A Configuration Model for Hospital Design Support Systems

Jia, Z.; Nourian, Pirouz; Luscuere, P; Wagenaar, C.

DOI
10.3390/buildings15020163

Publication date
2025

Document Version
Final published version

Published in
Buildings

Citation (APA)
Jia, Z., Nourian, P., Luscuere, P., & Wagenaar, C. (2025). A Configuration Model for Hospital Design Support Systems. *Buildings*, *15*(2), Article 163. https://doi.org/10.3390/buildings15020163

buildings

MDPI

Article

A Configuration Model for Hospital Design Support Systems

Zhuoran Jia [1,*], Pirouz Nourian [2], Peter Luscuere [1] and Cor Wagenaar [3]

[1] Faculty of Architecture and the Built Environment, Delft University of Technology, Julianalaan 134, 2628 BL Delft, The Netherlands; pluscuere@gmail.com

[2] Faculty of Geo-Information Science and Earth Observation, University of Twente, Hallenweg 8, 7522 NH Enschede, The Netherlands; p.nourian@utwente.nl

[3] Faculty of Arts, University of Groningen, Oude Boteringestraat 34, 9712 GK Groningen, The Netherlands; c.wagenaar@rug.nl

* Correspondence: z.jia@tudelft.nl

Abstract: Hospital layout significantly influences hospital service quality, demanding robust tools for informed decision-making during the layout design stage. This study presents a novel Hospital Configuration Model as the foundational component of a Hospital Design Support System, which utilizes simulation modeling to provide evaluation mechanisms on hospital efficiencies and functionalities. The Hospital Configuration Model integrates four critical data types—geometric, topological, semantic, and operational—into a machine-readable digital twin, enabling comprehensive spatial and procedural analyses. The Hospital Configuration Model facilitates simulation modeling to optimize hospital layouts and predict performance metrics such as crowdingness, patient waiting times, patient walking distance, and difficulty in wayfinding. In conclusion, the Hospital Configuration Model is the core and foundation of developing the Hospital Design Support System for evaluating hospital functionalities and efficiencies, and the potential applications of the model include digital twin development, facility management, and safety enhancement. Future research directions should, in particular, include developing the proposed Hospital Design Support System and establishing a standard way of extracting hospital operational information into an industry-standard data model.

Keywords: decision support systems; configuration model; digital twin; hospital layout design

1. Introduction

Studies have shown that the layout of a hospital has great impacts on its functionality. For instance, Jia et al. [1] summarized the problems and challenges caused by inappropriate hospital layout designs. Chraibi et al. [2] investigated how the layout of the operating theatres affects staff travel distance. Ulrich et al. [3] discussed how the physical layout affects patient outcomes and operational efficiency in healthcare settings. Burgio et al. [4] analyzed nursing staff behaviors in healthcare facilities, showing the influence of environmental layouts on staff interactions. Peponis et al. [5] investigated wayfinding within hospital environments, emphasizing the importance of spatial configuration in user navigation. The reasons why hospital layout has such a big influence on its functionality are twofold. Firstly, from the functionality aspect, the medical procedures inside hospitals are complex. Secondly, from the aspect of configuration, hospitals are as complex as small cities, where corridors are similar to streets and functional units are similar to different land uses in cities [1]. Hence, combining these two aspects indicates that the layout of a hospital significantly affects users' visibility and walkability inside the hospital. When architects

check for updates

Received: 26 November 2024
Revised: 3 January 2025
Accepted: 6 January 2025
Published: 8 January 2025

Citation: Jia , Z.; Nourian, P.; Luscuere, P.; Wagennar, C. A Configuration Model for Hospital Design Support Systems. *Buildings* **2025**, *15*, 163. https://doi.org/10.3390/buildings15020163

design a hospital, they are not merely making a building, but a system that can have many risks and problems if not treated carefully [1].

To improve the design of the system of the hospital, we propose to introduce an early operational insight into the hospital design process through the development of a decision support system, namely, the Hospital Design Support System (HDSS). The HDSS is intended to provide reliable and transparent assessment mechanisms for predicting the performance of different hospital layout designs. To be able to provide such assessment mechanisms, we need a configuration model, which is a layout representation model of the hospital system containing four types of information, i.e., geometric information, topological information, semantic information, and operational information. Table 1 provides a detailed explanation and examples for each type of information. The "Explanation" column describes the specific data contained within each information type, while the "Example" column illustrates these data using a Python 3.11 dictionary format.

Table 1. Examples of four types of information in a Hospital Configuration Model.

Information Type	**Explanation**	**Example**
Geometric Information	Room boundary consisting of a series of 3D points	{‘central_waiting’: [‘−20, 34, 4′, ‘−20, 29, 4′, ‘−19, 29, 4′, ‘−19, 39, 4′, ‘ −20, 34, 4′]}
Topological Information	A network graph consisting of nodes and edges	{′Graph1′: [{“node1″: {“id″: “R1″}, “node2″: {“id″: “R3″}, “edge1″:{“id″: “e1″}}]}
Semantic Information	Room name	{‘Department$Imaging’: [‘Central_waiting’]}
Operational Information	A patient journey through the hospital (a series of rooms that the patient needs to attend)	{‘patient_journey_1′: [‘Entrance/Exit’, ‘Registration’, ‘consulting’, ‘Entrance/Exit’]}

1.1. Research Problem and Questions

The **research problem** of this paper can be stated as follows:

- Despite the hospital layout having significant influences on hospital functionalities and operational efficiencies, there is a lack of robust tools for systematically assessing hospital layout designs in terms of operational efficiencies and functionalities at the layout design stage. To enable a robust tool to assess and predict hospital layout performance using simulation modeling, a Hospital Configuration Model integrating geometric, topological, semantic, and operational information is essential. This research addresses the need for a Hospital Configuration Model to serve as the core of the proposed tool, enabling evaluations of hospital layout designs to improve operational efficiency.

