtor:	**Armin Memari**
Mentor:	**doc. dr. Luka Pajek, mag. inž. stavb.**
Somentor:	**dr. Ilaria Pigliautile, dott. ing.**
Naslov:	**Dvosmerna integracija agentnih modelov z BIM za simulacijo energijske učinkovitosti stavb**
Tip dokumenta:	**Magistrsko delo**
Obseg in oprema:	**76 str., 32 sl., 8 pregl., 72 vir.**
Ključne besede:	**integracija BIM-ABM, simulacija energijske učinkovitosti stavb, vedenje uprabojevalcev, dinamično načrtovanje, agentno modeliranje (ABM)**

Izvleček:

To delo obravnava pomembno “vrzel v učinkovitosti” pri energijski simulaciji stavb, ki nastane, kadar se v tradicionalnih delovnih tokovih BIM uporabljajo fiksni, standardizirani urniki zasedenosti. Za izboljšanje natančnosti napovedovanja raziskava predstavlja nov, avtomatiziran okvir za dvosmerno integracijo med informacijskim modeliranjem stavb (BIM) in modeliranjem z agenti (ABM). Okvir uporablja petplastni delovni tok za kroženje podatkov med podrobnim modelom v Autodesk Revit in platformo ABM. V praksi se BIM model izvozi v ABM, kjer visoko ločljive simulacije na podlagi gibanja uporabnikov ustvarijo dinamične, prostoru prilagojene profile zasedenosti. Osrednji prispevek je namenski vtičnik pyRevit, ki samodejno ponovno integrira urnike, ustvarjene z ABM, v okolje Revita. Z nadomestitvijo enotnih statičnih urnikov z dinamičnimi se izboljšajo predpostavke o notranjih obremenitvah v energijskih simulacijah. Primerjalna analiza pokaže, da vključitev dinamičnih podatkov o zasedenosti merljivo vpliva na rezultate simulacij, pri čemer se pomembno spremenijo kazalniki, kot sta letna intenzivnost rabe energije (EUI) in operativne emisije ogljika. Rezultati potrjujejo, da je avtomatizirana, dvosmerna integracija BIM–ABM izvedljiva pomembna za doseganje bolj realističnih simulacij učinkovitosti stavb. Predlagani okvir tako ponuja razširljiv pristop za vključevanje dinamike človeškega vedenja v proces načrtovanja, kar vodi k energijsko učinkovitejšim in uporabniku prijaznejšim stavbam.

BIBLIOGRAPHIC– DOKUMENTALISTIC INFORMATION AND ABSTRACT

UDC: **004.414.23:620.92(043.2)**

Author: **Armin Memari**

Supervisor: **Assist. Prof. Luka Pajek, Ph.D.**

Cosupervisor: **Ilaria Pigliautile, Ph.D.**

Title: **A Bidirectional Framework for BIM and Agent-based Models in Building Energy Simulation**

Document type: **Master thesis**

Scope and tools: **76 p., 32 fig., 8 tab., 72 ref.**

Keywords: **BIM-ABM Integration, Building Energy Simulation, Occupancy Behavior, Dynamic Scheduling, Agent-based Modeling (ABM)**

Abstract:

This work addresses the significant “performance gap” in building energy simulation, which arises when fixed, standardized occupancy schedules are used in traditional BIM workflows. To improve prediction accuracy, this study presents a novel automated framework for bidirectional integration between Building Information Modeling (BIM) and Agent-Based Modeling (ABM). The framework uses a five-layer workflow for closed-loop data exchange between a detailed Autodesk Revit model and the ABM platform. In practice, the BIM model is exported to ABM platform, where high-fidelity simulations generate dynamic, space-specific occupancy profiles from simulated occupant movements. The core contribution is a custom pyRevit add-in that automatically reintegrates these ABM-generated schedules into the Revit environment. By replacing uniform static schedules with dynamic ones, this process refines the internal load assumptions in energy simulations. Comparative analysis shows that incorporating the dynamic occupancy data has a measurable impact on simulation outcomes, notably changing metrics like annual Energy Use Intensity (EUI) and operational carbon emissions. The results confirm that automated, bidirectional BIM–ABM integration is both feasible and valuable for making building performance simulations more realistic. Ultimately, this framework offers a scalable approach for embedding human behavior dynamics into the design process, leading to more energy-efficient, occupant-centric buildings.

ACKNOWLEDGEMENTS

First, I would like to thank God for granting me the strength, patience, and perseverance to complete this work.

I am deeply grateful to my supervisor, doc. dr. Luka Pajek and co-supervisor, dr. Ilaria Pigliautile, for their invaluable guidance, support, and encouragement throughout this journey. Their expertise and dedication have been essential to the completion of this thesis.

I would like to sincerely thank my parents and my dear sister Rojin, for their unwavering love, support, and belief in me. Their constant encouragement has been a source of motivation at every stage.

My appreciation also goes to all those involved in the BIM A+ program, from whom I have learned a great deal. I am thankful for the knowledge, experiences, and opportunities provided through this program.

I am equally grateful to my colleagues in the BIM A+ program, with whom I have shared this journey. The challenges, achievements, and friendships we built together made this experience truly meaningful.

Finally, I would like to thank my friends in Iran, Sina, Vahid, Akbar, and Rasoul, with whom I have shared many years of growth, learning, and lasting friendship.

TABLE OF CONTENTS

INDEX OF FIGURES

INDEX OF TABLES

1 INTRODUCTION

1.1 Background and Context

The building sector accounts for approximately 40% of global energy consumption and 36% of carbon dioxide emissions, positioning building performance optimization as a critical component of climate change mitigation strategies [1]. Within this context, Building Information Modeling (BIM) has emerged as a transformative approach for integrating design, construction, and operational data throughout the building lifecycle [2]. BIM platforms provide comprehensive repositories of geometric, semantic, and parametric information that can inform energy performance analysis from the earliest design stages [3].

However, current building energy simulation practices face significant limitations in accurately representing occupant behavior, which is widely recognized as one of the largest sources of uncertainty in building performance predictions [4]. Traditional energy models typically employ fixed occupancy schedules that assume uniform behavior patterns across similar building types, regardless of variations in spatial configuration, user demographics, or operational contexts [5]. This oversimplification contributes to the well-documented "performance gap" between predicted and actual building energy consumption [6].

The emergence of Agent-Based Modeling (ABM) presents a compelling opportunity to address these limitations by simulating individual and collective occupant behaviors through autonomous agents that interact with their environment and each other [7]. Unlike traditional deterministic approaches, ABM can capture the heterogeneous, stochastic, and adaptive nature of human behavior in buildings, generating dynamic occupancy profiles that more accurately reflect real-world conditions [8]. Recent research has demonstrated that ABM-enhanced building simulations can identify energy-saving potentials of 10-25% in residential buildings and 5-30% in commercial buildings [9].

1.2 Problem Statement

Despite the recognized potential of integrating BIM and ABM for enhanced building performance simulation, significant barriers persist in achieving seamless interoperability between these platforms. Current integration attempts are characterized by manual data transfer processes, platform-specific implementations, and unidirectional workflows that limit the potential for iterative design optimization [10]. The lack of standardized protocols for bidirectional data exchange creates substantial barriers to widespread adoption of integrated approaches [11].

Most critically, existing BIM-ABM integration frameworks remain largely theoretical or limited to proof-of-concept demonstrations that do not address the practical challenges of automated workflow implementation [12]. The absence of "round-trip" data exchange capabilities means that insights gained from ABM simulations cannot effectively inform design modifications within BIM environments, severely limiting the iterative potential that drives effective building design processes [13].

Furthermore, the complexity of manual data transformation between platforms introduces significant risks of errors, inconsistencies, and loss of semantic information during the exchange process [14]. These challenges are compounded by the diverse data structures, coordinate systems, and semantic definitions employed by different software platforms, creating interoperability issues that prevent effective integration [15].

1.3 Research Objectives

This research aims to develop and implement a novel, automated framework for bidirectional integration between BIM and ABM to enhance the accuracy of building energy simulation. The primary objectives are to:

1. Develop a workflow for automated data exchange between BIM and ABM platforms.
2. Implement a method to reintegrate dynamic occupancy schedules generated by ABM into the BIM environment.
3. Quantify the impact of ABM-derived occupancy data on energy simulation results compared to traditional static schedules.

1.4 Research Methodology Overview

This study employs a Design Science Research (DSR) approach within a case study framework, focusing on the creation and evaluation of artifacts designed to solve identified problems [16]. The research combines qualitative literature synthesis with quantitative experimental validation through the development and testing of a five-layer BIM-ABM integration workflow.

The methodology progresses through systematic stages: BIM model preparation and baseline energy simulation, ABM-based occupancy simulation using high-fidelity pedestrian modeling, automated data reintegration through custom scripting solutions, and comparative performance analysis. A detailed office building model serves as the controlled experimental environment, enabling precise evaluation of the framework's impact on energy simulation accuracy.

1.5 Scope and Limitations

The scope of this research encompasses the technical development of bidirectional BIM-ABM integration workflows, with specific focus on occupancy-driven building energy simulation enhancement. The study is limited to office building typologies and utilizes specific software platforms to ensure controlled experimental conditions.

Key limitations include the use of synthetic rather than real-world occupancy data for validation, the focus on a single building typology, and the reliance on specific software platforms that may limit generalizability. Future research should address these limitations through expanded validation studies and multi-platform implementation strategies.

1.6 Thesis Structure

This thesis is organized into six chapters that systematically address the research objectives:

Chapter 2 presents a comprehensive literature review examining the current state of BIM and ABM integration, identifying key challenges and research gaps that motivate this investigation.

Chapter 3 details the methodological framework, including the five-layer workflow design, case study description, and analytical procedures employed for framework validation.

Chapter 4 describes the implementation process, including technical details of the bidirectional integration workflow, custom scripting development, and integration testing procedures.

Chapter 5 discusses the outcomes of the study and addresses the research questions, interpreting the comparative results in context.

Chapter 6 concludes with a synthesis of findings, discussion of implications for practice and research, and recommendations for future development of BIM-ABM integration approaches.

2 LITERATURE REVIEW

2.1 Introduction & Overview of the Problem

The integration of Building Information Modeling (BIM) and Agent-Based Modeling (ABM) has emerged as a promising approach to improve the accuracy of building energy simulations, particularly by capturing the dynamic and stochastic nature of occupant behavior. Traditional energy simulation methods often rely on static occupancy profiles, which can lead to discrepancies between predicted and actual energy consumption [17, 18]. Recent research has focused on developing frameworks that combine BIM's rich geometric and semantic data with ABM's ability to simulate individual and collective occupant behaviors, resulting in more realistic and actionable insights for building performance optimization [17, 18, 19]. However, challenges remain in achieving seamless interoperability, automating data exchange, and validating these frameworks across diverse building types and operational contexts [3, 11]. This review synthesizes the current state of research on bidirectional BIM-ABM frameworks, highlights key methodologies, and identifies future directions for enhancing building energy simulation through dynamic occupancy modeling.

2.2 Building Information Modeling (BIM) in Energy Simulation

Energy simulation and data modeling have become indispensable, but the process is complicated by fragmented tools and assumptions. As shown in **Figure 1** the evolution of BIM applications in energy analysis has progressed from simple geometric modeling to comprehensive performance assessment tools. Gerrish et al. [14] identified key challenges in BIM application to building energy performance visualization and management, noting that while BIM provides rich geometric and semantic information, the translation to energy models often encounters interoperability issues.

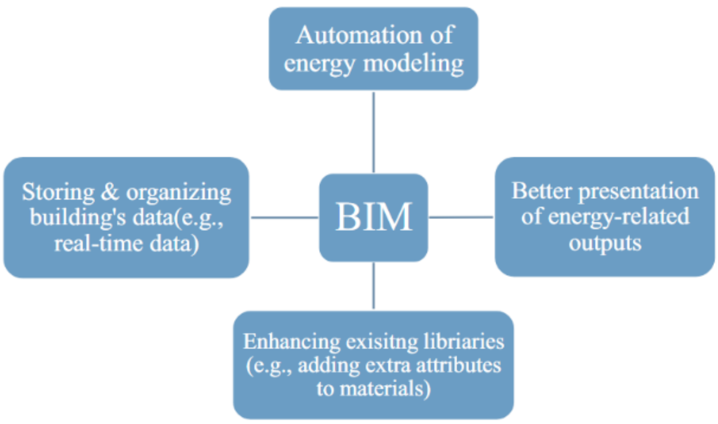


Figure 1 Application of BIM in BEM [3]

Similarly, Ciccozzi et al. [10] conducted a comprehensive review of interoperability strategies between BIM and Building Energy Models (BEM), highlighting the persistent challenges in data exchange protocols and the need for standardized workflows. Leading studies report that many BIM-to-energy workflows *do* work in principle, but often require extensive fixes. For instance, one study found a semi-automated BIM-to-BEM workflow produced energy models that were up to 7.5% smaller than the original design, with missing elements in the thermal envelope [20].

2.2.1 Evolution from Static to Dynamic Approaches

Traditional energy models often use fixed (static) occupancy schedules; which "are acknowledged as the main source of discrepancy between the predicted and actual building energy performance" [21]. The limitations of fixed schedules become apparent when analyzing actual building performance data. Research demonstrates that uniform scheduling across similar building types fails to capture site-specific variations in occupancy patterns, circulation flows, and operational requirements [22, 23]. In other words, treating occupant behavior as a fixed schedule usually fails to reflect reality, creating large "performance gaps" between simulated and real energy use [21, 24]. Buildings with identical functions but different spatial configurations, user demographics, or operational contexts exhibit markedly different energy consumption patterns that static schedules cannot represent [25, 26].

Contemporary research has increasingly focused on dynamic approaches that incorporate real-time occupancy data. Page et al. [27] developed a generalized stochastic model for occupant presence simulation that accounts for variability in occupancy patterns, as illustrated in **Figure 2**. This work was further extended by Aerts et al. [28], who proposed methods for identifying and modeling realistic domestic occupancy sequences that better reflect actual building usage patterns.

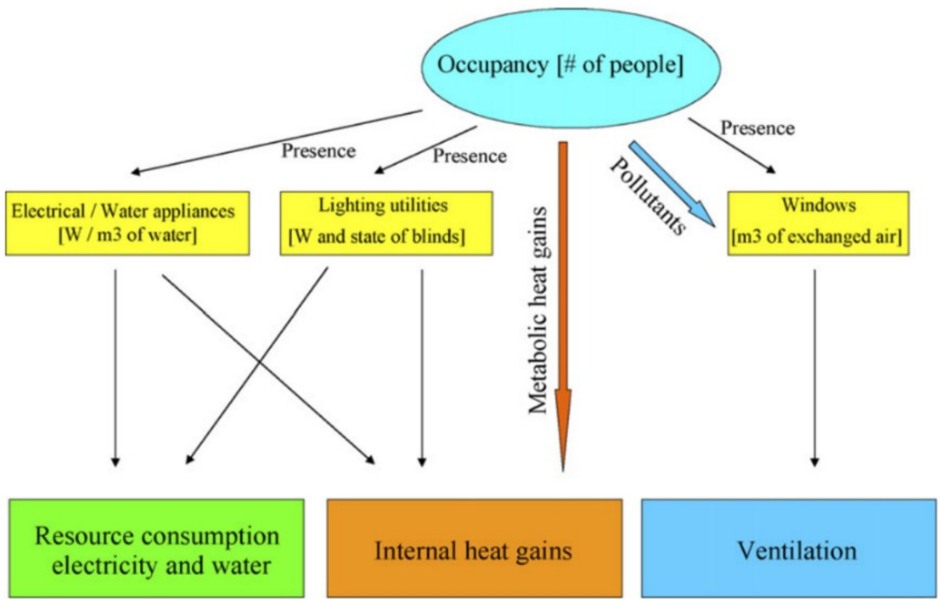


Figure 2 Occupancy model outputs for stochastic behavior models [27]

2.2.2 The Role of BIM in Occupant-Centric Simulations

BIM platforms serve as comprehensive data repositories that can support occupant-centric simulations through their rich semantic information. The spatial relationships, room functions, and occupancy parameters embedded in BIM models provide essential inputs for agent-based simulations [29]. However, the effective utilization of this information requires sophisticated data extraction and transformation processes that can bridge the gap between architectural design intent and energy simulation requirements [13, 30].

2.3 Agent-Based Modeling (ABM) in the Built Environment

The appeal of ABM in buildings lies in its ability to naturally capture heterogeneity. Each person's schedule or comfort needs can be different. Agents may "think and act similar to humans" by design [31]. ABM has been used in the built environment for a range of tasks: analyzing occupancy comfort (thermal, visual), studying evacuation in emergencies, and even modeling neighborhood energy flows. It is especially suited for capturing feedback, as agents can communicate and adjust the system [4, 7]. As shown in **Figure 3**, there is a paradigm shift from conventional occupant modeling that treats occupants as passive sources of heat, moisture, and emissions to next generation modeling that considers occupants as active decision-making agents.

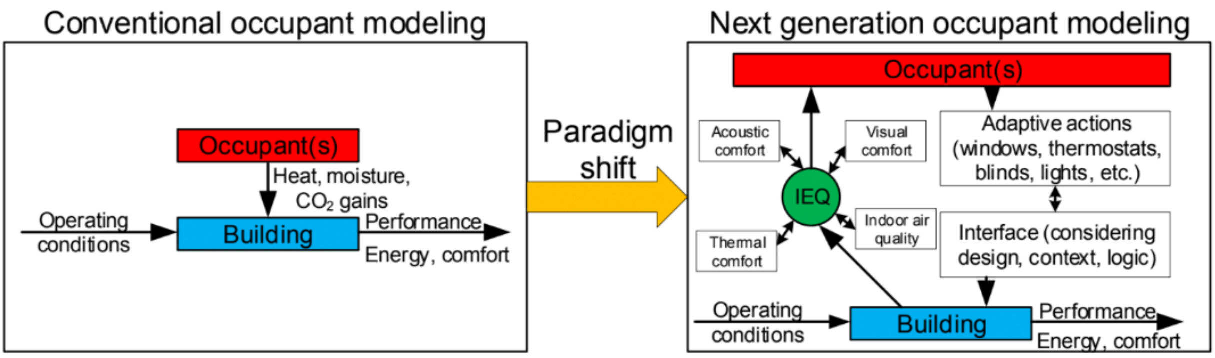


Figure 3 Shift from passive occupants to active, adaptive agents in buildings [32]

2.3.1 ABM Fundamentals and Applications in Buildings

Agent-based modeling fundamentally differs from traditional simulation approaches by representing individual entities (occupants) as autonomous agents with their own behavioral rules and decision-making processes [33, 34]. These agents (Figure 4) interact with their environment and with each other, creating emergent patterns that reflect real-world complexity [8, 35].

The application of ABM in building contexts has expanded significantly over the past decade. Andrews et al. [36] pioneered the integration of agent-based approaches in building design, demonstrating how occupant behavior modeling could inform architectural decisions. This foundation was built upon by subsequent researchers who developed more sophisticated models capable of capturing diverse occupant behaviors and their energy implications [37, 38].

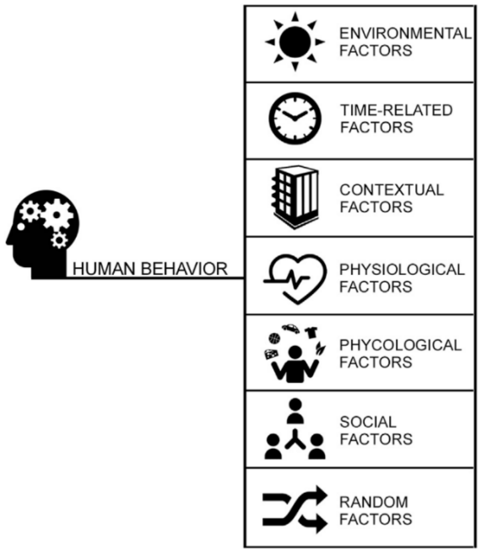


Figure 4 Defining the categories that shape human energy-related behavior in buildings [39]

2.3.2 ABM for Occupant Behavior Simulation and Data Generation

ABM's strength in occupant behavior simulation lies in its ability to model stochastic and heterogeneous behaviors that traditional deterministic models cannot capture. Jia et al. [40] developed a systematic approach to agent-based modeling of occupant behaviors in commercial buildings, emphasizing the importance of validation against real-world data. Their work highlighted the need for robust calibration procedures to ensure model accuracy and reliability.