The research problem leads to the following **research questions**:

- Why do we need a Hospital Configuration Model?
- What information do we need in the Hospital Configuration Model?
- How to extract such information into the Hospital Configuration Model?

1.2. Related Works

Currently, there are several publications in the scientific literature devoted to the issue of extracting layout representation models from digital building models such as Building Information Models (BIM), Industry Foundation Classes (IFC) models, and Computer-Aided Design (CAD) models, etc. In particular, Diakite et al. [6] developed a tool for automatically generating IndoorGML models from the IFC model using C++20. The IndoorGML model is an industry-standard model that incorporates geometric, topological, and semantic information, it is a specialized layout representation model designed for indoor spatial analysis and navigation [6]. However, the IndoorGML files generated by this tool do not include any semantic information, which is inadequate for our research. Similarly, Intratech [7] developed a plugin for AutoCAD 2020 and Revit 2020 for extracting IndoorGML. The IndoorGML files generated by this plugin also lack semantic information, and this plugin relies on proprietary formats. Tong and Zheng [8] developed a tool for transforming IFC models to IndoorGML files using Autodesk's Revit 2020 and Dynamo 2.1.0, McNeel's Rhino 7 and Grasshopper 2.0, and Python 3.11. The IndoorGML files generated by this tool have semantic information. However, the tool only works for modularized buildings with simple geometries (i.e., rectangular-shaped rooms with four sides). Since hospitals are complex buildings with rooms and corridors of all kinds of irregular shapes, this tool is not suitable for our research. Jeong et al. [9] developed a tool for manually creating and editing IndoorGML files; however, this tool does not support generating IndoorGMLs from other existing sources. Similarly, Brincovean and Butean [10] created software for editing IndoorGML's topological data, which also lacks functionality for generating IndoorGML from other sources. Taehoon [11] developed a Java-based graphical editor for manually drawing IndoorGML's geometric and topological information; however, this tool cannot extract IndoorGML from other sources. Claridades et al. [12] developed a methodology for integrating the IndoorGML model with other 3D geometric information for supporting indoor navigation; however, the proposed methodology does not support the generation of IndoorGML files. Yuan and Schneider [13] constructed an indoor network model integrating geometric information for supporting indoor navigation; however, the model is not built in the industry-standard IndoorGML format, which reduces the data interoperability. Teo and Cho [14] developed a methodology for extracting geometric network models from BIM models for various indoor and outdoor route planning applications; however, their output also lacks compatibility with IndoorGML standards, which reduces the data interoperability. Khan et al. [15] developed an approach for transforming IFC files into CityGMLs (i.e., an Open Geospatial Consortium (OGC) standard designed for the representation, storage, and exchange of 3D urban spatial data [16]) and subsequently to transform CityGMLs into IndoorGML files. This approach cannot generate IndoorGMLs directly from IFC files; it can only generate CityGMLs from IFC files and then convert CityGMLs into IndoorGMLs, which is a cumbersome and complicated process. Srivastava et al. [17] developed a methodology for extracting IndoorGMLs from CAD drawings. While this methodology works for CAD models, it does not work for BIM/IFC models. CAD models are primarily 2D geometric drawings and often lack semantic information. Generating IndoorGML from CAD requires manual mapping to infer relationships and semantics, which can be error-prone and incomplete. Hashim et al. [18] developed a workflow for converting point cloud data into the Sketchup model and extracting IndoorGMLs from the Sketchup model. Unlike BIM models, SketchUp models do not inherently include rich semantic information; hence, the extracted IndoorGML files also lack semantic information. To summarize, the current gap in the literature lies in the absence of a tool capable of handling complex buildings with irregular shapes while generating IndoorGML files enriched with semantic information. Additionally, no existing tool can convert IndoorGML files into Hospital Configuration Models (HCMs) that integrate operational data and evaluate the alignment between operational needs and the spatial configuration of hospitals. This study addresses these gaps by proposing a novel methodology for the semi-automatic generation of IndoorGML models from BIM/IFC models and the subsequent conversion of these IndoorGML models into Hospital Configuration Models (HCMs).

1.3. Contributions and Novelties

This research introduces a novel framework for supporting hospital design by proposing the Hospital Configuration Model (HCM) as the foundational component of a Hospital Design Support System (HDSS). The contributions and novelties of this research can be summarized as follows:

- **Generation of IndoorGMLs:**
  Our proposed methodology incorporates a tool for the semi-automatic generation of IndoorGML files from widely used BIM/IFC models and the automated conversion of these IndoorGML files into Hospital Configuration Models (HCMs). Our tool is specifically designed to handle complex buildings with irregular shapes, ensuring that the resulting IndoorGML files contain accurate and comprehensive semantic information.
- **Development of the Hospital Configuration Model (HCM):**
  The HCM integrates four critical data types—geometric, topological, semantic, and operational—into a comprehensive, machine-readable digital twin model. By bridging spatial information with operational workflows, the HCM ensures that hospital layouts are evaluated not only for spatial efficiency but also for their alignment with medical procedures and operational needs.
- **Construction of Activity Relations Chart (ARC) Models:**
  This study proposes a method for systematically building Activity Relations Chart (ARC) models, which can be used for modeling and optimizing hospital layouts. The ARC model is a tool for representing relationships between different spatial units within a building [19] and can be thought of as the simplified graph-theoretical equivalent of the HCMs.

The goal of the paper can be summarized as making a machine-readable model/digital twin of a hospital that can bring the operations of a hospital into a spatial information model as attributes. The rest of the paper is structured as follows: The following Section 2 called Background answers research question 1: Why a Hospital Configuration Model (HCM) is essential to have as the core of an HDSS as an information system? Then, Section 3, called Research Methodology, answers the research question 2: What specific pieces of information content are needed in an HCM, based on the arguments and reflections provided in the background? Afterwards, Section 4 answers the research question 3 of how to extract such information into the Hospital Configuration Model. We dive deeper into extracting the proposed pieces of essentially required information from what is typically available as data and information models on hospitals, both the Building Information Models or BIM data models and hospital Business Process Model and Notation, or BPMN, data models. In Section 5, called Discussion, we talk about key findings, novelties, and limitations of this research. Finally, in Section 6, called Conclusion, we introduce the implications of the HCM and propose potential future research directions.

2. Background

This section explains the necessity of developing an HCM as the core of HDSS. One of the major functions of the proposed HDSS is to run simulation modeling to simulate complex dynamic situations in a hospital environment. The simulation modeling requires a configuration model as the base where simulation can be implemented. Furthermore, another major functionality of HDSS is to ensure that the configuration of a hospital fits how the hospital is operated. In other words, the space of the hospital should be laid out in a way that serves the purpose of improving hospital operations. An HCM is therefore essential to check this fit between the hospital space and medical procedures (hospital operations).

2.1. Simulation Modeling for Ex Ante Assessment

The primary objective of developing an HDSS is to provide robust mechanisms for evaluating the potential impact of hospital configuration decisions on various quality indicators or quantitative outcomes. These outcomes may include measures of functional efficiency, levels of crowding, and other relevant performance metrics. The basic idea for devising such assessment mechanisms is developing spatial analysis procedures based on spatial queries, which tend to be about spatial networks and traversals on their graph models. However, in the bigger scheme of configurational assessment, some inherently dynamic phenomena can only be properly understood through simulation modeling. There are multiple paradigms and at least two simulation modeling approaches that are commonly used for the study of complex systems. Complex system modeling is the bigger picture in which the whole case of making an HDSS is considered because hospitals are obviously complex socio-technical organizations that are not only complex from a spatial point of view but also from an organizational and operational point of view. It is non-trivial to understand how they work, let alone to be able to come up with recommendations as to how their functionality or operational efficiency can be improved. Thus, we must look into the bigger picture of simulation modeling for understanding hospitals as complex systems to approach the daunting task of HDSS development. To keep the scope of the paper manageable, however, here we only look at the necessity of having a hospital configuration model from the point of view of simulation modeling paradigms and approaches (which are not all necessarily relevant to the case of HDSS development, but for the sake of generality, we mention them all). There are four **simulation modeling paradigms** [20]:

- **Discrete Event Simulation (DES)**: A Discrete-Event Simulation (DES) model is a model of a system in which events occur at specific points in time, causing changes in the system state [21]. A DES model consists of:
    - Discrete event: The discrete event is the cause of the system state change. The state of the system in the DES model only changes due to the occurrence of events [22]. For example, in a hospital DES model, the patient's walking distance in the hospital only changes if the patient moves to the next room.
    - Clock: The clock records the duration of the simulation. The DES model is dynamic as time is a critical variable, i.e., the state variables of the system change over simulation time [22]. For example, in a hospital DES model, the walking distance of the patient increases as the simulation time increases.
    - Random number generators: A random number generator can generate random variables for the DES [22], e.g., medical service time or patient inter-arrival rate.
    - Statistics: This summarizes the results of the simulation, such as patient waiting times or patient walking distances [22].
    - Ending condition: The DES ends when the ending condition is met [22], e.g., a hospital DES model is set to end when a certain number of patients are discharged.

  The proposed HDSS can use DES to simulate patients' medical procedures in hospitals and predict hospital performance by calculating performance indicators, such as people density, patient waiting time, patient walking distance, etc.
- **Agent-Based Modeling (ABM)**: An Agent-Based Model comprises individual agents, their interactions with one another, and their interactions with the surrounding environment [23]. Agents are small computer programs that represent various types of entities [23]. For example, in a hospital ABM model, agents can be patients, nurses, doctors, etc. The environment in the ABM model can be a network graph where agents can interact [23]. The agents have several characteristics which are summarized as follows:

  - Autonomy: Agents are autonomous entities that behave without guidance from central controllers; they are capable of making independent decisions [24].
  - Heterogeneity: Agents can have various features, such as ages, jobs, genders, etc. [24]. For example, in a hospital ABM model, agents can have different roles, such as patients, medical staff, technical staff, etc.
  - Active: Agents are active in an ABM model because they are goal-directed; they are assigned to different goals and they need to achieve them [24]. To achieve their goals, agents are equipped with the capacity to perceive their environment and interact with other agents. Additionally, they are enabled to make logical decisions that facilitate goal attainment [24].
  - Interactive: Agents can interact with other agents and also with the environment [24].
  - Mobility: Agents can move in the ABM environment [24]. For instance, the patient or the staff agent of a hospital ABM model can move in the environment (i.e., a graph) to achieve their goals.
  - Adaptation/Learning: Agents can be designed to be adaptive; they can alter their states based on previous states [24]. For instance, in a hospital agent-based model, a doctor agent becomes available for new patients once they have completed treatment of the current patient.