The data generation capabilities of ABM are particularly valuable for building energy simulation. Unlike static schedules, ABM can generate dynamic occupancy profiles that reflect the complexity of human behavior patterns [41]. These profiles can then be used to drive more accurate energy simulations that better predict actual building performance.

2.4 ABM Applications in Building Energy Simulation

Agent-Based Modeling (ABM) has emerged as a powerful tool in building energy simulation, offering a more nuanced approach to representing occupant behavior and its impact on building energy performance. Occupant behavior is widely recognized as a major source of uncertainty in building performance predictions [4, 42], and traditional energy estimation tools often assume constant occupant behavior, leading to discrepancies between predicted and actual energy consumption. ABM addresses this limitation by accounting for diverse and dynamic energy consumption patterns among occupants, as well as the potential changes in their energy use behavior attributable to interactions with the building environment and with each other [18].

Several notable case studies have demonstrated the practical implementation of ABM in building energy simulation in **Table 1**. Azar et al. [43] proposed an ABM framework integrating people movement, energy consumption, and thermal comfort, validated by literature and actual data, which, in a campus case study, identified an optimal HVAC strategy achieving 19% energy savings without compromising comfort. Chen et al. [23] developed a comprehensive approach for simulating and visualizing occupant behavior in office buildings, combining an Occupancy Simulator for presence and movement, an occupant behavior model co-simulated with EnergyPlus, and an agent-based model in AnyLogic to visualize interactions and energy impacts, helping stakeholders better understand occupant energy use. Qiao & Yunusa-Kaltungo [44] developed a hybrid agent-based machine learning method for human-centered energy consumption prediction, combining the behavioral modeling capabilities of ABM with the predictive power of machine learning algorithms. Their approach showed significant improvements in prediction accuracy compared to traditional methods. Uddin et al. [45] presented a comprehensive study predicting occupant energy consumption in different indoor layout configurations using a hybrid agent-based modeling and machine learning approach. This research demonstrated how spatial configuration affects occupant behavior patterns and subsequently impacts energy consumption, providing valuable insights for building design optimization.

Table 1 Studies on occupant behavior in buildings using ABM [40]

REFERENCES	BUILDING TYPE	MODELED BEHAVIORS	BEHAVIOR DRIVERS /STIMULUS	KEY MODELING RULES	PLATFORM	IS VALIDATION INCLUDED?	ABM BASED ON REAL BUILDING
[18]	Commercial buildings	Blinds, lighting and equipment, Hot water use	Energy conservation events; word of mouth influence	High energy consumers will turn to low energy consumers over time	AnyLogic	No	No
[46]	Residential buildings	Not specified, but a generic modeling	Usual time; environmental factor	Based on belief-desire-and-intention (BDI) architectures	Brahms	No	No
[8]	Commercial (office) buildings	Clothing adjustment; personal fans on/off; personal heaters on/off; thermostat up/middle/down; Windows open/closed	Thermal conditions (temperature, humidity, air velocity)	Perceptual Control Theory (PCT), with a complex customized modeling rules	MATLAB	Yes	Yes
[34]	Commercial (office) buildings	Blind use; clothing adjustment; door use; fan/heater use; window use	PMV value that is influenced by temperature, air speed, RH, etc.	OODA (observe, orient, decide, and act) Loop based on three beliefs	MATLAB	No	No
[38]	Commercial buildings	Adjust clothes; use local heater/fan; contact manager; adjust overhead light, task light, and blinds	Load shedding events; communication with manager	Building occupant, tenant representative, and building manager have different behavior options	NetLogo	Yes	One building for calibration, the other for verification

[47]	Residential buildings	Window and air conditioning (AC) use	Temperature	Probability profiles for the modeled behaviors based on temperature variation	Repast	No	Yes
[48]	Office buildings	Lighting control; window operation; HVAC control	Temperature; CO2 concentration; daylight level	A drivers, needs, actions, and systems (DNAs) schema; Weibull functions to determine probability of behaviors	obFMU (customized with Functional mock-up unit)	No	Not specified
[40]	Commercial (office) buildings	Open and close of blinds, window, and door	External perceptions, value systems of human	Agents with perceptions, cognition, emotions, and physiology; behavior driven by thermal, visual comfort, and air quality	PMFserv	Yes	Yes

By incorporating ABM techniques, building energy simulations can provide more accurate predictions and insights, ultimately contributing to the design and operation of more energy-efficient buildings. The energy-saving potential of occupant behavior is estimated to be in the range of 10%-25% for residential buildings and 5%-30% for commercial buildings [9], underscoring the importance of accurately modeling and understanding occupant behavior in building energy performance.

2.4.1 Data Requirements for Occupant-Driven Energy Simulation

A major challenge for ABM and any occupant-aware model is data. To calibrate or validate these models, one needs real-world occupancy traces, preferences, and interactions [49]. Various methods have been tried. Some researchers instrument buildings: motion sensors, badge swipes, CO_2 sensors and WiFi logs can all provide occupancy data [50, 51]. Ni et al. [52] developed a cloud-based digital twin for historic buildings that collects real-time sensor data and stores it in the cloud. This setup is intended to feed AI models for occupancy prediction. Similarly, Uddin et al. [21] planned a validation using on-site sensors and even paper surveys to capture how occupants actually behave.

The ultimate goal of occupant modeling is to serve the building design and operator community, requiring models to balance practicality with accuracy. As illustrated in **Figure 5**, not every single contributing factor to occupant behavior can be captured; however, models can still provide reasonable predictions for most situations [5]. The quality and granularity of data significantly impact the accuracy of ABM simulations. Hoes et al. [53] emphasized the importance of understanding user behavior patterns and their representation in building simulation models. More recently, Quintana et al. [54] proposed cohort comfort models that leverage occupant similarity to predict personal thermal preferences with reduced data requirements, addressing one of the persistent challenges in occupant behavior modeling.

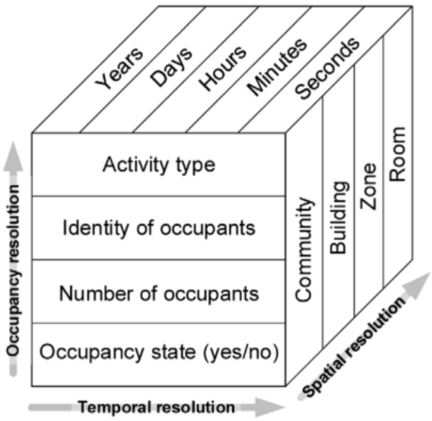


Figure 5 Resolution Requirements for Occupancy Modeling [5]

2.4.2 Limitations and Challenges of ABM for Energy Simulation

Despite its advantages, ABM for energy simulation faces several limitations and challenges. Malik et al. [7] identified ten key questions concerning agent-based modeling of occupant behavior, highlighting issues related to model complexity, computational requirements, validation procedures, and the need for standardized frameworks. The computational intensity of ABM simulations, particularly for large buildings with numerous occupants, remains a significant barrier to widespread adoption [5, 42].

Validation remains a critical challenge, as the stochastic nature of ABM makes it difficult to establish clear performance metrics and benchmarks [24, 55]. Furthermore, the diversity of human behavior patterns across different cultural contexts, building types, and operational conditions limits the generalizability of ABM models developed for specific contexts.

2.5 The Intersection: BIM and ABM Integration

The integration of BIM and ABM represents a significant advancement in building performance simulation, combining the geometric and semantic richness of BIM with the behavioral modeling capabilities of ABM [19, 56]. This intersection addresses key limitations of traditional approaches by enabling dynamic, occupant-driven simulations that leverage comprehensive building information.

2.5.1 Review of Unidirectional BIM to ABM Workflows

Most existing implementations focus on unidirectional data flow from BIM to ABM platforms. Micolier et al. [12] developed Li-BIM, an agent-based approach to simulate occupant-building interaction directly from Building Information Modeling data. Their framework demonstrated the potential for extracting spatial and semantic information from BIM models to populate ABM simulations, though it remained limited to forward data transfer.

Afkhamiaghda et al. [57] explored occupant behavior-based design using BIM-GIS integration, presenting an alternative design approach for architects. Their work highlighted the value of incorporating occupant behavior considerations early in the design process, though the integration remained primarily unidirectional from design intent to behavior simulation.

2.5.2 A Glimpse into Existing Bidirectional Frameworks

Several studies have leveraged the integrated BIM-ABM framework for multi-objective optimization, targeting energy consumption, thermal comfort, and sustainability metrics. Advanced algorithms such as NSGA-II, LightGBM, and evolutionary strategies have been used to identify Pareto-optimal solutions, enabling informed decision-making for design and retrofit scenarios [58, 59, 60]. These approaches have led to substantial improvements in energy efficiency and occupant satisfaction.

Dorrah & Marzouk [61] presented an integrated multi-objective optimization and agent-based building occupancy modeling approach for space layout planning. Their framework demonstrated bidirectional communication between design optimization and occupancy simulation, though the implementation remained focused on spatial optimization rather than comprehensive energy performance feedback.

2.5.3 Challenges and Limitations of Current Integration Methods

The integration of BIM and ABM faces significant technical and methodological challenges. Interoperability remains a primary concern, as different software platforms use varying data structures, coordinate systems, and semantic definitions [15]. The lack of standardized protocols for data exchange between BIM and ABM platforms creates barriers to seamless integration and limits the reproducibility of research findings.

Automation of the integration process presents another significant challenge. Manual data transfer and transformation processes are time-consuming, error-prone, and limit the scalability of integrated approaches [13]. The iterative nature of design processes further complicates automation, as changes in BIM models require corresponding updates in ABM simulations and subsequent reanalysis of results.

2.6 Key Technologies for Data Exchange & Integration

Effective integration of BIM and ABM requires robust data exchange mechanisms and integration technologies. The selection and implementation of appropriate data protocols significantly impact the success and efficiency of integrated workflows [62, 63].

2.6.1 Traditional Data Protocols

Industry Foundation Classes (IFC) and Green Building XML (gbXML) represent the most established protocols for building data exchange. IFC provides comprehensive semantic information about building elements, spaces, and relationships, making it valuable for ABM population [20]. However, the complexity of IFC schemas and the lack of standardized occupancy-related parameters limit its direct applicability to ABM integration.

gbXML, while more focused on energy analysis, provides structured space and zone information that can support occupancy modeling. However, its primary orientation toward HVAC systems and thermal analysis means that behavioral parameters required for ABM are often missing or inadequately represented [10]. This approach allows for the development of specialized tools that can bridge the semantic gap between BIM and ABM platforms while maintaining data integrity and traceability. A comparison between two data exchange protocols is demonstrated in **Table 2**.

Table 2 Comparing Two BIM-BEM Data Protocols [63]

Level	IFC	gbXML
Geometry	• Not suitable for irregular-shaped building design. • For complex interiors, additional modeling or annotations outside the standard IFC schema may be required.	• Not well-suited for modeling interior spaces, rooms, and their contents in detail. • Does not support curved surfaces or non-planar geometries.
Material	• There needs to be more validation for curved walls and free forms.	• Uses theoretical thermal characteristics which brings inaccuracies.
Space Type	• Conversion of 'IFC data' to 'IDF data' and space type in the 'IFC file' in the current application is not as smooth as in EnergyPlus.	• The relevant building design could be exported from gbXML to obtain level-2 BIM-based BEM functionality.
Thermal Zone	• The location of the building and simulation control data needs to be entered manually into the system.	• Preparation is required for Revit model creation to fulfil BEM needs.
Space Load and HVAC	• The establishment of HVAC transformation has not been achieved yet because of the drawbacks of the IFC file format.	• The absence of information like materials amid the process of transformation requires the modeler to input them manually.

2.6.2 Scripting and API-based Integration

Scripting and API-based approaches offer more flexible and automated integration solutions. pyRevit, in particular, provides direct access to Revit's internal data structures, enabling custom data extraction and manipulation workflows [64]. This approach allows for the development of specialized tools that can bridge the semantic gap between BIM and ABM platforms while maintaining data integrity and traceability.

The flexibility of scripting approaches enables the development of bidirectional data exchange mechanisms, where simulation results can be fed back into BIM models for visualization and further analysis [65]. However, these approaches require specialized programming knowledge and may be platform-specific, limiting their broader adoption [64].

Recent studies highlight similar challenges and opportunities in the energy modeling domain. For example, ElSayed et al. [66] describe how scripting environments such as OpenStudio "Measures" and APIs in tools like IES-VE allow users to programmatically control simulations and automate repetitive modeling tasks. They emphasize that while these scripting and API-based methods enable customization and workflow acceleration, they still demand advanced coding skills, which poses a barrier for non-expert users

2.6.3 The Role of Open-Source Solutions

Open-source solutions play an increasingly important role in BIM-ABM integration, providing accessible and customizable platforms for research and development. NetLogo, for instance, offers a versatile environment for ABM development with capabilities for importing and processing building data from various sources [67]. The platform's multi-agent programmable modeling environment has been used by hundreds of thousands of researchers worldwide for modeling complex systems [68].

The combination of open-source BIM libraries (such as IfcOpenShell) with open-source ABM platforms creates opportunities for developing comprehensive, cost-effective integration solutions [69, 70]. IfcOpenShell provides essential capabilities for reading, writing, and modifying IFC files, serving as a foundation for cross-platform OpenBIM applications [69]. However, the fragmented nature of open-source ecosystems and the need for technical expertise in multiple domains remain barriers to widespread adoption [70].

2.7 Research Gap and Problem Statement

The reviewed literature demonstrates that integrating BIM and ABM provides a robust framework for simulating dynamic occupancy and improving the accuracy of building energy performance predictions [17, 18, 19, 61, 71]. The ability to model individual and collective occupant behaviors, coupled with real-time data and advanced optimization techniques, enables more effective design, retrofit, and operational strategies. However, the field still faces significant challenges related to interoperability, automation, and standardization of data exchange between BIM and ABM platforms [3, 11, 13]. While middleware and ontology-based solutions show promise, their adoption is limited by the diversity of software tools and the complexity of real-world building systems.

The integration of BIM and agent-based models through bidirectional frameworks represents a significant advancement in building energy simulation, enabling more accurate, occupant-driven performance modeling and supporting multi-objective optimization for energy efficiency and comfort. While substantial progress has been made, challenges related to interoperability, automation, and generalizability remain, highlighting the need for further research and standardization.

Current research trends indicate a growing focus on hybrid approaches that combine ABM with machine learning techniques, cloud-based data collection systems, and advanced optimization algorithms. These developments suggest that the future of BIM-ABM integration lies in creating more intelligent, adaptive, and automated systems that can handle the complexity of real-world building operations while providing actionable insights for building performance optimization.

Although BIM (Building Information Modeling) and ABM (Agent-Based Modeling) integration shows promise, several key challenges remain before it can be widely used in real design projects. The biggest gap is the lack of **two-way data exchange**. Most existing studies only send data one way (from BIM to ABM), but very few allow ABM simulation results to update the BIM model in return. This "feedback loop" is critical for making design decisions, yet practical examples of it are still rare.

Another issue is the lack of **automation and scalability**. Current methods often require manual setup and data handling, which are slow, error-prone, and difficult to manage in large or complex projects. Even when some automation exists, changes to the BIM model often break the workflow, forcing all simulations to be redone. Without standard rules for how data should move between BIM and ABM, it's hard to create reliable, repeatable processes.

Maintaining **data consistency** is also a major challenge. Information often gets lost or altered when moving between BIM and ABM because they use very different ways of representing buildings and people's behavior. Important relationships and identifiers may not carry over correctly, which makes results less trustworthy.

Finally, there is very little **validation** to prove that ABM actually improves building energy simulations in measurable ways. Without strong evidence of benefits, many professionals hesitate to use these complex methods in practice.

In short, the main gap is the lack of tested, automated, and reliable two-way BIM-ABM integration methods that can support real-world design workflows.

2.7.1 Research Questions

This research is guided by three fundamental research questions that address the core challenges in BIM-ABM integration:

RQ1: How can bidirectional data exchange between BIM and ABM platforms be achieved while maintaining data integrity and semantic consistency?

RQ2: What are the most effective approaches for automating the integration workflow to support iterative design processes?

RQ3: How do ABM-generated occupancy profiles impact building energy simulation accuracy compared to traditional static schedules?

These questions frame the investigation and provide measurable criteria for evaluating the success of the proposed framework.

2.8 Study Objectives

This research aims to address the identified gaps through the development and validation of a comprehensive bidirectional integration framework. The primary objective is to create an automated, seamless workflow that enables dynamic occupancy data generated through ABM simulation to enhance BIM-based energy performance analysis while maintaining data integrity and semantic consistency throughout the process.

The specific objectives include:

Develop a Bidirectional Integration Framework: Create a systematic workflow that enables seamless data exchange between BIM and ABM platforms, addressing current interoperability limitations through standardized protocols and automated data transformation processes.

Automate Data Exchange Processes: Design and implement custom scripting solutions that eliminate manual data transfer requirements, enabling iterative design workflows that can rapidly incorporate behavioral insights into architectural design decisions.

Validate Framework Effectiveness: Conduct comprehensive comparative analysis to quantify the impact of ABM-generated dynamic occupancy profiles on building energy simulation accuracy relative to traditional static scheduling approaches.

3 METHODOLOGY

3.1 Introduction and Research Design

This chapter outlines the methodological framework employed to investigate the integration of Building Information Modeling (BIM) and Agent-Based Modeling (ABM) for enhanced building energy simulation. The research is guided by a pragmatic philosophy, seeking to develop and test a practical solution to the identified problem of interoperability and static occupancy assumptions. The strategy combines a **qualitative literature synthesis** (presented in Chapter 1) with a **quantitative, case-based experimental approach**. The core of this methodology is a novel five-layer workflow designed to establish a continuous, bidirectional data exchange loop between design and simulation environments. This chapter details each layer of this workflow, along with the case study building, the tools and technologies utilized, and the procedures for comparative analysis.

3.2 Qualitative and Quantitative Methods

A mixed-methods approach is used in this study to produce a thorough and reliable analysis. The main method used to carry out the **qualitative research** is a review of the literature, which summarizes the body of knowledge regarding BIM, ABM, and how they are combined to simulate building energy. This stage helps to:

1. Determine the main issue and the research gap in the workflows used for BIM-ABM integration today.
2. Provide the theoretical underpinnings and explain why a new framework is necessary.
3. Explore existing approaches and their limitations, informing the design of the proposed solution.

The **quantitative research** component is centered on the case study and framework implementation. This phase focuses on the measurable aspects of the research, including:

1. The creation and manipulation of the BIM model in Autodesk Revit.
2. The execution of the ABM simulation to generate a quantifiable dataset of occupant movements and counts.
3. The programmatic data exchange and reintegration, which transforms the quantitative output of the ABM simulation into actionable data within the BIM environment.
4. The final comparative analysis of the energy simulation results, which provides a numerical measure of the framework's effectiveness. This comparison of **Annual Energy Use Intensity** and **Operational Carbon** between the baseline and dynamic scenarios provides a data-driven basis for the study's conclusions.

3.3 Research Strategy and Case Study Description

3.3.1 Research Strategy

This study adopts a **Design Science Research (DSR)** strategy within a case study framework. DSR is concerned with the creation and evaluation of artefacts designed to solve identified problems. In this context, the artefact is the five-layer BIM-ABM integration workflow and the accompanying custom scripts. The case study, an office building, provides the real-world context in which this artefact is developed, implemented, and evaluated.

3.3.2 Case Study Building

A detailed BIM model of a typical office building was developed in Autodesk Revit 2025 to serve as the testbed for this research. This single-building focus allows for a detailed analysis of occupant behavior and its effects on energy consumption, providing a controlled environment to test the BIM-ABM integration framework. The building's layout and daily operations are critical to the simulation. The **primary spaces** for the simulation are nine enclosed offices and two open-plan offices, which serve as the main destinations for the simulated agents. The movement patterns of occupants in these spaces also govern the occupancy of **dependent areas**, such as corridors, pantries, and restrooms.

Function: Office space, as illustrated in **Figure 6** comprising a mix of enclosed private offices, open-plan work areas, meeting rooms, and support spaces (print room, pantry, restrooms).

Building Features		
	Item	Value
General	Total Building Area	431.67 m²
	Net Conditioned Building Area	383.02 m²
	Unconditioned Building Area	48.65 m²
	Number of Staff	34
Site and Climate Data	Weather File	Accra Kotoka Intl AP AA GHA ISD-TMYx WMO#=654720
	Latitude [deg]	5.61
	Longitude [deg]	-0.2
	Elevation	62.50 m
	Time Zone	0.00

Figure 6 Case Study - Architectural Floor Plan

Detail: The case study building contains a high geometric and semantic detail, including walls, floors, roofs, doors, windows, and critical MEP spaces, as presented in **Figure 7**.