  ABM can also be applied in the proposed HDSS for studying individual behaviors, interactions between patients and staff, or patient flows in the hospital.

- **Continuous Simulation**: Continuous simulations are designed to model systems in which the system states change continuously over time. For example, in a hospital continuous simulation model, the patient's length of stay increases continuously over simulation time. Continuous simulation models use differential equations or other mathematical models for defining the changing rate of the system states over time [25]. Continuous simulations can be compared with DES, where state variables in continuous simulation models change continuously over time, while in DES models they change at distinct points in time. Continuous simulations can be used for studying the spread of a contagious airborne disease (e.g., influenza or COVID-19) throughout a hospital to understand infection risk in different areas.
- **System Dynamics**: System dynamics is a type of continuous simulation that is developed for supporting policy making in complex and dynamic systems [25]. In system dynamics models, the behavior of the system is created by the interactions between different components over time. The key components of a system dynamics model are introduced in the following:
  - Stocks: Stocks are accumulations of resources in a system, they represent the state of the system [25], e.g., the number of patients in a hospital.
  - Flows: Flows represent the changing rates of stocks over time [25]. In a hospital, for example, the flow could be the rate at which new patients are admitted or discharged.
  - Information links: In a system dynamics model, information links connect stocks with flows and transfer information from a stock to the flow; they define how a stock influences the values of the flow [25]. For example, in a hospital system dynamics model, by linking stock (i.e., number of patients in the hospital) to the flow (i.e., patient inter-arrival rate), the patient inter-arrival rate can be influenced by the current number of patients in the hospital.

  System dynamics can be applied in hospital management in terms of understanding patient flow, resource allocation, the spread of disease, etc.

And there are mainly two **simulation approaches** [26]:

- **Causal or signal-flow-based modeling as in Simulink** [27].
- **Acausal or equation-based modeling as in Modelica** [28].

Both of these simulation modeling approaches result in the construction of network models to be used in running the simulation model. However, the first type of network produced in signal flow-based simulation modeling is a Directed Acyclic Graph (DAG) that is used almost directly as a computational network model, whereas, the second type of network model produced in equation-based modeling is closer to our configuration model. It is a network model that closely resembles the physical interconnections of elements in the system. It does not, however, readily represent a computational network model. Such a model still needs to be coupled with mathematical models to be converted into a simulation model. In an equation-based simulation modeling language like Modelica, this step happens thanks to the computational engine of the language, but in our workflow, we have to consider this as a secondary step of modeling to be performed by the modeler for the domain-specific simulation tasks. However, discussing these details is beyond the scope of this paper. Instead, here, we reflect on the requirements of an HCM for being applicable and useful for building various simulation models, such as the four types of simulation models mentioned previously.

Reflecting on the property of the HCM from the point of view of simulation modeling, what should the properties of an HCM be to be ready for simulation models? According to the previous introduction of the four types of simulation models, DES is perhaps the most suitable simulation modeling paradigm for studying the operational efficiency of hospitals. However, ABM simulation models are also used for studying stressful, chaotic, extreme or urgent situations in which the human agents might behave like herds or flocks of animals, following and interacting with each other closely. Continuous simulation models and system dynamics models are less suitable for modeling hospital operations. Continuous simulation models are more applicable in specific contexts, such as studying the spread of contagious airborne diseases in hospitals, and system dynamics models are more suitable for supporting the design of hospital policies or management strategies. So, in short, DES and ABM models are the two simulation modeling paradigms that are most suitable for this research, with the ABM simulation models able to be used to assess the extraordinary working situations and the DES to assess the ordinary working situations of the hospitals.

Ideally, our proposed HCM can cater for the needs of both the simulation modeling approaches. This means that our HCM should have the essential spatial and operational pieces of information that can potentially be further elaborated automatically to extract higher resolution and more detailed information models as bases of such simulation models. For example, if you consider a base simple floor plate in an HCM, it can be further meshed into a high-res grid for running an ABM, but only if necessary. However, it is not necessary to store high-resolution detail information in the HCM at all times, as the high-resolution mesh can be generated on-demand using the essential boundary information already stored in the HCM.

2.2. Operational vs. Spatial Information

The most important reason to consider making an HDSS is to enhance the fit between the hospital's operational information and spatial information. A hospital's operational information pertains to the processes, workflows, and activities that occur within a hospital. It encapsulates the dynamic aspects of how the facility operates, focusing on the flow of people, materials, and resources through various functional units. Key elements include:

- Patient Journeys: A sequence of steps or locations that patients visit during their hospital stay; for an instance of the patient journey, please see 'Operational Information' in Table 1.
- Resource Utilization: Details of how hospital resources (e.g., rooms and beds) are allocated and used.