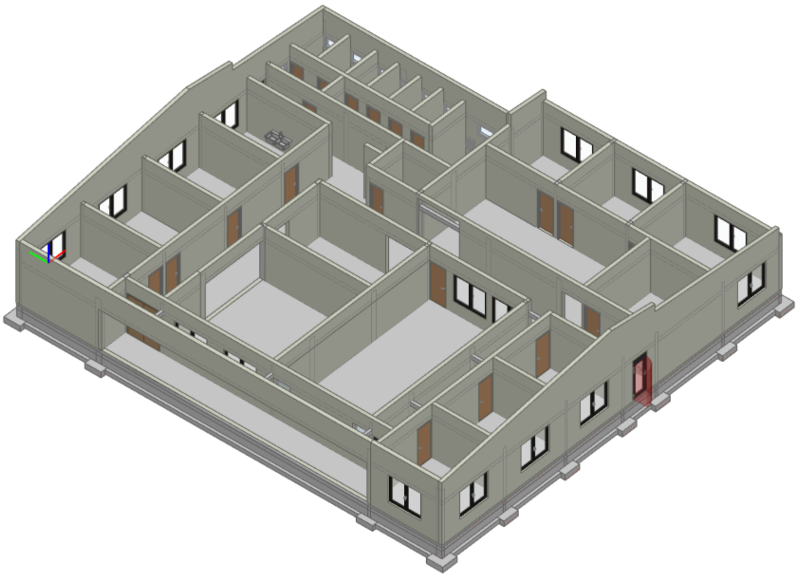


Figure 7 Case Study – Orthogonal View (roof is removed for better perception)

Baseline Condition: Spaces were assigned standard Revit Space Types pre-configured with fixed, uniform occupancy schedules (e.g., 8 AM to 5 PM, Monday to Friday). This represents the conventional, static approach and serves as the baseline for comparison.

Agent Definition: The definition of the simulated agents—representing building occupants—was explicitly derived from stakeholder requirements rather than general per capita metrics. The number of agents and their distribution across the office spaces were defined based on the specific operational needs and expected occupancy patterns provided by the building users. This user-centric approach ensures that the agent population accurately reflects the real-world intended use of the building, providing a validated and realistic foundation for simulating occupant behavior and its impact on energy performance.

To accurately model the dynamic behavior within this context, a predefined operational timetable for the building was used to calibrate the simulation's schedules and routes. This timetable (Table 6) specifies the typical usage patterns throughout the working day, providing the basis for agent arrival times and destination sequences.

3.4 The Five-Layer BIM-ABM Integration Workflow

The methodology follows a systematic workflow that progresses through five key layers to achieve a continuous data loop between design and simulation. The output of one layer becomes the input for the next, as visualized in **Figure 8**.

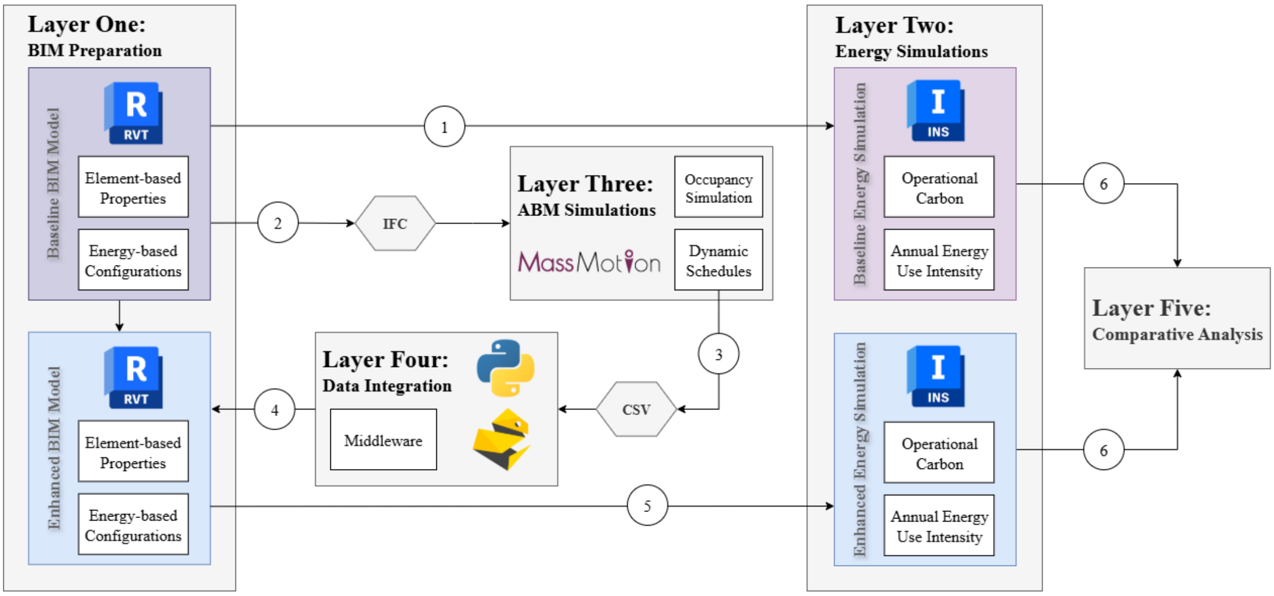


Figure 8 Methodology Flowchart based on Five Layers of Workflow

Layer One: BIM Preparation

This initial layer involves preparing the central data model for export and simulation.

1. **Process:** The authored Revit model, containing both element-based (geometry, materials) and energy-based (space types, fixed schedules) data, was exported using the IFC4.3 open standard. This format was selected for its rich semantic capabilities and compatibility with downstream simulation tools.
2. **Output:** A neutral, vendor-agnostic IFC file that preserves the geometric and functional properties of all building elements, especially IfcSpace entities and their unique IfcGUID identifiers.

Layer Two: Energy Simulation

This layer is executed twice to establish a before-and-after comparison, using the Autodesk Carbon Insight engine native to Revit.

1. **Baseline Simulation:** The initial energy analysis is performed on the model in its original state, utilizing the default fixed occupancy schedules. This generates a benchmark report of energy performance.
2. **Enhanced Simulation:** After dynamic data is reintegrated in Layer 4, a second, identical energy analysis is run. The only variable changed is the occupancy schedule data.
3. **Output:** Two sets of energy reports quantifying Annual Energy Use Intensity (EUI), operational carbon, and other key performance indicators.

Layer Three: ABM Simulation

The IFC file from Layer one is used to generate dynamic, behavior-driven occupancy data.

1. **Platform:** Oasys MassMotion [72] was chosen for its high-fidelity pedestrian simulation and direct IFC import functionality, which maintains spatial accuracy.
2. **Process:** The IFC model was imported to create a navigable 3D environment. Realistic movement was simulated by defining agent profiles (average walking speeds, body dimensions), origin-destination points within spaces, and a probabilistic operational timetable (Table 6) governing agent arrival, activity, and departure over a standard workday (07:00 – 17:00).
3. **Output:** A CSV file containing hourly occupancy counts for each individual space, identified by its IfcGUID. This file represents a dynamic occupancy profile, replacing a static schedule with time-resolved data.

Layer Four: Data Integration

This layer is the technological innovation of the workflow, automating the return of data from ABM to BIM.

1. **Tool:** A custom pyRevit add-in was developed using IronPython to serve as middleware.
2. **Process:** The script performs a multi-step process:
   a. Parsing: Reads the dynamic occupancy data from the MassMotion CSV output.
   b. Mapping: Uses the IfcGUID parameter to accurately link each data row to its corresponding MEP Space element within the Revit model.
   c. Schedule Generation: Programmatically creates new Revit Day Schedules and Year Schedules based on the normalized hourly occupancy values.
   d. Model Update: Duplicates original Space Types, assigns the new dynamic schedules to them, and applies these updated types to the spaces.
3. **Outcome:** The BIM model is automatically updated from a state of static assumptions to one of behavior-driven dynamic profiles, completing the "round-trip" data exchange.

Layer Five: Comparison Analysis

The final layer involves a quantitative evaluation of the framework's impact on energy performance prediction.

1. **Method:** A systematic comparison of the energy reports generated in the two instances of Layer two (Baseline vs. Enhanced).
2. **Key Metrics:** The analysis focuses on the differences in:
   a. Annual Energy Use Intensity (EUI)

b. Operational Carbon Emissions
c. Peak Load Values

3. **Purpose:** This comparison provides empirical, data-driven evidence of the effect that integrating dynamic occupant behavior has on simulation accuracy, validating the practical value of the proposed framework.

This workflow structure ensures that the entire process—from model creation to final comparative analysis—is automated and repeatable, addressing the research gap related to manual, error-prone data transfer.

3.5 Tools and Technologies

The implementation of the bidirectional BIM-ABM framework, as highlighted in **Table 3**, relies on a specific set of software, programming languages, and libraries, chosen for their capabilities in handling BIM data, conducting agent-based simulations, and enabling data exchange. These tools and their versions are detailed in the table below.

Table 3 Key Tools and Technologies

Tool/Technology	Version	Role in the Workflow	Stage
Autodesk Revit	2025	Core model creation, energy simulation (Carbon Insight)	Layer One/ Two
Industry Foundation Classes (IFC)	4.3	Neutral format for transferring model data to ABM	Interoperability
Oasys MassMotion	11.0	High-fidelity 3D pedestrian simulation and data generation	Layer Three
pyRevit	5.2.0	Framework for developing the custom Revit add-in	Layer Four
Python	3.11.0	Core language for data processing scripts	Layer Four
IfcOpenShell	0.8.2	Python library for reading and processing IFC files	Interoperability
Shapely	2.1.0	Python library for 2D geometric operations	Interoperability
NetLogo	6.4.0	An open-source ABM platform used for a proof-of-concept	Layer Three
Comma-Separated Values (CSV)	-	Lightweight format for transferring occupancy data	Interoperability
Autodesk AutoCAD	2025	CAD authoring and visualizing and modifying DXF files	Layer Three

4 IMPLEMENTATION

4.1 BIM-ABM Integration, A Workflow

In conventional BIM environments (Figure 9), operating schedules are typically standardized across buildings of the same type, regardless of variations in size, design, or functional patterns. For example, all educational facilities may be assigned identical operating schedules, despite differences in capacity or circulation patterns [22]. In contrast, agent-based simulation platforms can differentiate occupancy behaviors within similar building typologies, capturing nuanced variations in space usage. Traditional Building Information Modeling (BIM) environments lack mechanisms to represent dynamic occupant behavior realistically, while Agent-Based Modeling (ABM) platforms offer the ability to model such behaviors with higher fidelity.

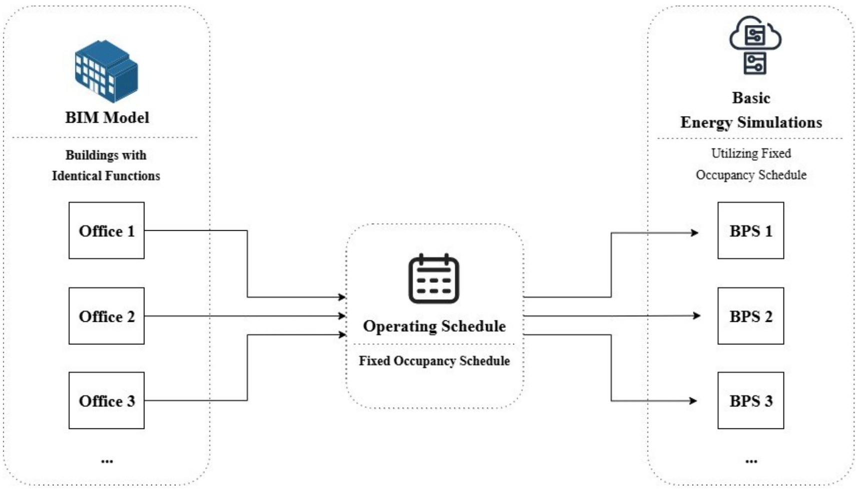


Figure 9 Conventional Building Performance Simulation via Fixed Operating Schedules

The ultimate goal is to enhance the accuracy of Building Performance Simulation (BPS) by incorporating dynamically simulated occupancy data. In this framework, BIM provides spatial and typological information to ABM tools, which simulate occupant movement and behavior based on spatial constraints and behavioral rules. The resulting occupancy schedules are then transferred back to BIM to inform energy simulations.

4.2 Intended Agent-Based Simulation

The simulation of occupant movement patterns within a building plays a critical role in shaping energy performance predictions. These movement algorithms determine the timing and duration of occupant dwelling within specific spatial zones, which in turn affects internal heat gains, ventilation loads, and lighting usage. The dwelling of agents in each IfcSpace entity across time intervals forms the foundation of dynamic occupancy schedules.

In this study, the behavior of simulated agents was continuously tracked throughout the building during defined circulation and evacuation scenarios. These patterns influence the intensity and frequency of space utilization and directly inform how operating schedules should be defined for energy simulation purposes. The simulation results, therefore, serve as a dynamic proxy for temporal space occupation, replacing fixed assumptions typically embedded in Space Type parameters in BIM.

However, a major limitation arises from the built-in population count functionality within Revit. The default calculation method assigns Number of People to every space, including dependent or circulation spaces that are not naturally occupied directly (e.g., corridors, storage rooms, stairwells). In practice, the occupancy of such spaces is contingent upon the usage of adjacent or primary spaces rather than being independent. This lack of contextual accuracy inflates population estimates and reduces the reliability of Revit's outputs for energy and evacuation modeling.

To address this shortcoming, an Agent-Based Modeling simulation is integrated with BIM. By leveraging ABM, as illustrated in **Figure 10,** dynamic occupancy schedules could be generated based on actual simulated movement patterns rather than static counts. This integration captures the interdependencies between primary and secondary spaces and more accurately represents how occupants transition through circulation zones in relation to their primary activities. The outcome is a more faithful representation of occupant-driven building performance, enhancing both energy simulation and safety analysis.

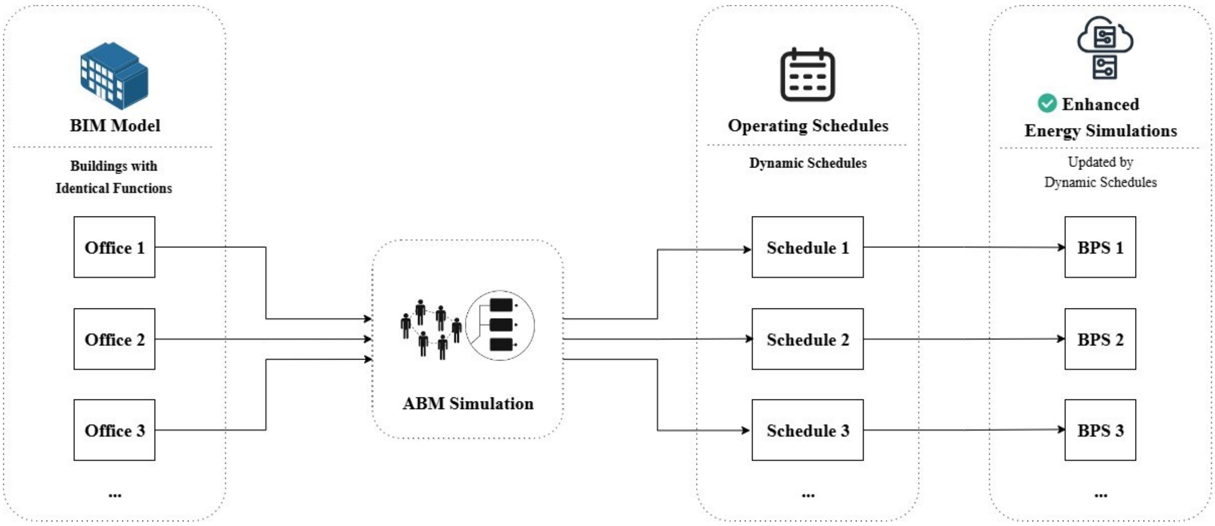


Figure 10 Utilization of Agent-Based Modeling in Building Performance Simulation

Importantly, these movement patterns are highly **context-specific**. Even among buildings with identical functions—such as schools or offices—the circulation logic, spatial configuration, and occupant behavior may vary significantly. Consequently, applying a generic schedule across all buildings of a similar type fails to capture such variances. Through ABM, it becomes possible to generate **building-**

specific occupancy schedules grounded in actual movement scenarios, enhancing the granularity and accuracy of BPS models.

4.3 BIM–ABM Data Exchange, A Round Trip

To initiate data exchange in the BIM–ABM integration process, it is first necessary to identify the ABM platform to be used, as each platform has distinct requirements for data input and output. This determination influences the overall data transfer strategy. One of the most challenging aspects of this process involves importing spatial and geometric data from the BIM model into the ABM environment.

Considering the interoperability concerns—which are central to this study—the data exchange mechanism must be tailored to the specific ABM simulation methodologies adopted by different software environments. Most critically, the definition of the simulation environment forms the basis of outbound BIM data transfer. As shown in **Figure 11** this environment modeling can be achieved either through **native interoperability** provided by the ABM software or via **indirect methods** such as custom scripting and data transformation. Conversely, the return flow of data—transferring simulation results from the ABM platform back into the BIM model—is not always straightforward, but the structure and format of simulation outputs are generally similar, which facilitates their reintegration into the BIM environment with appropriate handling.

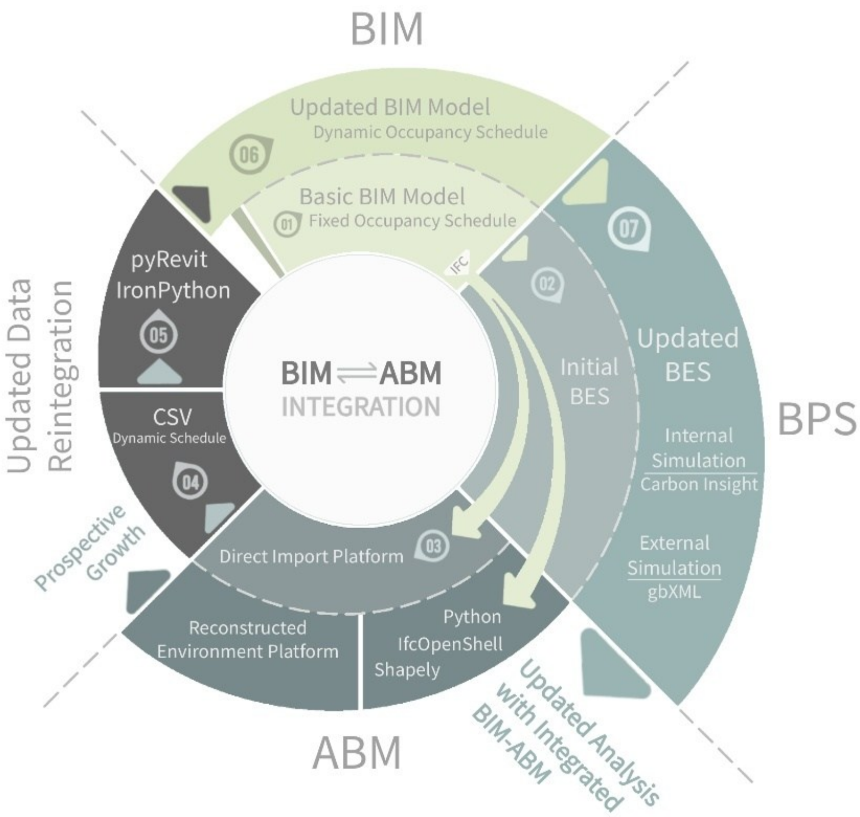


Figure 11 Bidirectional Flow of Data in BIM-ABM Integration

4.4 BIM Model Preparation

The BIM environment used in this project is Autodesk Revit. A detailed architectural model was created with high precision in terms of geometry and material specification, enabling thorough representation of building systems. From a data structure standpoint, Revit organizes these components under categories like Walls, Floors, Roofs, and others, with their properties defined through FamilyInstance and ElementType. Spaces within the model are defined using Space elements, which correspond to IfcSpace in the IFC schema. While the physical attributes of the model are robust, the default operating schedules applied to spaces are fixed and uniform across functionally similar areas (Table 4). This study builds upon the model's strengths by proposing a method to replace those fixed schedules with dynamic ones generated through ABM integration, thereby increasing the behavioral accuracy of energy simulations.