Spatial information focuses on the layout/configuration of the hospital. It provides a static framework that determines how operational activities are accommodated within the facility. Key elements include:

- Geometric Information: Details about the shapes, dimensions, and coordinates of physical spaces, such as room boundaries. For an example of a room boundary, please see 'Geometric Information' in Table 1.
- Topological Information: The connectivity between spatial units, represented as a network of nodes (e.g., rooms) and edges (e.g., connections between rooms). Table 1's 'Topological Information' provides an example of a simple network.

While operational information captures the dynamic aspects of hospital activities, spatial information provides the static framework that houses these activities.

The purpose of enhancing the fit between a hospital's operational information and spatial information can be expressed as ensuring that the configuration of a hospital is fit for the purpose for which it is built.

The operational steps can be related to the spatial units of a hospital—the sequences or medical procedures inevitably entail the transport of people (patients and staff), materials, and equipment inside the hospital. Therefore, the challenge of operational management of the hospital will significant involve transport planning and operations research in a spatial sense. In conclusion, it can be said that the problem of designing an optimal hospital configuration is about the fitness of the hospital configuration for undertaking the medical procedures that are supposed to take place through the spatial layout of the hospital. Figures 1–3 illustrate some representative examples of medical procedures in real-world hospitals; Figures 4 and 5 together show the ideal configuration obtained from these medical procedures. For further explanation on how to achieve an ideal hospital configuration with regard to hospital medical procedures, please see Section 3.2.

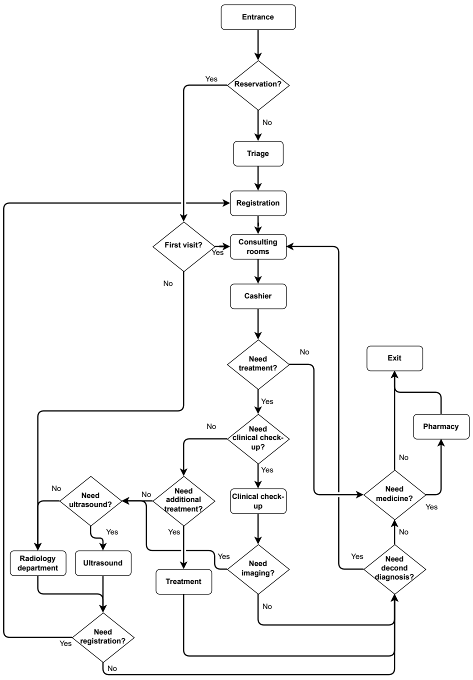


Figure 1. Patients' paths in the Outpatient Department of Panyu Central Hospital, image source: [29].

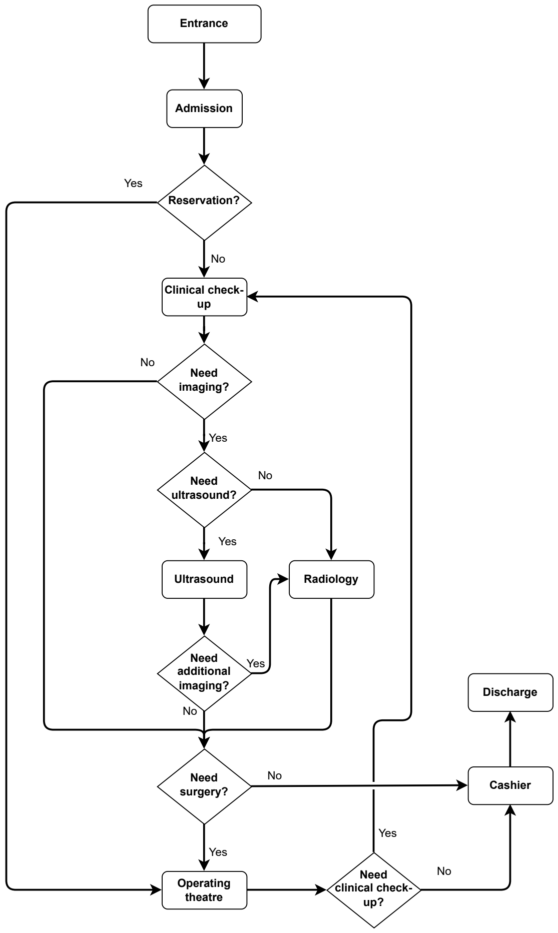


Figure 2. Patients' paths in the Inpatient Department of Panyu Central Hospital, image source: [29].