Table 4 Key BIM Parameters for Energy Simulation

Parameter Name	Location in Revit	Role in Simulation
Occupancy Density	Space Type Properties	Defines the nominal number of occupants per unit area for each space.
Operating Schedule	Space Type Properties	Governs the temporal variation of internal gains from occupants, lighting, and equipment.
Space Type	Space Element Properties	A category that bundles parameters for a specific type of space, e.g., "Office."
IfcSpace	Space Element	A BIM entity that encapsulates geometric and spatial characteristics of a room or zone.

Among all the elements in the model, spaces play a crucial role in Building Performance Simulation (BPS). In **Figure 12** it is evidenced that spatial zones are modeled using the Room or Space category (linked to IfcSpace in the IFC schema), and each is associated with a Space Type that includes key parameters such as occupancy density and operating schedules.

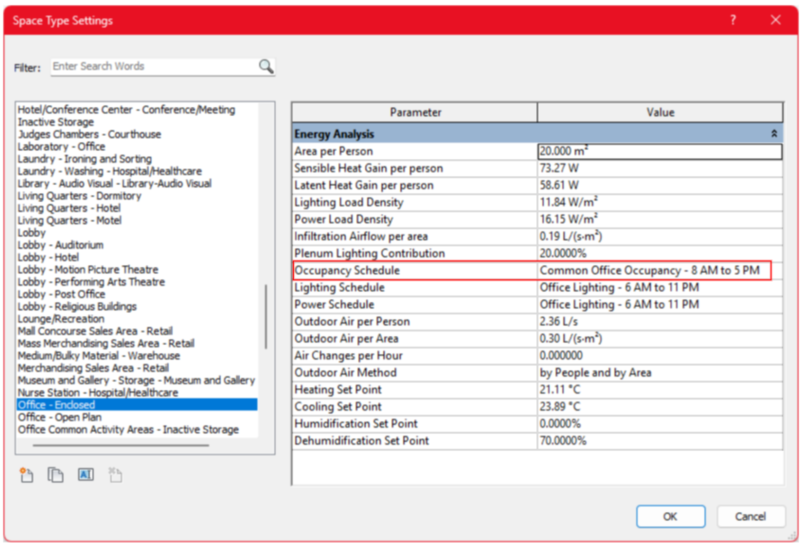


Figure 12 Key Parameters of Space Type, including Occupancy Schedule

These schedules govern the temporal variation of internal gains, particularly those arising from occupants, and thus significantly influence the thermal simulation results. Energy simulation software encapsulates occupant-related thermal loads indirectly through these schedules, which also embed assumptions regarding metabolic rate values, particularly in its recent versions. Consequently, modifying the occupancy schedule is the primary means by which user behavior and internal load assumptions can be adjusted in the energy model.

4.5 Initial BIM-based Energy Simulation

There are two primary approaches for conducting energy simulation using a BIM model. The first is an internal method using Autodesk Revit's native energy analysis tool, Carbon Insight. This approach enables energy simulations to be carried out directly within the BIM environment based on the spatial configuration, materials, and operating schedules defined in the model. The second approach involves exporting the model data into interoperable file formats such as gbXML or IFC, which can then be imported into specialized energy simulation software platforms like OpenStudio, EnergyPlus, or similar tools. Each method presents different benefits in terms of workflow integration, simulation control, and customization.

In this thesis, the first approach was selected. One major advantage of using Carbon Insight is that it allows the model to remain within the Revit environment without requiring external manipulation. All necessary modifications, including updates to occupancy schedules and space attributes, can be executed directly within Revit. Additionally, changes made through the Revit API not only affect internal energy simulation inputs but are also reflected in any exported data (e.g., gbXML or IFC), ensuring consistency across both internal and external simulation workflows.

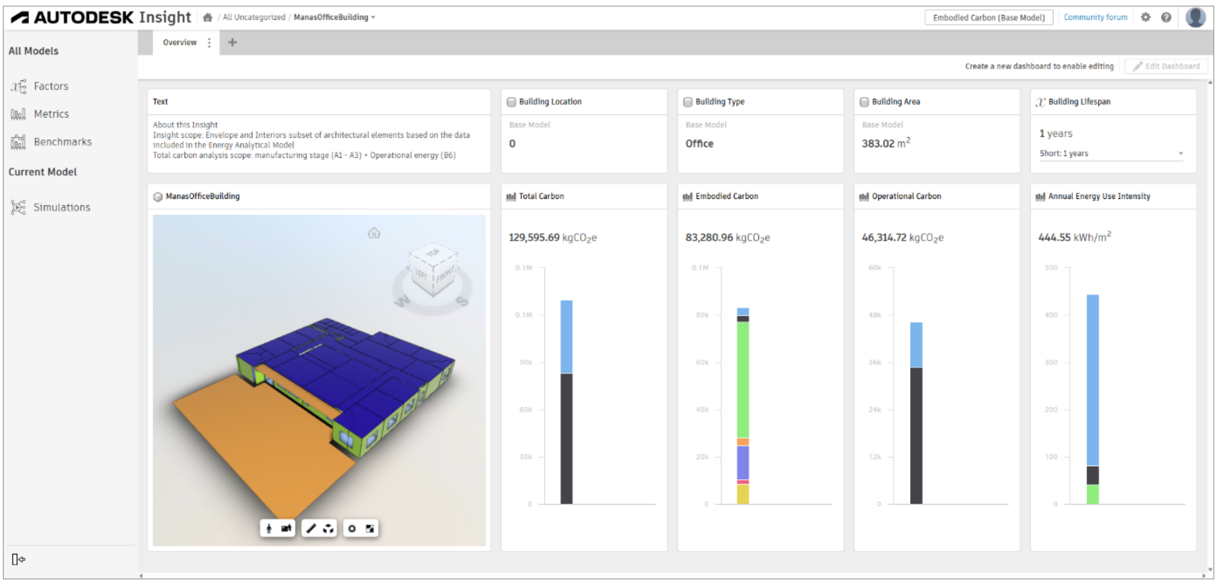


Figure 13 Initial Energy Simulation Results Generated by Carbon Insight

As demonstrated in **Figure 13**, running the energy simulation in Carbon Insight generates an overall report of calculations which contains the carbon emission amount and the annual energy use intensity. The initial energy simulation outputs will be compared with a secondary simulation after updating the operating schedule data based on ABM simulations. Any observed variations in energy consumption between the two scenarios will demonstrate the impact of replacing fixed schedules with dynamically generated ones, offering quantifiable evidence of the benefits of ABM integration in BPS. **Figure 14** clearly declares the workflow of implementing ABM simulation in BIM in order to enhance energy simulations process.

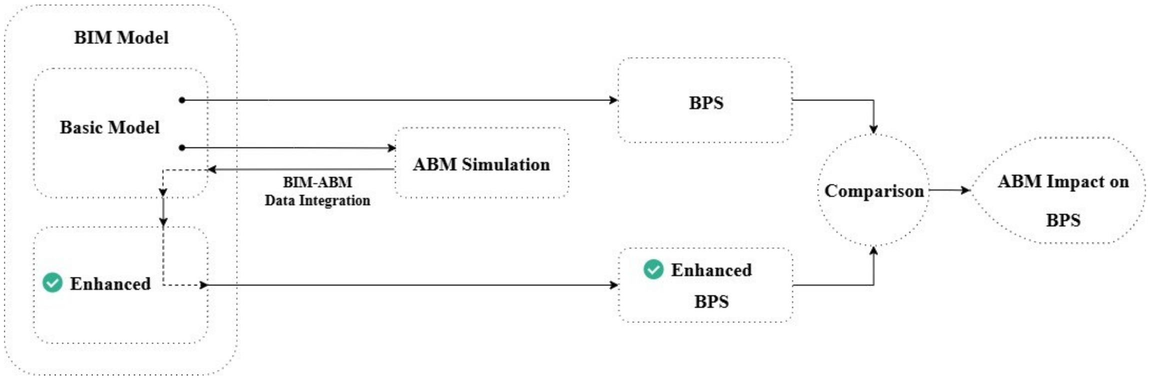


Figure 14 Impact of Agent-Based Modeling on Building Performance Simulation

4.6 ABM Simulation Workflow

Agent-Based Modeling (ABM) offers a powerful means of simulating occupant behavior in built environments. Several ABM platforms exist, each with different capabilities and strengths. Notable tools include NetLogo, known for its flexibility and educational orientation; AnyLogic, which integrates discrete-event and system dynamics; and MassMotion, which specializes in high-fidelity pedestrian movement simulation in architectural settings. These platforms vary significantly in terms of their simulation algorithms, 3D support, and interoperability with BIM.

NetLogo provides a programmable 2D grid movement area ideal for rule-based simulations and exploratory modeling. However, it lacks native support for complex 3D geometries and typically requires manual configuration of spatial inputs. AnyLogic, on the other hand, offers a hybrid modeling framework combining agent-based, discrete-event, and system dynamics methodologies, allowing users to simulate complex systems at multiple levels of abstraction. While it provides greater flexibility for modeling behavioral logic and stochastic interactions, its integration with architectural geometry is less direct and typically requires external preprocessing. In contrast, MassMotion is designed specifically for modeling pedestrian dynamics in real architectural contexts and supports direct IFC imports, making it highly suitable for BIM integration.

An initial prototype ABM pipeline was developed using NetLogo to test the workflow in a 2D space. However, for the full implementation the workflow adopted Oasys MassMotion, which supports native IFC integration and 3D simulation. Given these distinctions, the choice of ABM platform significantly affects the data preparation and integration strategies. MassMotion, because of its comprehensive capabilities in simulating agent movement behavior—especially within complex architectural environments—was selected as the core platform for the BIM–ABM integration workflow throughout this thesis. Its compatibility with IFC-based models and advanced features for pedestrian simulation make it particularly well-suited for dynamic occupancy modeling in energy simulations. However, considering that ABM platforms differ widely in their data exchange structures, this study also accounts for preparatory measures needed for other platforms. These include differences in how environments are defined, which may rely on native import tools or require the use of custom code for geometric and behavioral data translation.

In light of these structural differences, ABM platforms can be categorized based on how they handle environmental data input. **Direct Import Platforms**, such as MassMotion, offer seamless integration by supporting native IFC file imports. This capability enables users to operate directly within the architectural model, preserving spatial fidelity and minimizing preprocessing requirements. On the other hand, **Reconstructed Environment Platforms**, such as NetLogo, cannot directly ingest BIM or IFC files. As a result, their simulation environments must be manually reconstructed within the platform or generated using custom code that translates elements and related geometry into compatible formats. These distinctions shape the required workflow adaptations, as illustrated in **Figure 15**, for effective BIM–ABM integration depending on the selected platform.

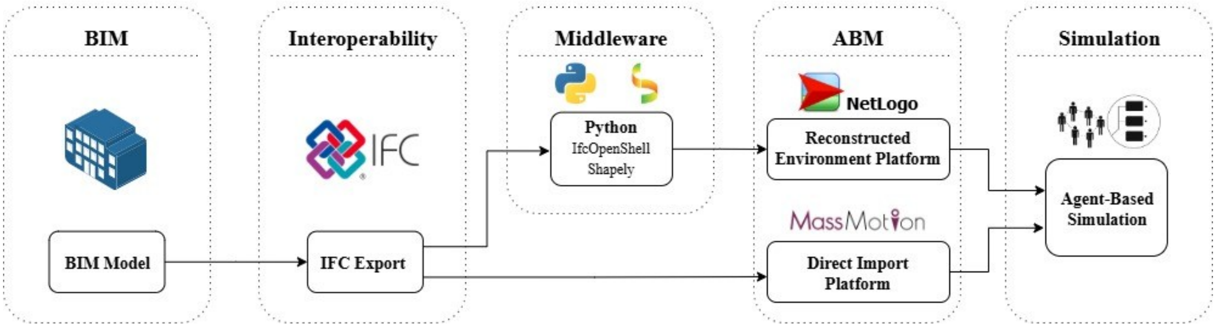


Figure 15 Flow of Data Between BIM and ABM Platforms

4.6.1 Reconstructed Environment Platforms Workflow: NetLogo

The goal of this stage is to establish an interoperable pipeline between BIM, which was authored in Autodesk Revit, and ABM, developed in NetLogo. This integration allows spatial boundary data derived from the BIM environment to define the physical constraints in which agents will operate. In this phase, the focus is on extracting spatial geometries—such as floors, walls, and other bounding elements—from an Industry Foundation Classes (IFC) file exported from Revit and formatting them into a CSV structure suitable for NetLogo import.

To support this workflow, a custom data extraction pipeline was developed using Python, specifically leveraging the IfcOpenShell library for IFC schema parsing and the Shapely library for geometric operations. The exported IFC file defines building spaces as IfcSpace entities, which encapsulate the geometric and spatial characteristics required for simulation. The script parsed each IfcSpace, extracted its boundary representations, computed centroid coordinates, and exported this information to a CSV file. CSV was selected due to its simplicity, human and machine readability, and ease of modification, making it well-suited for iterative simulation workflows. The resulting file was used as input for agent instantiation in the NetLogo environment. This early implementation served as a foundational effort in automating data translation between BIM and ABM domains.

Workflow Overview

The implementation of this integration pipeline consists of the following steps and the main stages are illustrated in **Figure 16**:

1. **Export of IFC Model:** The Revit model is exported using the IFC4.3 schema, ensuring full geometric fidelity and compatibility with modern parsers.
2. **Space Determination in GUI:** A graphical user interface (GUI) is created to allow users to select the thermal spaces involved in the project. This enables the extraction of boundary elements based on the chosen spaces.
3. **Parsing and Processing via Python:** Python scripts using IfcOpenShell are developed to load the IFC model and extract relevant physical elements such as walls, floors, ceilings, and openings.
4. **Geometry Simplification and Projection:** Extracted 3D geometry is simplified by projecting it onto the XY plane (a 2D floor plan representation).
5. **CSV Export:** Resulting geometry is serialized to a CSV file, preserving element type and coordinate data for use within the NetLogo environment.

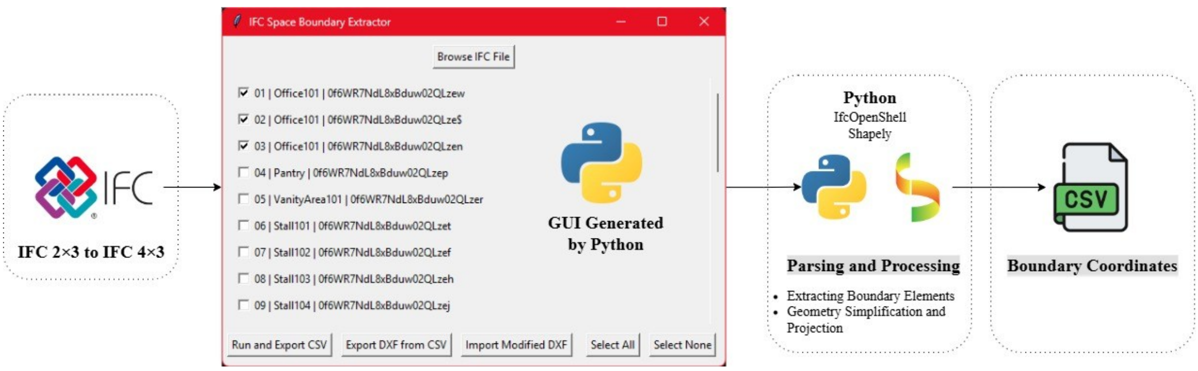


Figure 16 Data Integration Pipeline between BIM and Reconstructed Environment Platforms

IFC File Processing in Python

A) IFC File Opening and Geometry Extraction

The initial step is to open the IFC file and parse its contents using IfcOpenShell. This is done by calling the `ifcopenshell.open()` function on the exported IFC4.3 file. To ensure that the geometry is extracted with correct spatial placement, world coordinates are enabled using the geometry settings provided by `ifcopenshell.geom.settings()`.

A key decision was to avoid using the utility function `get_shape_vertices`, which has been deprecated or reorganized in recent versions of IfcOpenShell. Instead, geometry is extracted directly from the `shape.geometry.verts` property. This yields a flat list of 3D vertex coordinates in the format:

```
[x1, y1, z1, x2, y2, z2, ..., xn, yn, zn]
```

To convert this into a usable format for 2D geometry operations, a helper function projects the 3D vertices into 2D by discarding the Z-axis component. This results in a sequence of XY coordinate pairs that define the outline of the respective BIM elements.

B) Processing Floors and Walls

Two main element categories are extracted during this phase:

1. IfcSlab: Typically used for horizontal building elements such as floors and ceilings. Each slab's geometry is processed into a 2D polygon to represent its outline for further analysis.
2. IfcWall and IfcWallStandardCase: Vertical boundary elements are treated similarly, though their geometry often represents narrow, linear polygons (walls with thickness). In both cases, polygons are stored and later written to file with their associated type labels (Floor or Wall).

■ Sample Processing Code:

```
def extract_2d_shape(self, element, settings):
    try:
        shape = ifcopenshell.geom.create_shape(settings, element)
        verts = shape.geometry.verts
        points = [(verts[i], verts[i+1]) for i in range(0, len(verts), 3)]
        hull = MultiPoint(points).convex_hull
        coords = list(hull.exterior.coords)[:-1] if isinstance(hull, Polygon)...
        return [(round(x, 2), round(y, 2)) for x, y in coords]
    except:
        return []
```

Walls are processed using the same approach, and their resulting polygon geometries are appended to a separate list.

C) Unification of Floor Areas

Since a building may contain multiple slab elements defining a single continuous space, all extracted floor polygons are merged into a unified shape using Shapely's `unary_union()` function. This is essential to define a coherent walkable area for the agents in the ABM model.

```
Type,x1,y1,x2,y2,x3,y3,...,xn,yn
```

Where:

1. **Type:** The type of the boundary element, either *Floor* or *Wall*.
2. **x1, y1, ..., xn, yn:** The flattened list of 2D coordinates (projected from the 3D vertices), forming a polygonal loop.

Example CSV Rows:

```
Floor,0.0,0.0,10.0,0.0,10.0,5.0,0.0,5.0
Wall,0.0,0.0,10.0,0.0
```

This CSV format is intentionally kept simple and flexible. Each row corresponds to a single polygon describing a boundary element. In NetLogo, these rows will be parsed and used to reconstruct the environment.

■ Sample Processing Code:

```
def extract_bounding_box_2d(self, element, settings):
        try:
            shape = ifcopenshell.geom.create_shape(settings, element)
            verts = shape.geometry.verts
            points = [(verts[i], verts[i+1]) for i in range(0, len(verts), 3)]
            if not points:
                return [], 0.0
            xs = [x for x, y in points]
            ys = [y for x, y in points]
            min_x, max_x = min(xs), max(xs)
            min_y, max_y = min(ys), max(ys)
            thickness = max(max_x - min_x, max_y - min_y) * 0.1
            coords = [
```

```
                (min_x, min_y), (min_x, max_y),
                (max_x, max_y), (max_x, min_y),
                (min_x, min_y)
            ]
            coords = [(round(x, 2), round(y, 2)) for x, y in coords]
            return coords, round(thickness, 3)
        except:
            return [], 0.0
```

D) DXF Export and Roundtrip Editing for Visual Verification

DXF export and reimport extend the CSV-based pipeline by adding a visual editing layer. While CSV provides a simple structure for simulation in NetLogo, DXF makes the extracted geometries viewable and editable in CAD tools such as Autodesk AutoCAD or FreeCAD. This enables users to inspect, adjust, and correct floor and wall polygons before feeding them back into the workflow. As shown in **Figure 17**, the user interface is equipped to a button for exporting DXF file in order to visualizing the CSV coordinates, and another button for reimporting the revised DXF (if needed).

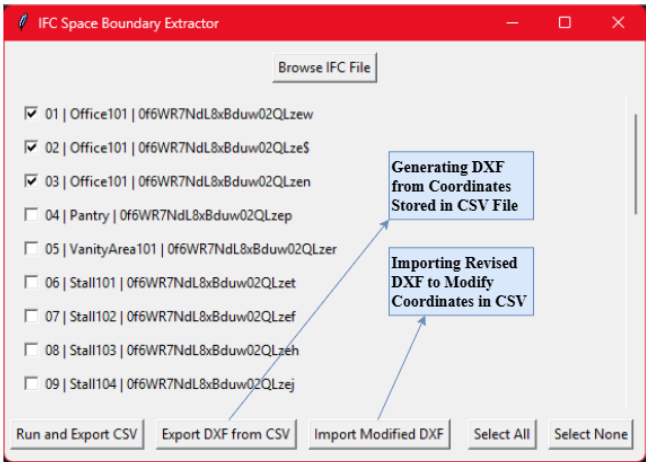


Figure 17 Coordinates Visualization and Modification Tools via DXF

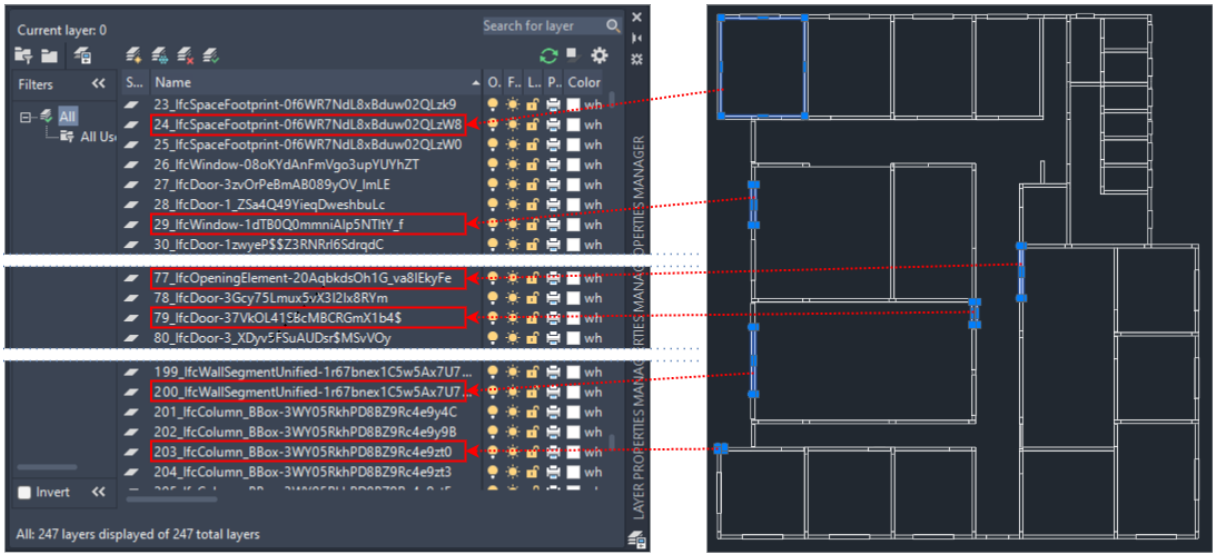
The export process reads polygon coordinates from the CSV, writes them as closed polylines in DXF, and assigns each to a layer named after its element type and identifier. Labels marking space identifiers can also be added at polygon centroids.

■ Sample Processing Code:

```
doc = ezdxf.new(dxfversion='R2010')
msp = doc.modelspace()
with open(csv_path, newline='', encoding='utf-8') as f:
```

```
    reader = csv.DictReader(f)

    for row in reader:

        coords = [(float(x), float(y)) for x, y in (pair.split(",") for pair in row["Coordinates"].split(";"))]

        if coords and coords[0] != coords[-1]:

            coords.append(coords[0])

        layer_name = f"{row['ElementType']}-{row['ElementId']}"

        msp.add_lwpolyline(coords, close=True, dxfattribs={"layer": layer_name})
```

Once modified in CAD, the DXF can be reimported, with polylines parsed to regenerate the original CSV structure. This creates a round-trip workflow in which extracted BIM boundaries are not only computationally processed but also visually reviewed, revised, and reintegrated, ensuring both accuracy and flexibility in preparing geometry for simulation. **Figure 18** demonstrates the generated DXF file and the corresponding layers for each element.



a) b)

Figure 18 Visualizing CSV Coordinates in CAD Tool (b) with Corresponding Layers (a)

Outputs from This Phase

The produced outputs of this phase are listed in the table below (Table 5):

Table 5 Outputs of IFC File Processing in Python Based on Spaces

Output File	Description
Building.ifc	The original BIM model exported from Revit in IFC4.3 format
Boundaries.csv	CSV file containing 2D projected boundary data (floors and walls)

These outputs provide the necessary foundation for the next phase of the workflow, in which the CSV data is imported into NetLogo, spatially interpreted, and used to construct a walkable movement area for agent simulation.

Importing and Visualizing Spatial Data from CSV in NetLogo

While NetLogo supports agent behaviors and patch-level manipulation, importing structured spatial data from an external CSV file requires additional scripting effort. The coding language used in NetLogo is called NetLogo itself—a domain-specific language designed specifically for agent-based modeling. This section outlines a tested method for integrating boundary definitions—specifically, walls and floors—derived from BIM into NetLogo's patch-based simulation space (Figure 19). The focus is on how to interpret and render the movement area using CSV input, preparing the environment for subsequent agent-based simulations.

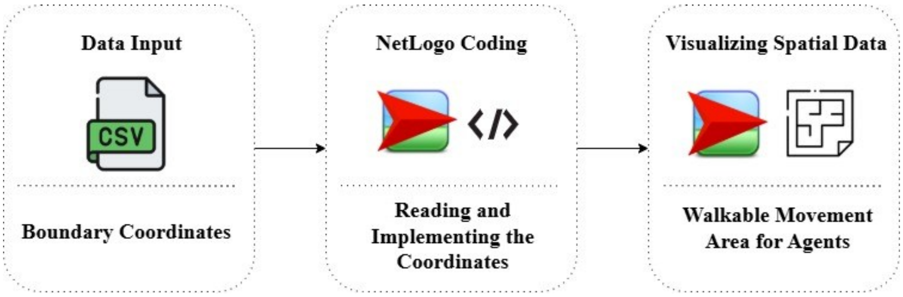


Figure 19 Visualizing CSV Boundary Data within NetLogo

A) Understanding the CSV Structure

The input file used for this task is a CSV file named `Boundaries.csv`. Each row has an element type (e.g., IfcWall, IfcDoor, IfcWindow) and a string of polygon coordinates. Coordinates are typically stored as "`x1,y1;x2,y2;x3,y3;...`".

Example row (simplified):

```
ID,Type,OtherInfo,"x1,y1;x2,y2;x3,y3;x4,y4"
1,IfcWall,...,"-15.8,12.8;-15.8,11.2;-14.2,11.2;-14.2,12.8"
```

Each row includes a Type (Floor or Wall) followed by some pairs of (x, y) coordinates, which form a polygon.

B) Preparing the NetLogo Model

Global Data Structures:

```
globals [
  polygons-data   ;; list of polygon coordinate lists
```

```
  polygon-types   ;; IFC element type for each polygon
]
turtles-own [
  vertex-index    ;; index of vertex within a polygon
  polygon-id      ;; ID of the polygon this vertex belongs to
]
```

1. `polygons-data`: stores the vertices of each polygon.
2. `polygon-types`: tracks IFC type (e.g., wall, door, window).
3. Turtles act as invisible vertices; links between turtles draw polygon edges.

C) Importing and Parsing CSV Data

1. Open file, skip header.
2. Extract IFC type and coordinate string.
3. Split coordinate string into `(x,y)` pairs.
4. Store polygon + IFC type.

■ Sample Processing Code:

```
to import-csv-polygons [filename]
  file-open filename
  let header file-read-line ;; skip header
  while [not file-at-end?] [
    let line file-read-line
    let fields parse-csv-line line
    let coords-str item 3 fields
    if (substring coords-str 0 1) = "\"" [
      set coords-str substring coords-str 1 (length coords-str - 1)
    ]
    let points []
    foreach split-coords coords-str [
      [pair] ->
      let xy split-xy pair
      let x read-from-string item 0 xy
      let y read-from-string item 1 xy
      set points lput (list x y) points
    ]
```

```
    let ifc-type item 1 fields
    set polygons-data lput points polygons-data
    set polygon-types lput ifc-type polygon-types
  ]
  file-close
end
```

D) Creating Polygons in NetLogo

For each polygon:

1. Create turtles at each vertex.
2. Connect them with links to form edges.
3. Color links based on IFC type.

■ Sample Processing Code:

```
to create-polygons
  let curr-polygon-id 0
  (foreach polygons-data polygon-types [
    [points ifc-type] ->
    let verts []
    foreach points [
      [pt] ->
      let x item 0 pt
      let y item 1 pt
      create-turtles 1 [
        setxy x y
        set polygon-id curr-polygon-id
        hide-turtle
      ]
      set verts lput turtle (max [who] of turtles) verts
    ]
    ;; color edges
    let edge-color
      (ifelse-value
        (ifc-type = "IfcDoor")   [green]
        (ifc-type = "IfcWindow") [yellow]
```

```
                              [red])
    let n length verts
    foreach n-values n [i -> i] [
      [i] ->
      let t1 item i verts
      let t2 item ((i + 1) mod n) verts
      ask t1 [ create-link-with t2 [ set color edge-color ] ]
    ]
    set curr-polygon-id curr-polygon-id + 1
  ])
end
```

E) Centering and Resizing the World

Step 1: Compute centroid of all polygons

```
to-report compute-global-centroid [polygons]
  let sum-area 0
  let sum-cx 0
  let sum-cy 0
  foreach polygons [
    [points] ->
    let result polygon-centroid-area points
    let cx item 0 result
    let cy item 1 result
    let area item 2 result
    set sum-area sum-area + area
    set sum-cx sum-cx + (cx * area)
    set sum-cy sum-cy + (cy * area)
  ]
  if sum-area = 0 [ report (list 0 0) ]
  report (list (sum-cx / sum-area) (sum-cy / sum-area))
end
```

Step 2: Shift polygons so the centroid is at (0,0)

```
to-report shift-polygons [polygons x-shift y-shift]
  report map [points ->
    map [pt -> (list ((item 0 pt) - x-shift) ((item 1 pt) - y-shift))] points
  ] polygons
end
```

Step 3: Resize the world around polygons with a margin

```
to resize-world-around-polygons
  let all-x []
  let all-y []
  foreach polygons-data [
    [points] ->
    foreach points [
      [pt] ->
      set all-x lput (item 0 pt) all-x
      set all-y lput (item 1 pt) all-y
    ]
  ]
  let margin 15
  resize-world
    floor (min all-x - margin)
    ceiling (max all-x + margin)
    floor (min all-y - margin)
    ceiling (max all-y + margin)
  set-patch-size 10
end
```

F) Visual Outcome

By running the command, as shown in **Figure 20**, all the coordinates visualize in the interface of the platform, and colorized based on their type inherited from BIM model and the script that implemented via NetLogo coding.

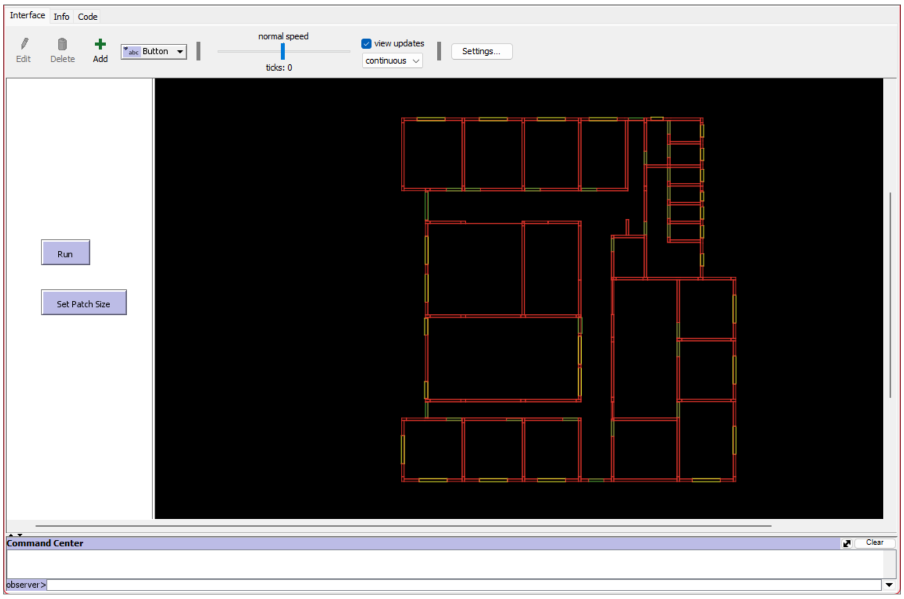


Figure 20 Generated Movement Area for Agents from Coordinates in CSV within NetLogo

1. **Red polygons** = walls
2. **Green polygons** = doors
3. **Yellow polygons** = windows
4. All polygons are shifted to be centered at `(0,0)` and scaled dynamically.
5. Edges are drawn with links; turtles remain hidden.

Discussion

This approach demonstrates how NetLogo can be extended to import and visualize structured spatial data from a CSV file. The combination of patch-based rendering, data scaling, and structured reading with the CSV extension enables researchers to use real-world layouts as movement areas of agents. This method is especially valuable in fields like crowd simulation, robotics, building modeling, and spatial analysis.

Evaluation and Conclusions from NetLogo Workflow

This phase successfully establishes a reproducible and automated pipeline to translate complex BIM geometries into a simplified 2D representation suitable for agent-based modeling. By leveraging open standards and tools, the method ensures scalability, transparency, and platform independence. The prepared CSV file serves as a structured interface between architectural models and behavior-driven simulations, facilitating advanced energy and occupant movement studies in the following stages of the study.

This workflow successfully demonstrated the feasibility of generating ABM input from BIM, but it was limited in geometric fidelity and simulation depth. NetLogo required manual configuration of movement rules and lacked native 3D support, making it less suitable for realistic occupancy modeling.

Nonetheless, this early effort served as a proof-of-concept and informed the development of the final workflow.

4.6.2 Direct Import Platforms Workflow: MassMotion

The selected ABM simulation platform was Oasys MassMotion, which provided advanced 3D simulation features and direct IFC model compatibility, making it an ideal choice for the remainder of the workflow journey.

Agent Simulation Parameters

The selected simulation case was an office building. Default agent profiles were used in MassMotion, which include empirically validated walking speeds, pathfinding logic, and collision avoidance behaviors. The Circulation tool in MassMotion defined origin and destination points within the building, enabling simulation of realistic occupant flows based on probabilistic movement patterns.

Agents in MassMotion perform movement simulations by continuously evaluating route options based on travel time, congestion, and accessibility. The simulation uses a combination of path planning and social force dynamics to simulate crowd behavior. This process results in a temporally resolved dataset describing the number of occupants present in each space at every hour.

IFC Import and Environment Setup

The Revit model was exported in IFC format, which MassMotion can ingest directly. This ensures spatial fidelity and preserves geometric detail without requiring manual processing. Within MassMotion, the IFC import generated a navigable 3D environment, automatically identifying walkable surfaces, barriers, and space boundaries (Figure 21).

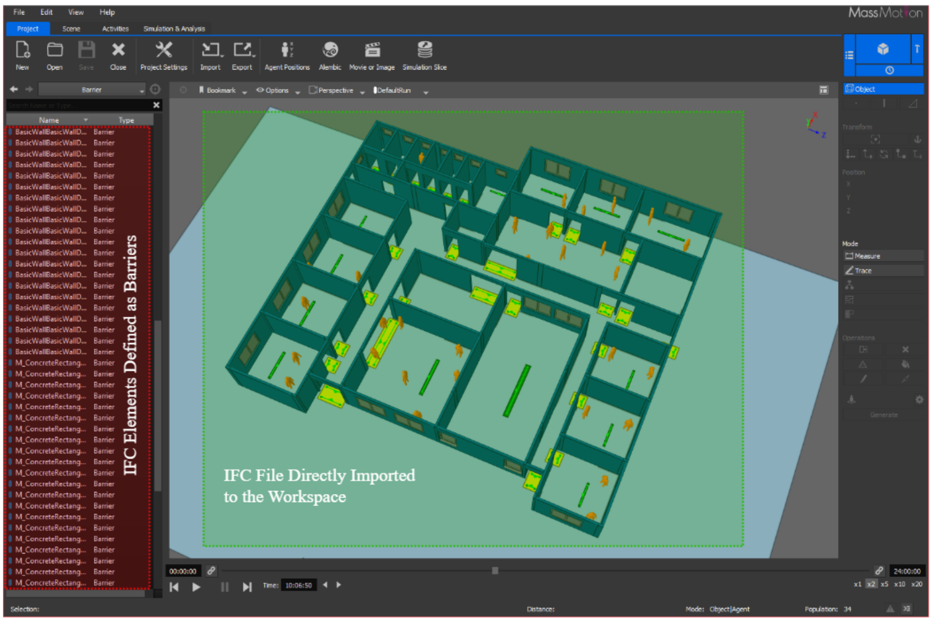
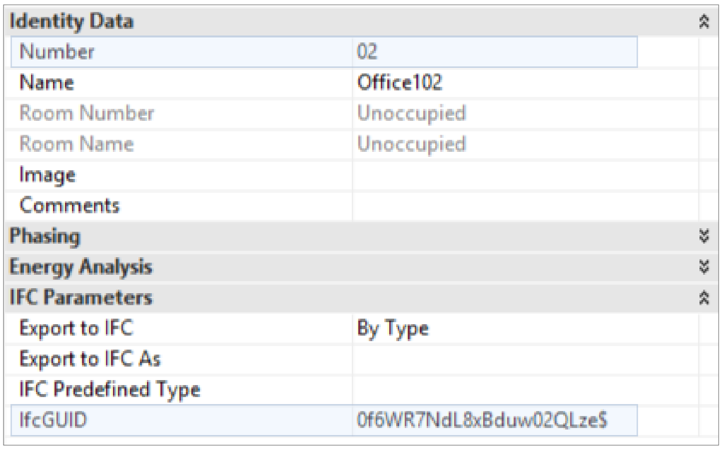


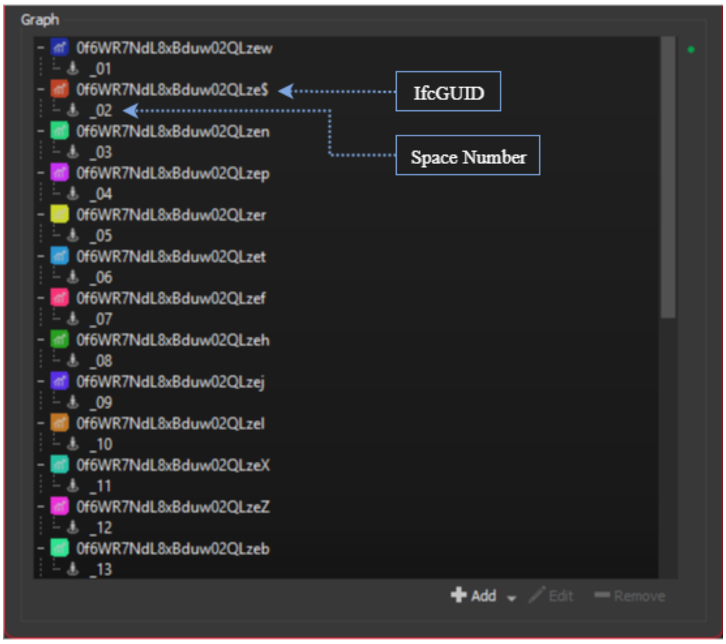
Figure 21 Direct IFC File Import into MassMotion Workspace

After importing the model:

A) Defining Destinations (Spaces/Zones): As shown in **Figure 22(a)** each space or zone is assigned a unique and consistent identifier (GUID in this study) to enable accurate mapping back into Revit. Since the imported spaces rely on Space Numbers as identifiers—which may be duplicated within the BIM model—this creates potential conflicts. To resolve this, before exporting the Population Count Graph in MassMotion, each space is stored within a group named according to its GUID (Figure 22-b). This GUID then serves as the linking reference for subsequent reintegration.



a)



b)

Figure 22 Space Identifiers in MassMotion (b) based on Space Properties in Revit (a)

For future development, the assignment of GUIDs to corresponding spaces or zones can be automated through scripting, which is essential in large-scale projects to eliminate the need for manual identifier allocation.

B) Defining Agent Schedules and Routes: The MassMotion schedule editor is used to specify arrival times, destination sequences, and route preferences. Before applying these schedules within MassMotion, a predefined operational timetable for the building must be established (Table 6), which then serves as the basis for calibrating the simulation. The influence of this schedule becomes evident in the behavior of agents within the model.