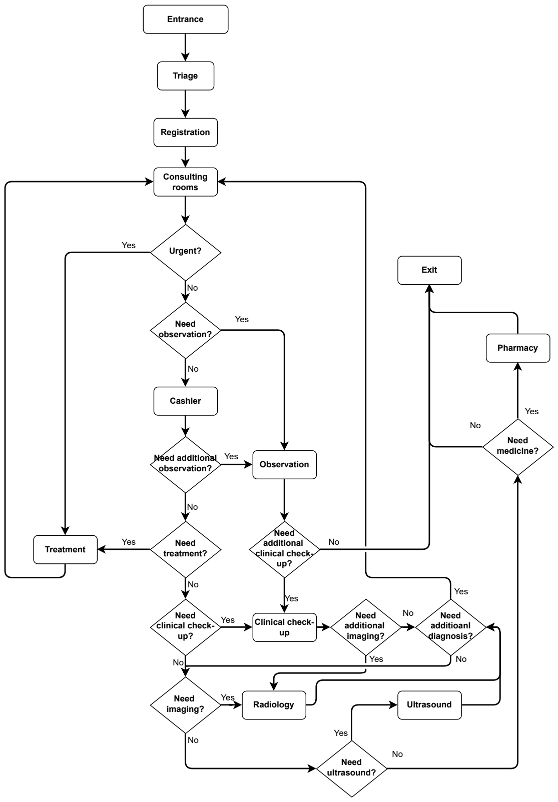
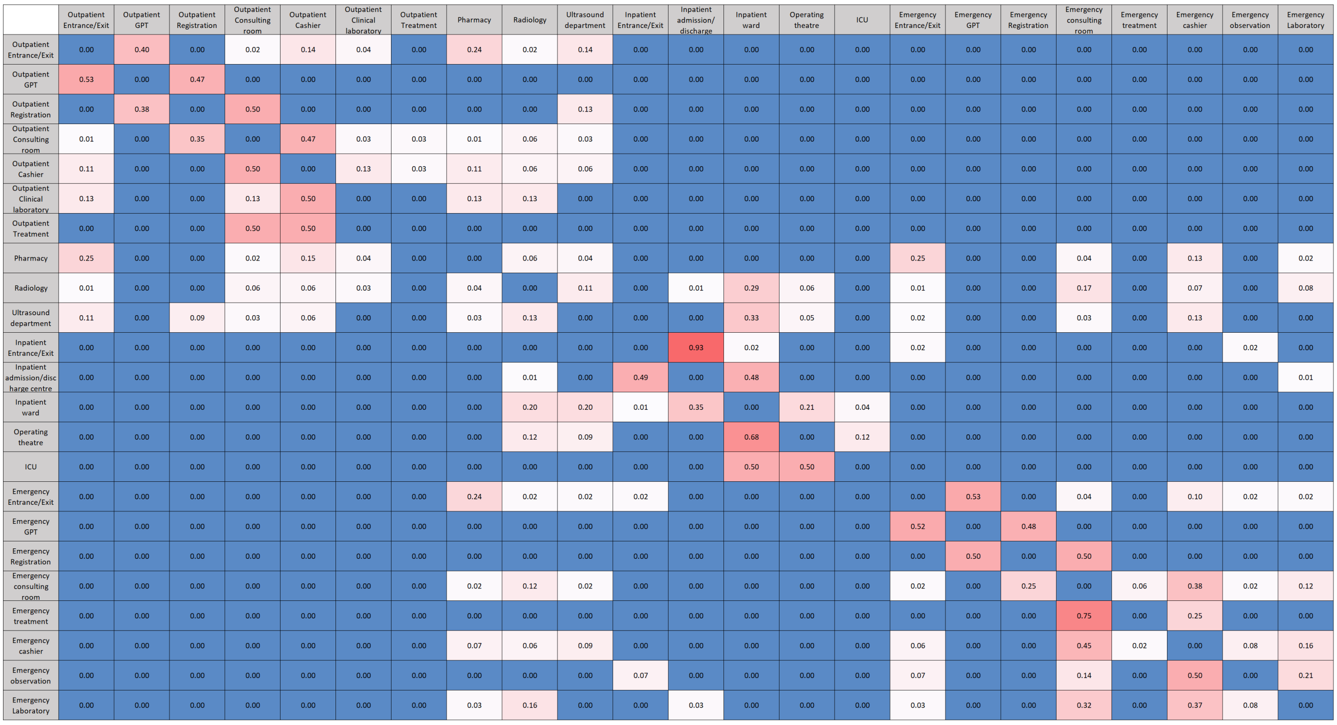


Figure 3. Patients' paths in the Emergency Department of Panyu Central Hospital, image source: [29].