Table 6 Operating Daily Timetable of the Case Study Building

Time Block	Permanent		Dependent					
			Service Areas				Occasional	Circulation
	Enclosed Offices	Open Offices	Printing Area	Restrooms	Pantry	Store	Conference Room	Corridor
7:00 - 12:00	Fully Occupied	Fully Occupied	Light Use	Regular Use	Coffee Break	Light Use	Scheduled/As Needed	Transit
12:00 - 13:00	Partially Occupied	Partially Occupied	Not in Use	Peak	High Use (Lunch)	Not in Use	Scheduled/As Needed	Transit
13:00 - 17:00	Fully Occupied	Fully Occupied	Moderate Use	Regular Use	Some Use	Light Use	Scheduled/As Needed	Transit

C) Simulating a Full Occupancy Day: The simulation is executed over a standard working day (07:00–17:00), using the number of occupants assigned to each permanent space or zone as the basis for agent population. In MassMotion, all agent entries, exits, and transitions across spaces are recorded. In this study, nine **enclosed offices** and two **open offices** are designated as the primary spaces, with all dependent spaces governed by the movement patterns of the occupants in these main spaces. The result of agents counts simulation is illustrated in **Figure 23**.

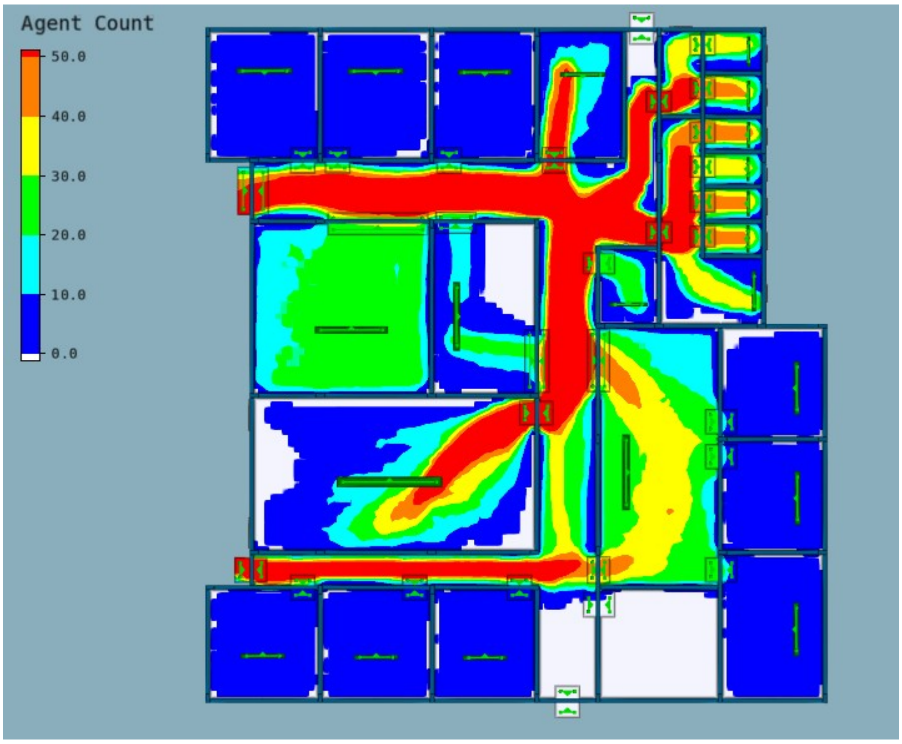


Figure 23 Agents Counts Spatial Heatmap Generated by MassMotion

Exporting the Occupancy Schedule

Upon completion of the simulation, MassMotion exported a CSV file (Table 7) containing hourly occupancy data per space. Each row of the CSV represented a space identifier and a corresponding vector of 24 values representing the number of occupants during each hour of a typical day. This CSV served as the primary input for reintegration into the BIM environment.

Table 7 Sample Rows from the CSV Dataset used for Hourly Occupancy Count

Start time	End time	0f6WR7NdL8xBduw02QLzew	0f6WR7NdL8xBduw02QLze$	0f6WR7NdL8xBduw02QLzen	...
…	…	…	…	…	
5:00:00	6:00:00	0	0	0	
6:00:00	7:00:00	0	0	0	
7:00:00	8:00:00	1.62472	1.47583	1.46278	
8:00:00	9:00:00	1.875	1.88694	1.75583	
9:00:00	10:00:00	1.74972	1.7475	1.55778	
10:00:00	11:00:00	1.85417	1.82806	1.82861	
…	…	…	…	...	

The CSV serves as a key input file for:

1. Updating occupancy parameters in Revit via Python API and pyRevit.
2. Overriding fixed assumptions in energy simulations.
3. Enabling data-driven feedback loops between simulated behavior and BIM performance analysis.

4.7 Data Reintegration with pyRevit

Since Revit does not support direct re-import of modified schedules, a custom add-in was developed using pyRevit and IronPython to automate the reintegration process (Figure 24). This implementation focused on integrating externally sourced occupancy schedules—formatted as CSV files—into Autodesk Revit, employing a structured scripting workflow to enable schedule parsing, normalization, mapping to MEP spaces via IfcGUIDs, and creation of corresponding occupancy schedule elements within the Revit environment.

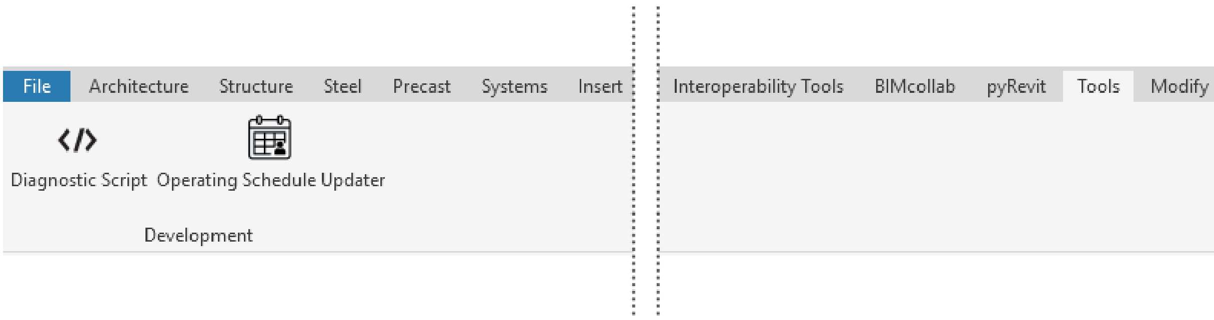


Figure 24 Custom Add-in Developed with pyRevit and Implemented Using IronPython Scripting

4.7.1 Data Reintegration Workflow

The overarching goal was to automate the process of linking space-specific occupancy data (hourly people counts) from an external CSV to the relevant elements within a Revit MEP model. As outlined in **Figure 25**, this workflow was broken down into the following sub-goals:

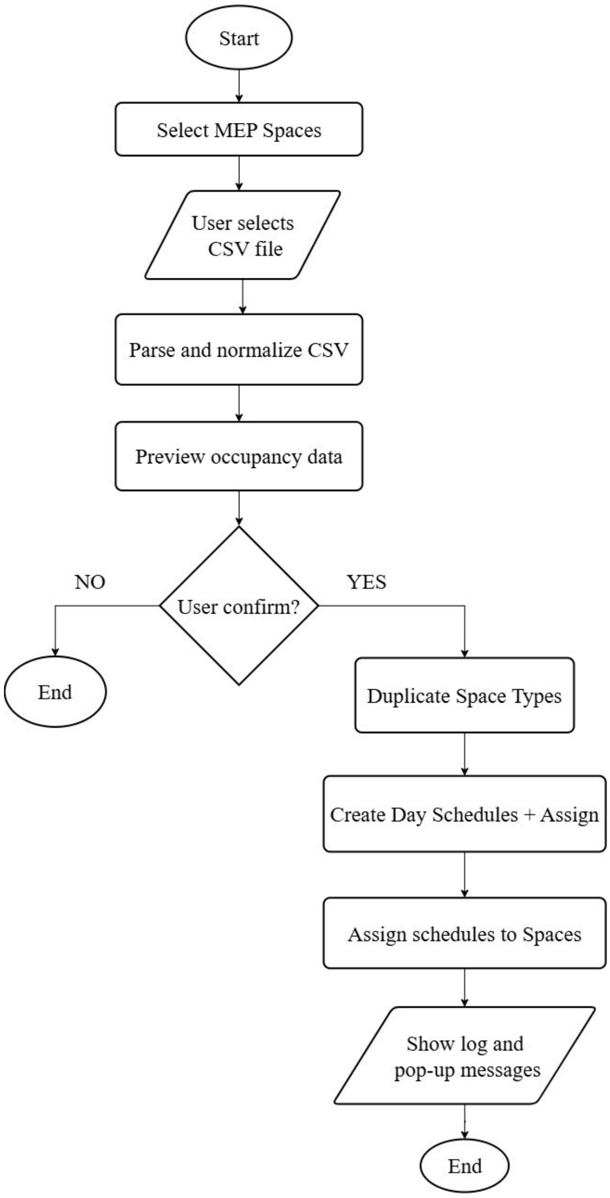


Figure 25 Flowchart of Data Reintegration with pyRevit

1. Read and normalize occupancy schedule data from a CSV file.
2. Match this data with MEP Spaces in Revit using IfcGUIDs.
3. Visually present schedule data for user-selected spaces.
4. Prompt the user to optionally duplicate the Space Types and associated Occupancy Schedules.
5. Programmatically create a new Day Schedule per space and assign it to the new Year Schedule.
6. Assign the updated schedule and space type back to each MEP Space.

4.7.2 Key Functional Components

A) CSV Data Parsing and Normalization

The implementation begins by parsing CSV files containing occupancy data, where each column represents an MEP space identified by its IfcGUID. The script is designed to exclude metadata and non-data rows (in this case, the first ten lines), ensuring that only the occupancy values are retained. Normalization is then applied by scaling each hourly value as a percentage relative to the maximum observed for that space, thereby allowing for consistent comparison across multiple space types.

■ Sample Processing Code:

```
def load_occupancy_schedule(csv_path):
    schedule = {}
    max_values = {}
    raw_max_people = {}
    with open(csv_path, 'r') as f:
        reader = csv.reader(f)
        for _ in range(10):
            next(reader, None)
        header = next(reader, None)
        guid_columns = header[2:]
        column_data = {guid.strip(): [] for guid in guid_columns}
        rows = []
        for row in reader:
            if len(row) < 2:
                continue
            start = row[0]
            row_values = []
            for i, guid in enumerate(guid_columns):
                value = row[i + 2] if i + 2 < len(row) else ""
                try:
                    num = float(value)
                except:
                    num = 0.0
                guid_clean = guid.strip()
                column_data[guid_clean].append(num)
```

```
                row_values.append(num)
            rows.append((start, row_values))
        for guid in guid_columns:
            clean = guid.strip()
            max_val = max(column_data.get(clean, [0])) or 1.0
            max_values[clean] = max_val
            raw_max_people[clean] = int(math.ceil(max_val))
        for row in rows:
            start, values = row
            for i, guid in enumerate(guid_columns):
                clean = guid.strip()
                raw_value = values[i]
                max_val = max_values[clean]
                norm = (raw_value / max_val) * 100.0 if max_val != 0 else 0
                rounded = int(round(norm))
                schedule.setdefault(clean, []).append((start, rounded))
    return schedule, raw_max_people
```

B) Space Selection via GUI

The script integrates a user interface through forms module of pyRevit, enabling the user to select relevant MEP spaces interactively. This selection mechanism filters out unplaced or zero-area spaces, ensuring that computational resources are directed exclusively toward meaningful building elements.

■ Sample Processing Code:

```
selection = forms.SelectFromList.show(
    sorted(label_map.keys()),
    multiselect=True,
    title="Select MEP Spaces",
    button_name="Load CSV File"
)
```

As shown in **Figure 26**, all the valid spaces are extracted from BIM model with their basic identifications such as Space Number, Space Name, and GUID. Presenting these data, helps the user to find the desired spaces, specially within large-scale models containing higher number of spaces.

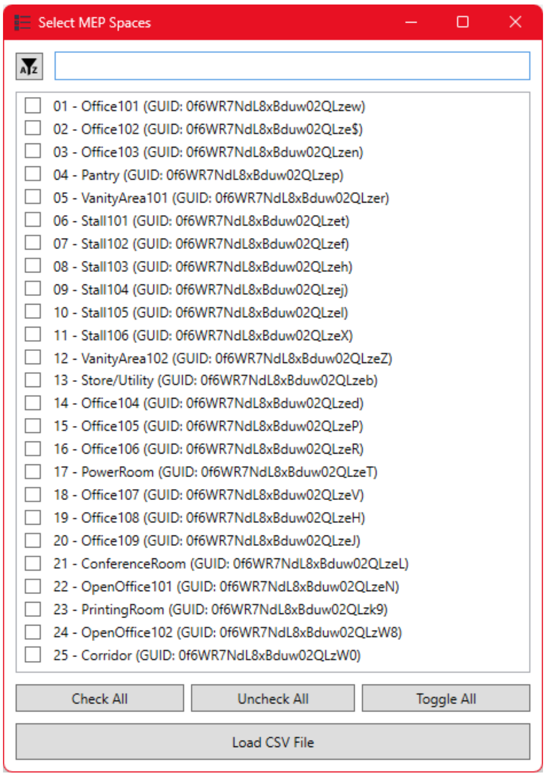


Figure 26 User Interface for Selecting Valid MEP Spaces Generated by IronPython

C) Space Labeling and GUID Mapping

Each space is labeled with its Space Number, Name, and IfcGUID. The retrieval of the IfcGUID is robust, attempting first the built-in parameter and then falling back to a custom parameter if the first is unavailable. This redundancy addresses potential inconsistencies in model authoring practices.

■ Sample Processing Code:

```
def get_ifc_guid(space):
    param = space.get_Parameter(DB.BuiltInParameter.IFC_GUID)
    if param and param.HasValue:
        return param.AsString()
    param = space.LookupParameter("IfcGUID")
    if param and param.HasValue:
        return param.AsString()
    return None
```

D) Previewing Occupancy Data

To support user validation of imported data, the script converts time strings into human-readable AM/PM format. This step facilitates interpretability during the preview stage, prior to committing changes to the model.

■ Sample Processing Code:

```
def format_time_am_pm(time_str):
    from datetime import datetime
    try:
        t_obj = datetime.strptime(time_str.strip(), "%H:%M")
        hour = t_obj.hour
        minute = t_obj.minute
        suffix = "am" if hour < 12 else "pm"
        hour_fmt = 12 if hour == 0 else (hour if hour <= 12 else hour - 12)
        return "{}:{:02d} {}".format(hour_fmt, minute, suffix)
    except:
        return time_str
```

All the necessary information that extracted both from BIM model and ABM simulations are combined and presented in an interface, as shown in **Figure 27**, in order to validate peer-to-peer data integration.

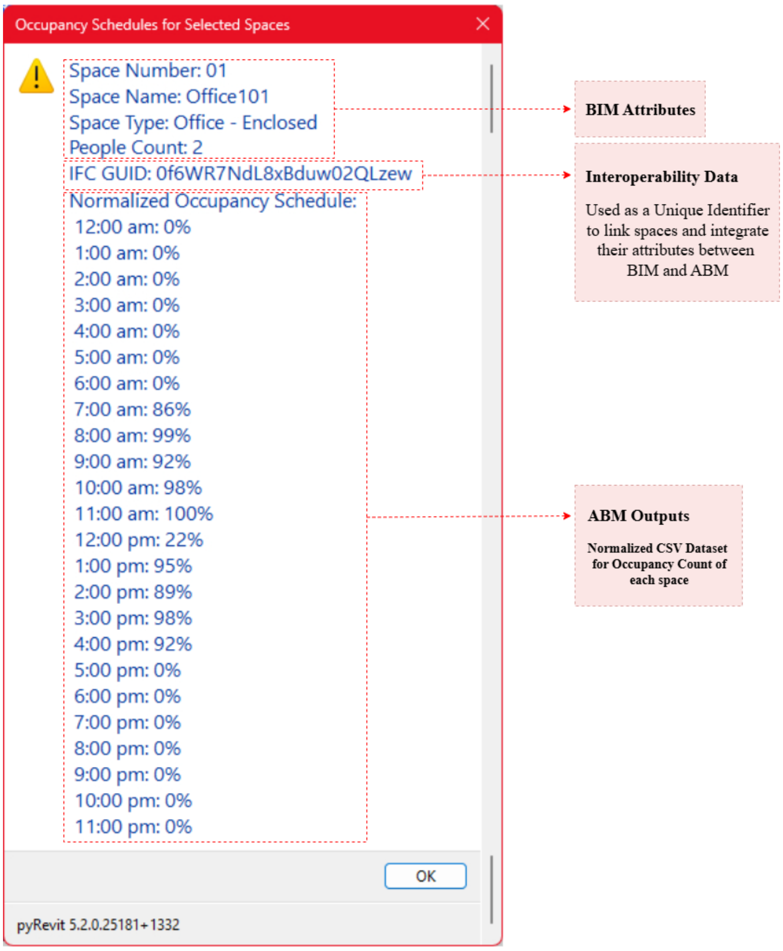


Figure 27 Previewing the Integrated Data Between BIM and ABM

E) Duplication of Space Types and Year Schedules

By user confirmation (Figure 28), all the Space Types, as well as operating schedules are being duplicated. Duplicated Space Types and their associated Year Schedules followed a naming pattern for traceability.

■ Sample Processing Code:

```
def duplicate_and_assign_space_type(space, doc, schedule_data):
    st_param = space.LookupParameter("Space Type")
    if not st_param or st_param.AsElementId() == DB.ElementId.InvalidElementId:
        return False, "Space Type parameter not assigned."

    orig_type_id = st_param.AsElementId()
    orig_type_elem = doc.GetElement(orig_type_id)
    if not orig_type_elem:
        return False, "Original Space Type element not found."

    number_param = space.get_Parameter(DB.BuiltInParameter.ROOM_NUMBER)
    name_param = space.get_Parameter(DB.BuiltInParameter.ROOM_NAME)
    number_str = number_param.AsString() if number_param else "N/A"
    name_str = name_param.AsString() if name_param else "N/A"
    new_name = "{}_{}_{}".format(orig_type_elem.Name, number_str, name_str)
```

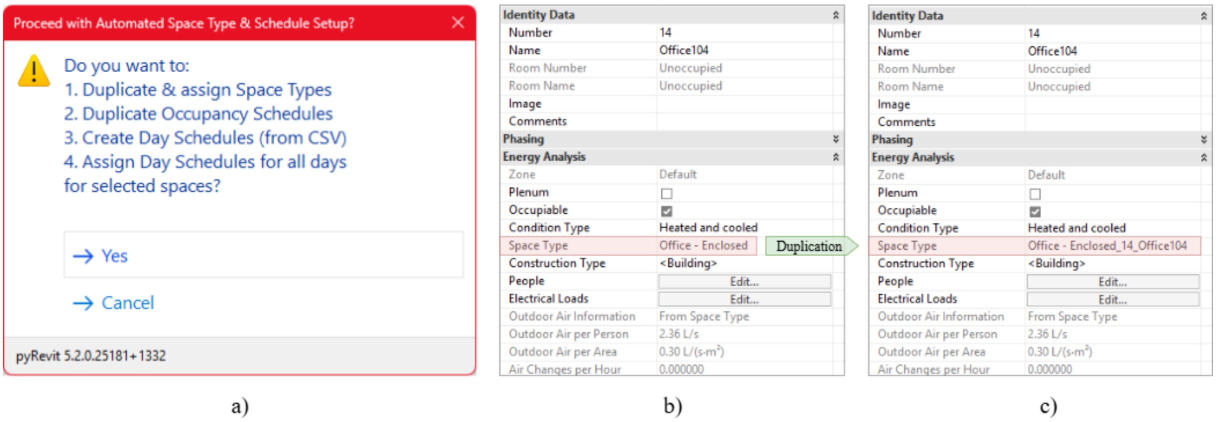


Figure 28 User Confirmation (a) Basic Space Type (b) Auto-Updated Space Type (c)

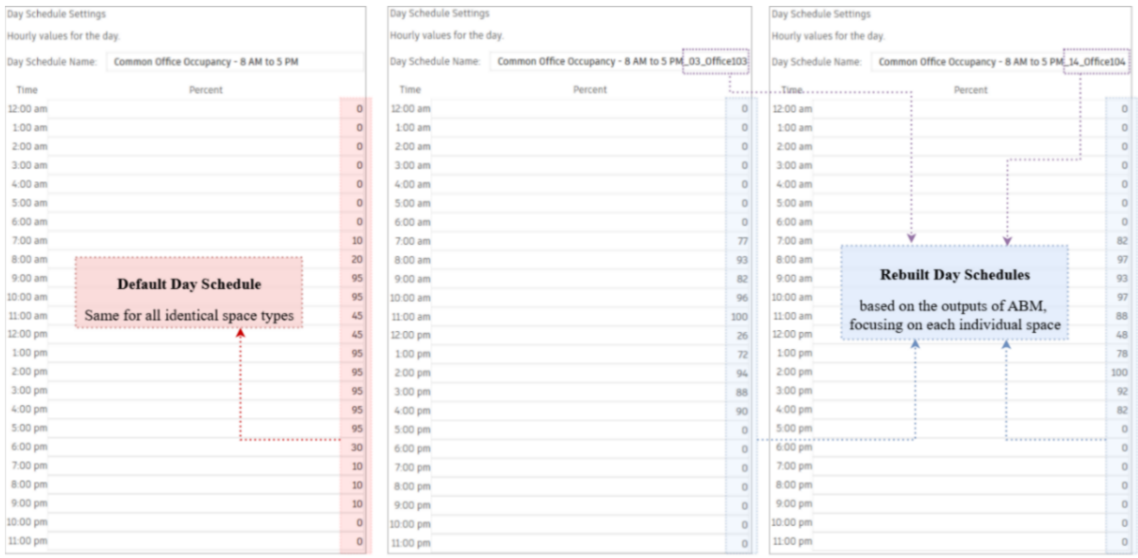
The processing code snippet shows that space type duplication is formatted as the original Space Type name, then the Space Number and Space Name, joined with dashes.

F) Creation of Day Schedules

Day schedules are constructed by iterating through each normalized hourly value and writing it into a Revit API BuildingOperatingDaySchedule object (Figure 29).

■ Sample Processing Code:

```
def create_day_schedule_with_csv_data(doc, schedule_name, csv_hourly_data):
    try:
        day_schedule = DB.Analysis.BuildingOperatingDaySchedule.Create(doc, schedule_name)
        if day_schedule and csv_hourly_data:
            for time_str, percentage in csv_hourly_data:
                hour_parts = time_str.strip().split(":")
                try:
                    hour = int(hour_parts[0])
                except:
                    continue
                if 0 <= hour <= 23:
                    usage_fraction = min(max(percentage / 100.0, 0.0), 1.0)
                    day_schedule.SetValueForHour(hour, usage_fraction)
        return day_schedule
    except Exception as e:
        print("Error creating day schedule '{}': {}".format(schedule_name, e))
        return None
```



a) b) c)

Figure 29 Day Schedules Before (a) and After ABM Outputs Implementation (b), (c)

G) Assigning Day Schedules to Year Schedules

Once created, each Day Schedule is mapped to all days in a calendar year using System.DateTime iteration (Figure 30). This method ensures solidity of occupancy assumptions all over the annual cycle.

■ Sample Processing Code:

```
def assign_day_schedule_to_year_schedule(year_schedule, day_schedule):
    if not year_schedule or not day_schedule:
        return False
    try:
        base_date = DateTime(2023, 1, 1, 0, 0, 0, DateTimeKind.Utc)
        for day_num in range(365):
            current_date = base_date.AddDays(day_num)
            try:
                year_schedule.SetScheduleForDay(current_date, day_schedule)
            except Exception as day_ex:
                print("Error assigning day schedule to day {}: {}".format(current_date.ToShortDateString(), day_ex))
        return True
    except Exception as e:
        print("Error assigning day schedule to year schedule: {}".format(e))
        return False
```

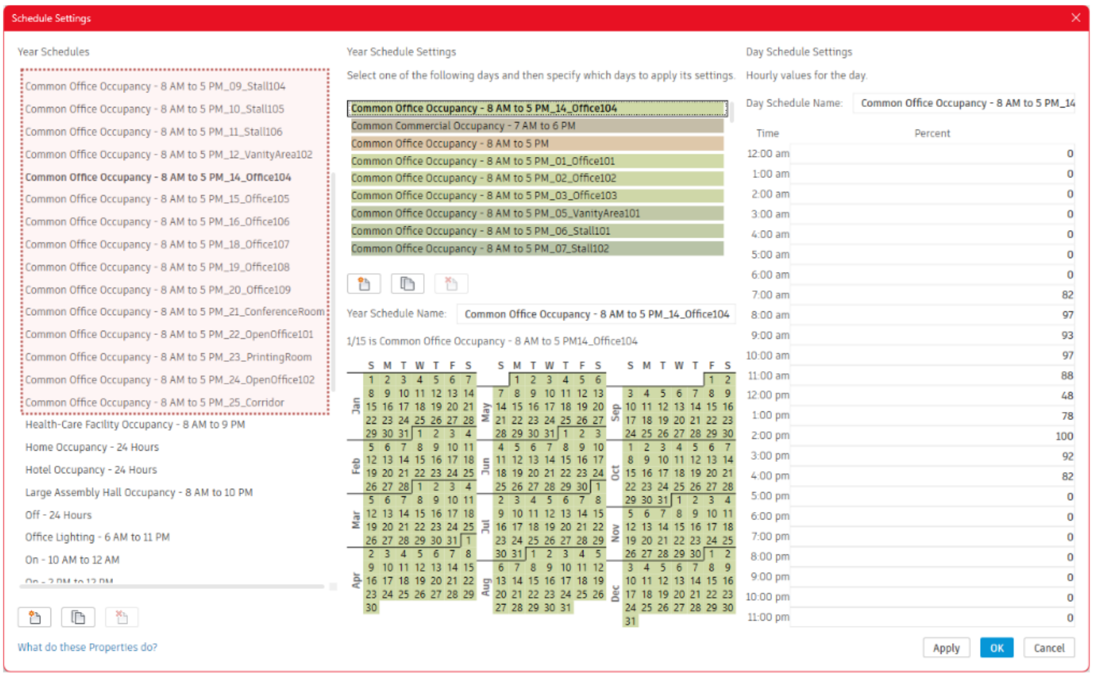


Figure 30 Newly Built and Assigned Day Schedules to Year Schedules

H) Final Assignment and Reporting

The script reassigned updated Space Types and printed results (Figure 31):

```
forms.alert("
".join(duplication_messages), title="Space Type Duplication Results", ok=True)
```

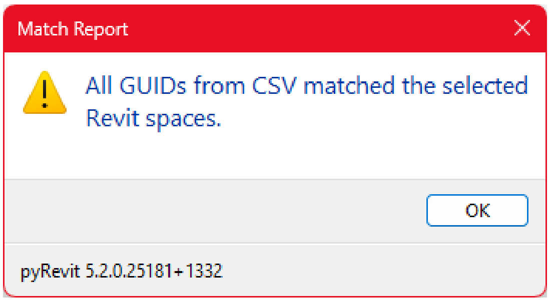


Figure 31 Final Report of Assigning Updated Data

4.7.3 Discussion

This implementation demonstrates the potential of pyRevit as an integrative platform for bridging external data sources with Building Information Modeling environments. The automation of occupancy schedule integration supports consistency in energy modeling workflows, enabling scalable analysis of space utilization. However, limitations remain, particularly in terms of API restrictions (e.g., IronPython constraints) and CSV formatting rigidity. Future development should focus on cross-platform CSV handling, the ability to export schedules for verification, and the integration of a visual analytics dashboard for improved decision support.

Future Recommendations:

1. Expand support for CSV formats from various ABM tools beyond MassMotion to improve cross-platform compatibility.
2. Add CSV export functionality for verification.
3. Create a dashboard for trend visualization.

This process enables scalable integration of occupancy models into BIM and supports further energy analysis automation.

4.8 Impact on Energy Simulation

Dynamically simulated occupancy schedules impact internal load modeling by indirectly adjusting thermal loads through changes in occupancy patterns in Carbon Insight.

Simulations using the dynamic schedule from MassMotion exhibited changes in peak and total internal gains. Spaces with variable or lower occupancy showed reduced energy demand. Conversely, an accurate representation of peak usage periods enabled better HVAC sizing and control strategies. This demonstrates the potential of integrating ABM-derived occupancy data to improve the precision of BPS.

4.8.1 Internal Simulation Comparison Workflow: Carbon Insight

After the initial energy analysis was carried out in Carbon Insight using default, fixed operating schedules embedded within the BIM model, the schedules were subsequently updated with dynamically generated data derived from the agent-based model. These revised schedules ensure that occupancy-driven variations in internal loads are accurately reflected in the model. With these updates in place, a secondary simulation was performed using Carbon Insight, and the results are demonstrated below in **Figure 32**.

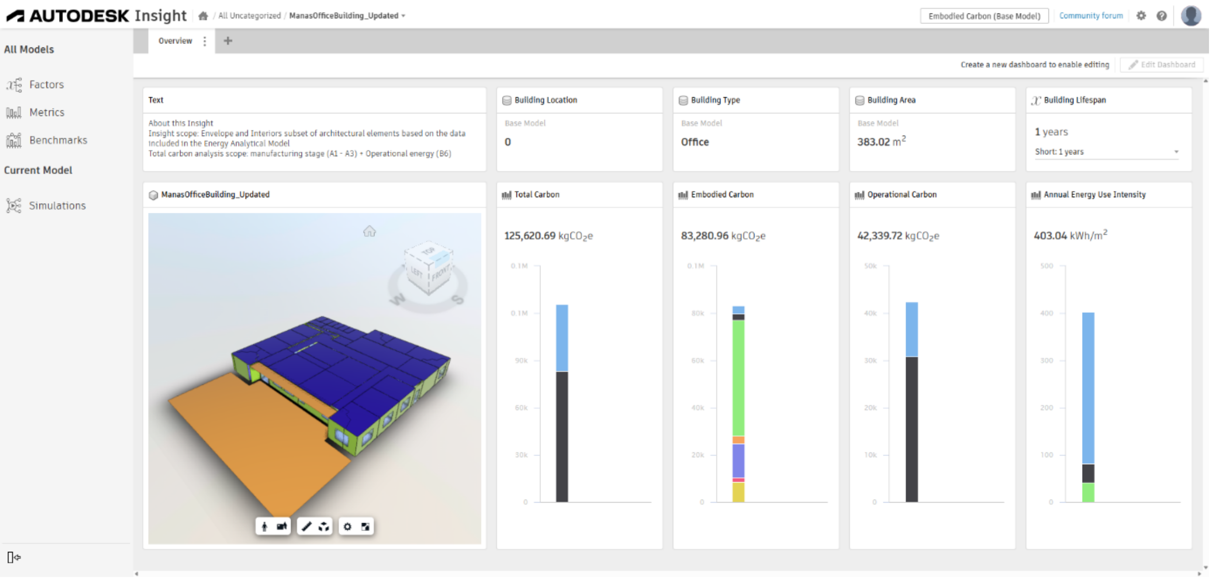


Figure 32 Secondary Energy Simulation Performed Using Revised Operating Schedules

The outputs from this updated simulation were then compared against the baseline results (Figure 13). Key performance indicators (KPI), including **Operational Carbon** and **Annual Energy Use Intensity**, were examined to highlight the differences attributable to fixed versus dynamic occupancy assumptions. This comparative analysis provided a basis for interpreting the impact of more realistic occupant behavior modeling on the accuracy of building performance simulation.

The Carbon Insight rerun with ABM-derived dynamic operating schedules produced outputs that differ from the baseline case using fixed schedules.

Table 8 Summary of KPIs for the static and dynamic occupancy scenarios

KPI	Static (Baseline)	Dynamic (ABM-derived)	Change (%)
Annual EUI (kWh/m²·year)	444.6	403.0	–9.3 %
Operational Carbon ($kgCO_2e$/year)	46,315.0	42,339.7	–8.6 %

The results clearly demonstrated the influence of incorporating time-specific, behavior-driven schedules. They not only refined the accuracy of energy demand predictions but also revealed variations in system requirements that would otherwise remain obscured under conventional assumptions. By quantifying these differences, the study underscores the benefits of integrating ABM-derived data into BIM-based simulations, illustrating how such an approach strengthens the reliability and decision-making potential of performance evaluations conducted within the BIM environment.

4.9 Extended Interoperability for Energy Simulation

A major benefit of embedding dynamic operating schedules directly within Revit is the flexibility it provides for downstream energy modeling. Once schedules are integrated as native parameters, they influence both internal simulations through Carbon Insight and external simulations via exported IFC or gbXML files.

A comparison between the gbXML files generated before and after the integration of BIM and ABM data reveals a clear limitation in the default BIM schedules. In the baseline model, schedules are applied repetitively across numerous spaces without accounting for their distinct physical attributes or accessibility patterns. Conversely, when dynamic schedules derived from ABM are introduced, the properties within the BIM model are updated, resulting in a revised gbXML file. This updated file provides a more detailed and interoperable dataset that can be effectively utilized for external energy simulations.

As illustrated in the following excerpts, the initial gbXML contains a multiply used operating schedule— `Common Office Occupancy - 8 AM to 5 PM` —applied broadly to nearly all spaces in the model, which is expected given the office building context.

■ gbXML snippet (basic):

```
</Schedule>
<Schedule id="aim0183" type="Fraction">
  <YearSchedule id="aim0184">
    <BeginDate>2023-01-01</BeginDate>
    <EndDate>2023-12-31</EndDate>
    <WeekScheduleId weekScheduleIdRef="aim0181" />
  </YearSchedule>
  <Name>Common Office Occupancy - 8 AM to 5 PM</Name>
```

After embedding dynamic schedules into the model, each space is assigned its own unique operating schedule. As a result, the number of schedules expands to match the number of individual spaces within the BIM model.

■ gbXML snippet (updated):

```
  </Schedule>
  <Schedule id="aim0183" type="Fraction">
    <YearSchedule id="aim0184">
      <BeginDate>2023-01-01</BeginDate>
      <EndDate>2023-12-31</EndDate>
      <WeekScheduleId weekScheduleIdRef="aim0181" />
    </YearSchedule>
    <Name>Common Office Occupancy - 8 AM to 5 PM_01_Office101</Name>
...
  </Schedule>
  <Schedule id="aim1505" type="Fraction">
    <YearSchedule id="aim1506">
      <BeginDate>2023-01-01</BeginDate>
      <EndDate>2023-12-31</EndDate>
      <WeekScheduleId weekScheduleIdRef="aim1503" />
    </YearSchedule>
    <Name>Common Office Occupancy - 8 AM to 5 PM_02_Office102</Name>
...
  </Schedule>
  <Schedule id="aim1602" type="Fraction">
    <YearSchedule id="aim1603">
      <BeginDate>2023-01-01</BeginDate>
      <EndDate>2023-12-31</EndDate>
      <WeekScheduleId weekScheduleIdRef="aim1600" />
    </YearSchedule>
    <Name>Common Office Occupancy - 8 AM to 5 PM_03_Office103</Name>
...
  </Schedule>
  <Schedule id="aim0697" type="Fraction">
    <YearSchedule id="aim0698">
      <BeginDate>2023-01-01</BeginDate>
      <EndDate>2023-12-31</EndDate>
      <WeekScheduleId weekScheduleIdRef="aim0695" />
```

```
    </YearSchedule>
    <Name>Common Office Occupancy - 8 AM to 5 PM_14_Office104</Name>
...
```

Unlike the manipulation of standalone gbXML files—which only affects external tools and leaves the Revit model unchanged—this method updates the core BIM data. As a result, the modifications become part of the building's operational definition, ensuring consistency and traceability across multiple simulation platforms.

This approach supports energy simulations beyond *Carbon Insight* by allowing the updated model to be imported into any tool compatible with IFC or gbXML. Therefore, project teams are not limited to one platform and can integrate ABM-enhanced schedules into a range of tools for diverse analytical needs.

5 DISCUSSIONS

5.1 Outcomes

The implementation of the proposed five-layer BIM-ABM integration workflow successfully demonstrates a functional and automated bidirectional data exchange between Autodesk Revit and Oasys MassMotion. The key outcomes of this implementation are:

Successful Technical Integration: The custom-developed pyRevit add-in effectively bridges the gap between the ABM output (CSV occupancy data) and the BIM environment. It automates the entire reintegration process, including parsing, normalization, GUID-based mapping, and the programmatic creation and assignment of dynamic Revit schedules. This automation directly addresses the research gap related to manual data transfer.

Enhanced Data Granularity for Simulation: The transition from static, fixed occupancy schedules to dynamic, space-specific profiles represents a significant leap in data quality for energy simulation. The ABM-generated schedules capture the nuanced, context-specific movement patterns of occupants, including peaks, lulls, and the interdependencies between primary and circulation spaces.

Proven Impact on Energy Modeling: The comparative analysis between the baseline and enhanced energy simulations (section 4.8) quantitatively demonstrates the framework's value. The observed differences in Annual Energy Use Intensity (EUI) and Operational Carbon emissions confirm that integrating dynamic occupant behavior data leads to more accurate and reliable building performance predictions, moving beyond the inaccuracies inherent in static assumptions.

In conclusion, the implementation phase proves that a seamless, automated BIM-ABM integration is not only theoretically possible but also practically achievable with current software and scripting capabilities. It provides a robust proof of concept that can be adopted and further refined to enhance the fidelity of building performance simulation.

5.2 Addressing Research Questions

This study is meticulously designed to provide direct answers to the research questions and to address the gaps identified in the literature review.

Question 1: "How can bidirectional data exchange between BIM and ABM platforms be achieved while maintaining data integrity and semantic consistency?".

This question is directly answered by the development and successful demonstration of the bidirectional workflow. The use of a standardized IFC file for the BIM-to-ABM transfer ensures that geometric and semantic data are maintained. The custom pyRevit and Python solution, which uses the unique IfcGUID for mapping, ensures that the dynamic data is accurately reintegrated into the correct spaces in the Revit model. Tangible evidence of this success is provided by the updated gbXML files, which show that the core BIM data has been modified to contain unique, space-specific operating schedules instead of a single, generic schedule.

Question 2: "What are the most effective approaches for automating the integration workflow to support iterative design processes?".

This is addressed by the creation of the custom, automated pyRevit add-in. The document notes that manual data transfer is a significant barrier to iterative design. The programmatic solution eliminates this manual process, allowing a designer to update the BIM model, run the ABM simulation, and reintegrate the new data with minimal effort. This automated loop turns a theoretical concept into a practical, repeatable workflow that can be easily integrated into a designer's toolkit.

Question 3: "How do ABM-generated occupancy profiles impact building energy simulation accuracy compared to traditional static schedules?".

This is answered definitively through the comparative analysis detailed in Section 4.8.1. By comparing the KPIs of the Baseline Scenario with the Dynamic Occupancy Scenario, the methodology provides quantifiable evidence that ABM-derived profiles lead to more accurate and nuanced energy simulations. The observed differences in energy consumption and operational carbon directly demonstrate the value of the framework and underscore the inaccuracies inherent in using static schedules.

6 CONCLUSION

This thesis addressed the significant research gap in bidirectional data exchange between Building Information Modeling (BIM) and Agent-Based Modeling (ABM) for enhancing building energy simulation. Traditional energy models rely on static occupancy schedules, which are a primary source of the performance gap between simulated and actual energy consumption. This research proposed, developed, and implemented a novel five-layer workflow to overcome this limitation by creating a closed-loop, automated integration framework.

The core achievement of this study is the successful demonstration of a seamless round-trip data exchange between Autodesk Revit (BIM) and Oasys MassMotion (ABM). The workflow began with the preparation of a detailed BIM model, established an energy performance baseline using static schedules, and generated dynamic, behavior-driven occupancy data through high-fidelity pedestrian simulation in MassMotion. The pivotal innovation was the development of a custom pyRevit add-in that automated the reintegration of this dynamic data back into the Revit model, updating the core occupancy parameters.

The results confirm the central hypothesis: ABM-generated occupancy profiles significantly impact energy simulation accuracy. The comparative analysis revealed measurable differences in key performance indicators (Annual EUI and Operational Carbon) between the baseline and dynamic occupancy scenario. This validates that replacing uniform, static schedules with context-specific, dynamic profiles derived from simulated occupant behavior leads to more reliable and actionable energy performance predictions. Furthermore, by embedding these dynamic schedules directly into the BIM model, the framework ensures that the enhanced data granularity is preserved for both internal (Carbon Insight) and external (gbXML/IFC export) simulation tools, greatly improving interoperability.