	Outpatient Entrance/Exit	Outpatient GPT	Outpatient Registration	Outpatient Consulting room	Outpatient Cashier	Outpatient Clinical laboratory	Outpatient Treatment	Pharmacy	Radiology	Ultrasound department	Inpatient Entrance/Exit	Inpatient admission/ discharge	Inpatient ward	Operating theatre	ICU	Emergency Entrance/Exit	Emergency GPT	Emergency Registration	Emergency consulting room	Emergency treatment	Emergency cashier	Emergency observation	Emergency Laboratory
Outpatient Entrance/Exit	0.00	0.40	0.00	0.02	0.14	0.04	0.00	0.24	0.02	0.14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Outpatient GPT	0.53	0.00	0.47	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Outpatient Registration	0.00	0.38	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Outpatient Consulting room	0.01	0.00	0.35	0.00	0.47	0.03	0.03	0.01	0.06	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Outpatient Cashier	0.11	0.00	0.00	0.50	0.00	0.13	0.03	0.11	0.06	0.06	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Outpatient Clinical laboratory	0.13	0.00	0.00	0.13	0.50	0.00	0.00	0.13	0.13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Outpatient Treatment	0.00	0.00	0.00	0.50	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Pharmacy	0.25	0.00	0.00	0.02	0.15	0.04	0.00	0.00	0.06	0.04	0.00	0.00	0.00	0.00	0.00	0.25	0.00	0.00	0.04	0.00	0.13	0.00	0.02
Radiology	0.01	0.00	0.00	0.06	0.06	0.03	0.00	0.04	0.00	0.11	0.00	0.01	0.29	0.06	0.00	0.01	0.00	0.00	0.17	0.00	0.07	0.00	0.08
Ultrasound department	0.11	0.00	0.09	0.03	0.06	0.00	0.00	0.03	0.13	0.00	0.00	0.00	0.33	0.05	0.00	0.02	0.00	0.00	0.03	0.00	0.13	0.00	0.00
Inpatient Entrance/Exit	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.93	0.02	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.02	0.00
Inpatient admission/discharge centre	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.49	0.00	0.48	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
Inpatient ward	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.20	0.20	0.01	0.35	0.00	0.21	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Operating theatre	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.12	0.09	0.00	0.00	0.68	0.00	0.12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ICU	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Emergency Entrance/Exit	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.24	0.02	0.02	0.02	0.00	0.00	0.00	0.00	0.00	0.53	0.00	0.04	0.00	0.10	0.02	0.02
Emergency GPT	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.52	0.00	0.48	0.00	0.00	0.00	0.00	0.00
Emergency Registration	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.50	0.00	0.00	0.00	0.00
Emergency consulting room	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.12	0.02	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.25	0.00	0.06	0.38	0.02	0.12
Emergency treatment	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.75	0.00	0.25	0.00	0.00
Emergency cashier	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.07	0.06	0.09	0.00	0.00	0.00	0.00	0.00	0.06	0.00	0.00	0.45	0.02	0.00	0.08	0.16
Emergency observation	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.07	0.00	0.00	0.00	0.00	0.07	0.00	0.00	0.14	0.00	0.50	0.00	0.21
Emergency Laboratory	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.16	0.00	0.00	0.03	0.00	0.00	0.00	0.03	0.00	0.00	0.32	0.00	0.37	0.08	0.00

Figure 4. Activity relationship chart of procedures in Panyu Central Hospital, red cells indicate strong connecting relationship between two procedures, blue cells indicate the weak connecting relationship between two procedures, image source: [29].

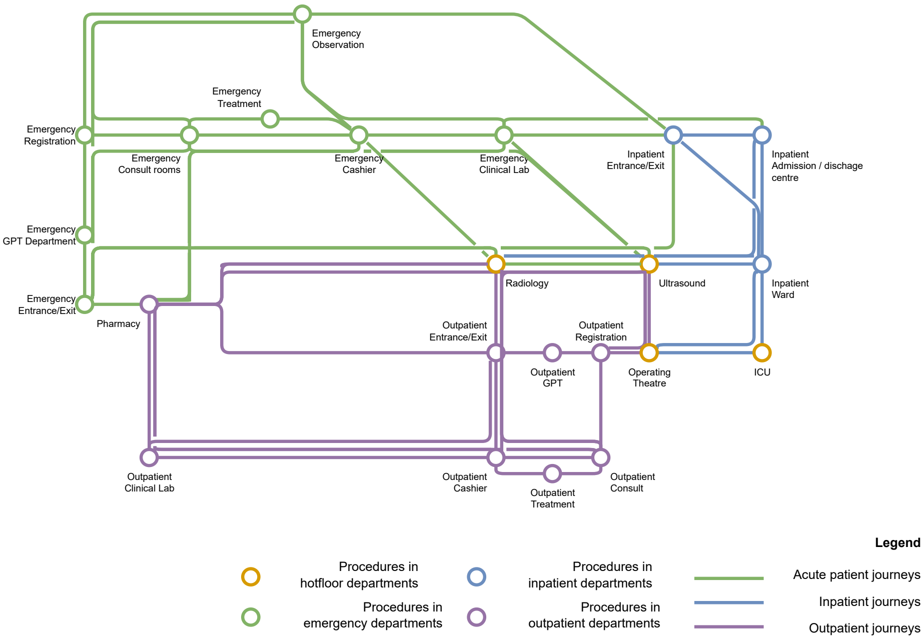


Figure 5. Metro network diagram of patients' paths in Panyu Central Hospital.

3. Research Methodology

This section is about what we need to have in an HCM and why. An HDSS is essential for designing a better hospital system, as it can quantitatively and systematically evaluate the design. A configuration model is a prerequisite for developing the HDSS because it contains spatial and non-spatial information about the hospital for evaluation. This section first talks about the required information in an HCM by introducing the use cases of an HDSS. It then introduces essential data needed in hospital ARC models, which are simplified equivalents of HCM. Additionally, this section discusses the required four types of data in an HCM; each type of data is explained with a detailed example. Lastly, this section introduces the available data models from which the four types of data can be extracted. The available data and information models include two types, i.e., spatial-information-type models, such as Building Information Models (BIM), Industry Foundation Classes (IFC), and IndoorGML, and operational-information-type models in the form of Business Process Model and Notation (BPMN) data models.

3.1. Use Cases of HDSS

This section explains what we envisage to be the use cases of the HDSS and concludes the list of necessary information in an HCM based on the desired functionalities of HDSS. The kind of information we need is not readily available in any existing file/information model. We are proposing a software application for assessing, managing or optimizing hospital configurations; the first logical step in developing such software is to have clear ideas about information processing and operations within the system.

To demonstrate the functionality and utility of the proposed HDSS, several use cases are described as examples by answering the following questions: Who would be the user of the HDSS? What questions can the HDSS answer? And at what stage of a project can these questions be answered? The required information for each use case is also summarized. However, this list is not meant to be and cannot possibly be exhaustive.