While this study establishes a robust foundation, future work should focus on validating and extending the framework. This includes testing its scalability across diverse building types and scales, such as hospitals or residential complexes, and conducting systematic validation against real-world occupancy data from sensors. Such efforts would be essential for transitioning BIM-ABM integration from a research concept into standard practice for high-performance building design and operation.

REFERENCES

[1] B. Motherway, “Energy Efficiency 2020,” IEA. Accessed: Aug. 31, 2025. [Online]. Available: https://www.iea.org/reports/energy-efficiency-2020

[2] R. Sacks, C. Eastman, G. Lee, and P. Teicholz, *BIM handbook: a guide to building information modeling for owners, managers, designers, engineers and contractors*, Third edition. Hoboken, New Jersey: John Wiley & Sons, Inc, 2018.

[3] E. Kamel and A. M. Memari, “Review of BIM’s application in energy simulation: Tools, issues, and solutions,” *Autom. Constr.*, vol. 97, pp. 164–180, Jan. 2019, doi: 10.1016/j.autcon.2018.11.008.

[4] Tianzhen Hong *et al.*, “Advances in research and applications of energy-related occupant behavior in buildings ☆,” *Energy Build.*, vol. 116, pp. 694–702, Mar. 2016, doi: 10.1016/j.enbuild.2015.11.052.

[5] Da Yan *et al.*, “Occupant behavior modeling for building performance simulation: Current state and future challenges:,” *Energy Build.*, vol. 107, pp. 264–278, Nov. 2015, doi: 10.1016/j.enbuild.2015.08.032.

[6] M. Sunikka-Blank and R. Galvin, “Introducing the prebound effect: the gap between performance and actual energy consumption,” *Build. Res. Inf.*, vol. 40, no. 3, pp. 260–273, June 2012, doi: 10.1080/09613218.2012.690952.

[7] J. Malik *et al.*, “Ten questions concerning agent-based modeling of occupant behavior for energy and environmental performance of buildings,” *Build. Environ.*, vol. 217, p. 109016, June 2022, doi: 10.1016/j.buildenv.2022.109016.

[8] Jared Langevin, J. Langevin, Jin Wen, J. Wen, Patrick L. Gurian, and P. L. Gurian, “Simulating the human-building interaction: Development and validation of an agent-based model of office occupant behaviors,” *Build. Environ.*, vol. 88, pp. 27–45, June 2015, doi: 10.1016/j.buildenv.2014.11.037.

[9] Yan Zhang *et al.*, “Rethinking the role of occupant behavior in building energy performance: A review,” *Energy Build.*, vol. 172, pp. 279–294, Aug. 2018, doi: 10.1016/j.enbuild.2018.05.017.

[10] A. Ciccozzi, T. de Rubeis, D. Paoletti, and D. Ambrosini, “BIM to BEM for Building Energy Analysis: A Review of Interoperability Strategies,” *Energies*, vol. 16, no. 23, p. 7845, Jan. 2023, doi: 10.3390/en16237845.

[11] M. H. Elnabawi, “Building Information Modeling-Based Building Energy Modeling: Investigation of Interoperability and Simulation Results,” *Front. Built Environ.*, vol. 6, Dec. 2020, doi: 10.3389/fbuil.2020.573971.

[12] A. Micolier, F. Taillandier, P. Taillandier, and F. Bos, "Li-BIM, an agent-based approach to simulate occupant-building interaction from the Building-Information Modelling," *Eng. Appl. Artif. Intell.*, vol. 82, pp. 44–59, June 2019, doi: 10.1016/j.engappai.2019.03.008.

[13] M. K. Bracht, A. P. Melo, and R. Lamberts, "A metamodel for building information modeling-building energy modeling integration in early design stage," *Autom. Constr.*, vol. 121, p. 103422, Jan. 2021, doi: 10.1016/j.autcon.2020.103422.

[14] T. Gerrish, K. Ruikar, M. Cook, M. Johnson, M. Phillip, and C. Lowry, "BIM application to building energy performance visualisation and management: Challenges and potential," *Energy Build.*, vol. 144, pp. 218–228, June 2017, doi: 10.1016/j.enbuild.2017.03.032.

[15] S. Durdyev, G. Dehdasht, S. R. Mohandes, and D. J. Edwards, "Review of the Building Information Modelling (BIM) Implementation in the Context of Building Energy Assessment," *Energies*, vol. 14, no. 24, p. 8487, Jan. 2021, doi: 10.3390/en14248487.

[16] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design Science in Information Systems Research".

[17] M. N. Uddin, H.-L. Chi, H.-H. Wei, M. Lee, and M. Ni, "Influence of interior layouts on occupant energy-saving behaviour in buildings: An integrated approach using Agent-Based Modelling, System Dynamics and Building Information Modelling," *Renew. Sustain. Energy Rev.*, vol. 161, p. 112382, June 2022, doi: 10.1016/j.rser.2022.112382.

[18] E. Azar and C. C. Menassa, "Agent-Based Modeling of Occupants and Their Impact on Energy Use in Commercial Buildings," *J. Comput. Civ. Eng.*, vol. 26, no. 4, pp. 506–518, July 2012, doi: 10.1061/(ASCE)CP.1943-5487.0000158.

[19] M. N. Uddin, Q. Wang, H. H. Wei, H. L. Chi, and M. Ni, "Building information modeling (BIM), System dynamics (SD), and Agent-based modeling (ABM): Towards an integrated approach," *Ain Shams Eng. J.*, vol. 12, no. 4, pp. 4261–4274, Dec. 2021, doi: 10.1016/j.asej.2021.04.015.

[20] G. Bastos Porsani, K. Del Valle de Lersundi, A. Sánchez-Ostiz Gutiérrez, and C. Fernández Bandera, "Interoperability between Building Information Modelling (BIM) and Building Energy Model (BEM)," *Appl. Sci.*, vol. 11, no. 5, p. 2167, Jan. 2021, doi: 10.3390/app11052167.

[21] M. N. Uddin, "Occupant behaviour modeling for building energy conservation : an integrated approach using agent based, system dynamics and building information modeling," Hong Kong Polytechnic University, 2022. Accessed: Aug. 26, 2025. [Online]. Available: https://theses.lib.polyu.edu.hk/handle/200/11818

[22] Yixing Chen, Y. Chen, Tianzhen Hong, T. Hong, Xuan Luo, and X. Luo, "An agent-based stochastic Occupancy Simulator," *Build. Simul.*, vol. 11, no. 1, pp. 37–49, Feb. 2018, doi: 10.1007/s12273-017-0379-7.

[23] Yixing Chen *et al.*, "Simulation and visualization of energy-related occupant behavior in office buildings," *Build. Simul.*, vol. 10, no. 6, pp. 785–798, Mar. 2017, doi: 10.1007/s12273-017-0355-2.

[24] O. Ahmed, N. Sezer, M. Ouf, L. (Leon) Wang, and I. G. Hassan, "State-of-the-art review of occupant behavior modeling and implementation in building performance simulation," *Renew. Sustain. Energy Rev.*, vol. 185, p. 113558, Oct. 2023, doi: 10.1016/j.rser.2023.113558.

[25] H. Burak Gunay, H. B. Gunay, William O'Brien, W. O'Brien, Ian Beausoleil-Morrison, and I. Beausoleil-Morrison, "A critical review of observation studies, modeling, and simulation of adaptive occupant behaviors in offices," *Build. Environ.*, vol. 70, pp. 31–47, Dec. 2013, doi: 10.1016/j.buildenv.2013.07.020.

[26] Kaiyu Sun, K. Sun, Tianzhen Hong, and T. Hong, "A framework for quantifying the impact of occupant behavior on energy savings of energy conservation measures," *Energy Build.*, vol. 146, pp. 383–396, July 2017, doi: 10.1016/j.enbuild.2017.04.065.

[27] Jessen Page *et al.*, "A generalised stochastic model for the simulation of occupant presence," *Energy Build.*, vol. 40, no. 2, pp. 83–98, Jan. 2008, doi: 10.1016/j.enbuild.2007.01.018.

[28] Diederik Aerts *et al.*, "A method for the identification and modelling of realistic domestic occupancy sequences for building energy demand simulations and peer comparison," *Build. Environ.*, vol. 75, pp. 67–78, May 2014, doi: 10.1016/j.buildenv.2014.01.021.

[29] G. Gourlis and I. Kovacic, "A holistic digital twin simulation framework for industrial facilities: BIM-based data acquisition for building energy modeling," *Front. Built Environ.*, vol. 8, Sept. 2022, doi: 10.3389/fbuil.2022.918821.

[30] A. M. Maglad *et al.*, "Bim-based energy analysis and optimization using insight 360 (case study)," *Case Stud. Constr. Mater.*, vol. 18, p. e01755, July 2023, doi: 10.1016/j.cscm.2022.e01755.

[31] M. N. Uddin, H.-H. Wei, H.-L. Chi, M. Ni, and N. Tamanna, "Building Layout Influence on Occupant's Energy Consumption Behaviour: An Agent-Based Modeling Approach," *Environ. Sci. Proc.*, vol. 15, no. 1, p. 22, 2022, doi: 10.3390/environsciproc2022015022.

[32] A. Wagner and L. O'Brien, "Occupant-Centric Building Design and Operation," International Energy Agency, Final Report EBC Annex 79, Oct. 2024. Accessed: May 05, 2025. [Online]. Available:

https://www.dropbox.com/scl/fi/g2c6humb89mgk92ns8gmw/Annex-79-final-report-and-deliverables.pdf?dl=0&e=1&rlkey=5dvnk4wzmkklebg17fruaf745

[33] Chuang Wang *et al.*, “A novel approach for building occupancy simulation,” *Build. Simul.*, vol. 4, no. 2, pp. 149–167, Dec. 2011, doi: 10.1007/s12273-011-0044-5.

[34] Yoon Soo Lee, Y. S. Lee, Ali Malkawi, and A. Malkawi, “Simulating multiple occupant behaviors in buildings: An agent-based modeling approach,” *Energy Build.*, vol. 69, pp. 407–416, Feb. 2014, doi: 10.1016/j.enbuild.2013.11.020.

[35] Jared Langevin, J. Langevin, Jin Wen, J. Wen, Patrick L. Gurian, and P. L. Gurian, “Quantifying the human–building interaction: Considering the active, adaptive occupant in building performance simulation,” *Energy Build.*, vol. 117, no. 117, pp. 372–386, Apr. 2016, doi: 10.1016/j.enbuild.2015.09.026.

[36] Clinton J. Andrews *et al.*, “Designing Buildings for Real Occupants: An Agent-Based Approach,” vol. 41, no. 6, pp. 1077–1091, Nov. 2011, doi: 10.1109/tsmca.2011.2116116.

[37] Zhikun Ding *et al.*, “Agent-based model for simulating building energy management in student residences,” *Energy Build.*, vol. 198, pp. 11–27, Sept. 2019, doi: 10.1016/j.enbuild.2019.05.053.

[38] H. C. Putra, C. J. Andrews, and J. A. Senick, “An agent-based model of building occupant behavior during load shedding,” *Build. Simul.*, vol. 10, no. 6, pp. 845–859, Dec. 2017, doi: 10.1007/s12273-017-0384-x.

[39] Francesca Stazi, F. Stazi, Federica Naspi, F. Naspi, Marco D’Orazio, and M. D’Orazio, “A literature review on driving factors and contextual events influencing occupants’ behaviours in buildings,” *Build. Environ.*, vol. 118, pp. 40–66, June 2017, doi: 10.1016/j.buildenv.2017.03.021.

[40] Mengda Jia *et al.*, “A systematic development and validation approach to a novel agent-based modeling of occupant behaviors in commercial buildings,” *Energy Build.*, vol. 199, pp. 352–367, Sept. 2019, doi: 10.1016/j.enbuild.2019.07.009.

[41] Yu Zhao *et al.*, “A systematic procedure to study the influence of occupant behavior on building energy consumption,” *Energy Build.*, vol. 43, no. 6, pp. 1409–1417, June 2011, doi: 10.1016/j.enbuild.2011.02.002.

[42] II Isabella Gaetani *et al.*, “A stepwise approach for assessing the appropriate occupant behaviour modelling in building performance simulation,” *J. Build. Perform. Simul.*, vol. 13, no. 3, pp. 362–377, Mar. 2020, doi: 10.1080/19401493.2020.1734660.

[43] Elie Azar, E. Azar, Christina Nikolopoulou, C. Nikolopoulou, Sokratis Papadopoulos, and S. Papadopoulos, "Integrating and optimizing metrics of sustainable building performance using human-focused agent-based modeling," *Appl. Energy*, vol. 183, pp. 926–937, Dec. 2016, doi: 10.1016/j.apenergy.2016.09.022.

[44] Q. Qiao and A. Yunusa-Kaltungo, "A hybrid agent-based machine learning method for human-centred energy consumption prediction," *Energy Build.*, vol. 283, p. 112797, Mar. 2023, doi: 10.1016/j.enbuild.2023.112797.

[45] M. N. Uddin, M. Lee, X. Cui, and X. Zhang, "Predicting occupant energy consumption in different indoor layout configurations using a hybrid agent-based modeling and machine learning approach," *Energy Build.*, vol. 328, p. 115102, Feb. 2025, doi: 10.1016/j.enbuild.2024.115102.

[46] A. Kashif, S. Ploix, J. Dugdale, and X. H. B. Le, "Simulating the dynamics of occupant behaviour for power management in residential buildings," *Energy Build.*, vol. 56, pp. 85–93, Jan. 2013, doi: 10.1016/j.enbuild.2012.09.042.

[47] A. Alfakara and B. Croxford, "Using agent-based modelling to simulate occupants' behaviours in response to summer overheating," in *Proceedings of the Symposium on Simulation for Architecture & Urban Design*, in SimAUD '14. San Diego, CA, USA: Society for Computer Simulation International, Apr. 2014, pp. 1–8.

[48] Tianzhen Hong *et al.*, "An occupant behavior modeling tool for co-simulation," *Energy Build.*, vol. 117, pp. 272–281, Apr. 2016, doi: 10.1016/j.enbuild.2015.10.033.

[49] Mengda Jia *et al.*, "From occupancy to occupant behavior: An analytical survey of data acquisition technologies, modeling methodologies and simulation coupling mechanisms for building energy efficiency," *Renew. Sustain. Energy Rev.*, vol. 68, pp. 525–540, Feb. 2017, doi: 10.1016/j.rser.2016.10.011.

[50] Jie Zhao *et al.*, "Occupant behavior and schedule modeling for building energy simulation through office appliance power consumption data mining," *Energy Build.*, vol. 82, pp. 341–355, Oct. 2014, doi: 10.1016/j.enbuild.2014.07.033.

[51] Kaiyu Sun, K. Sun, Tianzhen Hong, and T. Hong, "A simulation approach to estimate energy savings potential of occupant behavior measures," *Energy Build.*, vol. 136, pp. 43–62, Feb. 2017, doi: 10.1016/j.enbuild.2016.12.010.

[52] Z. Ni, Y. Liu, M. Karlsson, and S. Gong, "Enabling Preventive Conservation of Historic Buildings Through Cloud-Based Digital Twins: A Case Study in the City Theatre, Norrköping," *IEEE Access*, vol. 10, pp. 90924–90939, 2022, doi: 10.1109/ACCESS.2022.3202181.

[53] P Pieter-Jan Hoes *et al.*, "User behavior in whole building simulation," *Energy Build.*, vol. 41, no. 3, pp. 295–302, Mar. 2009, doi: 10.1016/j.enbuild.2008.09.008.

[54] M. Quintana, S. Schiavon, F. Tartarini, J. Kim, and C. Miller, "Cohort comfort models — Using occupant's similarity to predict personal thermal preference with less data," *Build. Environ.*, vol. 227, p. 109685, Jan. 2023, doi: 10.1016/j.buildenv.2022.109685.

[55] II Isabella Gaetani, I. Gaetani, P Pieter-Jan Hoes, P. P.-J. Hoes, Jlm Jan Hensen, and J. J. Hensen, "Occupant behavior in building energy simulation: towards a fit-for-purpose modeling strategy," *Energy Build.*, vol. 121, pp. 188–204, June 2016, doi: 10.1016/j.enbuild.2016.03.038.

[56] J. C. P. Cheng and V. J. L. Vincent J. L., "Integrating Agent-Based Human Behavior Simulation with Building Information Modeling for Building Design," *Int. J. Eng. Technol.*, pp. 473–477, 2013, doi: 10.7763/IJET.2013.V5.600.

[57] M. Afkhamiaghda, A. Mahdaviparsa, K. Afsari, and T. McCuen, "Occupants Behavior-Based Design Study Using BIM-GIS Integration: An Alternative Design Approach for Architects," in *Advances in Informatics and Computing in Civil and Construction Engineering*, I. Mutis and T. Hartmann, Eds., Cham: Springer International Publishing, 2019, pp. 765–772. doi: 10.1007/978-3-030-00220-6_92.

[58] H. H. Hosamo, M. S. Tingstveit, H. K. Nielsen, P. R. Svennevig, and K. Svidt, "Multiobjective optimization of building energy consumption and thermal comfort based on integrated BIM framework with machine learning-NSGA II," *Energy Build.*, vol. 277, p. 112479, Dec. 2022, doi: 10.1016/j.enbuild.2022.112479.

[59] Y. Shen and Y. Pan, "BIM-supported automatic energy performance analysis for green building design using explainable machine learning and multi-objective optimization," *Appl. Energy*, vol. 333, p. 120575, Mar. 2023, doi: 10.1016/j.apenergy.2022.120575.

[60] L. Zhao, W. Zhang, and W. Wang, "BIM-Based Multi-Objective Optimization of Low-Carbon and Energy-Saving Buildings," *Sustainability*, vol. 14, no. 20, p. 13064, Jan. 2022, doi: 10.3390/su142013064.

[61] D. H. Dorrah and M. Marzouk, "Integrated multi-objective optimization and agent-based building occupancy modeling for space layout planning," *J. Build. Eng.*, vol. 34, p. 101902, Feb. 2021, doi: 10.1016/j.jobe.2020.101902.

[62] V. Pereira, J. Santos, F. Leite, and P. Escórcio, "Using BIM to improve building energy efficiency – A scientometric and systematic review," *Energy Build.*, vol. 250, p. 111292, Nov. 2021, doi: 10.1016/j.enbuild.2021.111292.

[63] M. Alhammad, M. Eames, and R. Vinai, “Enhancing Building Energy Efficiency through Building Information Modeling (BIM) and Building Energy Modeling (BEM) Integration: A Systematic Review,” *Buildings*, vol. 14, no. 3, p. 581, Mar. 2024, doi: 10.3390/buildings14030581.

[64] E. Iran-Nejad, “pyRevit,” pyrevitlabs on Notion. Accessed: Aug. 27, 2025. [Online]. Available: https://pyrevitlabs.notion.site/

[65] E. Frits, “Create Interactive Revit Reports with pyRevit Linkify.” Accessed: Aug. 27, 2025. [Online]. Available: https://learnrevitapi.com/newsletter/create-interactive-revit-reports-with-pyrevit-linkify

[66] M. ElSayed, J. Shultz, and J. Kurtz, “User-friendly AI-driven automation for rapid building energy model generation,” *Energy Build.*, vol. 345, p. 116092, Oct. 2025, doi: 10.1016/j.enbuild.2025.116092.

[67] S. Tisue and U. Wilensky, “NetLogo: A simple environment for modeling complexity,” in *Proceedings of the International Conference on Complex Systems*, Boston, May 2004.

[68] U. Wilensky, “NetLogo.” Accessed: Aug. 27, 2025. [Online]. Available: https://ccl.northwestern.edu/netlogo/

[69] IfcOpenShell Community, “Introduction - IfcOpenShell 0.8.3 documentation.” Accessed: Aug. 27, 2025. [Online]. Available: https://docs.ifcopenshell.org/introduction.html?utm_source=chatgpt.com

[70] T. Krijnen and J. Beetz, “An IFC schema extension and binary serialization format to efficiently integrate point cloud data into building models,” *Adv. Eng. Inform.*, vol. 33, pp. 473–490, Aug. 2017, doi: 10.1016/j.aei.2017.03.008.

[71] M. Nyme Uddin *et al.*, “Occupant centric energy renovation strategy for hospital and restaurant building envelop using distinct modellingtools: A case study from low-income cultural context,” *Energy Build.*, vol. 272, p. 112338, Oct. 2022, doi: 10.1016/j.enbuild.2022.112338.

[72] “MassMotion Help Guide.” 2024. [Online]. Available: https://www.oasys-software.com/wp-content/uploads/2024/07/Oasys-MassMotion-User-Guide-11-8.pdf