- Use case 1: The architect can use the HDSS to semi-automatically create a hospital layout at the layout design stage of a new project or optimize the hospital layout of an existing project. For this use case, the operational information of patients' medical procedures in the hospital is needed for obtaining an Activity Relations Charts (ARC) model (for further explanations please see Section 3.2). Thus, the HCM should contain the operational information on patients' medical procedures in the hospital.
- Use case 2: The architect can use this HDSS to assess the safety of the hospital environment during the layout design stage. The environment's safety can be measured by the visibility and accessibility of spatial units within the hospital. As the visibility increases, the nurse can supervise bigger areas and hence the safety of the environment can be improved. A network graph consisting of nodes and edges is needed for this function, where each node represents a spatial unit of the hospital and each edge connecting two nodes represents the adjacent relationship between the two nodes. Hence, the HCM of the HDSS should contain the topological information of a network graph.
- Use case 3: The hospital director can use the HDSS to check if the hospital will be overcrowded during the layout design stage. For this use case, we need the topological information of the network graph. We also need to incorporate semantic information into the graph by assigning the area of each spatial unit to its corresponding node, so that the average people density in the room/spatial unit can be computed to indicate the crowdedness.
- Use case 4: The hospital director can use this system to check if the patient waiting time will be too long in a new hospital project during the layout design stage. In this use case, the graph is again needed. We also need to integrate semantic information into the graph by assigning the name of each spatial unit (e.g., 'central waiting' or 'registration', etc.) to its corresponding node. The patient's waiting time will be from the time the patient enters the waiting room till the time the patient enters the diagnosis room.
- Use case 5: The head nurse can use this system to check if patients' walking distances will be too long in a new hospital project during the layout design stage. For this use case, we need the operational information of patients' medical procedures for

obtaining the optimal patient paths with the shortest distance. We also need the topological information of the network graph of the hospital. Furthermore, it is necessary to incorporate semantic information into the graph by assigning the name of each spatial unit to its corresponding node. This will enable the identification of specific patient paths within the graph. Finally, it is essential to integrate geometric information into the graph by assigning 3D coordinates to each node. This will allow for the calculation of the distances along the patient's path.

- Use case 6: The architect can use this system to check how difficult it will be for first-time visitors to find their way in a hospital project during the layout design stage. For this function of measuring the difficulty in wayfinding, the extra walking distance will be the criterion of measurement. Hospital space is large and complicated; when first-time visitors enter the hospital to look for their destinations, they might get lost and go to several wrong places before arriving at their destinations. Hence, their actual travel journey will be different from the optimal journey (i.e., the shortest path); the difference between the shortest path's distance and the patient's actual travel journey's distance will indicate how difficult it is for patients to find their way. This use case requires the same information as the use case 5.
- Use case 7: The hospital director can use this system to develop a digital twin for simulating the operational management of the existing hospital during the operation and maintenance stage. A digital twin can help hospital directors assess the impact of changes on system performance and predict the result of specific medical procedures [30]. For this use case, the needed information is the topological information of the hospital graph and the operational information, such as the patient's journey.

In summary, the HCM of the HDSS requires four main types of information: operational information, such as patient paths in the hospital, topological information, such as the hospital's network graph, semantic information, such as each room's name and area, and geometric information, such as each room's location represented by a 3D coordinate. Each type of information is explained with an example in Table 1. Note that developing the calculation methods for the performance indicators mentioned in the use cases above, such as crowdingness, patient waiting time, patient walking distance, and difficulty in wayfinding, is beyond the scope of this research.

3.2. ARC Model

As mentioned in use case 1 of Section 3.1, Activity Relations Charts or ARC models [19] can be thought of as the simplified graph-theoretical equivalents (or excerpts) of the HCMs. These ARC models are large square matrices that denote complex directed graphs, which use numbers indicating the relative importance of links in terms of frequencies of travel/transport between spatial/operational units of a hospital. Thus, it is clear that these ARC models form the basis of configurational approaches to the design and optimization of hospital layouts in computational design [31]. However, there is little in the scientific or professional literature about how these ARC models can be made systematically. Here, we propose a conceptual process for building these ARC models in a participatory process in consultation with the directors and planners of hospitals. The idea is to construct an ARC model in multiple logical steps by collating or superimposing multiple operational "paths" consisting of nodes that represent spatial/operational stations or operational milestones and links that indicate the smallest operational/procedural actions and their temporal duration (these attributes may or may not be used later for building Discrete Event Simulation models [31,32]). Our proposed method is based on the idea of compiling a list of operational/spatial stations (rooms) and conducting a workshop/survey with the stakeholders to collect their proposed operational paths consisting of chains/sequences of these spatial/operational units. By putting together these paths, literally by adding the graph adjacency matrices representing these paths, we can then construct the main ARC model and its directed adjacency matrix in one go. If desired, this graph can then be row-normalized to extract the relative importance of the links between 0 and 1 [33].

Figures 1–5 together provide an illustrative example of the process for building the ARC models. Specifically, Figures 1–3 are the proposed operational paths in the form of BPMN models. A BPMN model is the industry standard that uses flow charts for modeling and illustrating processes in complex systems [34]. These BPMN models (or flow charts) show all the space-related procedures included in the patient journeys in outpatient, inpatient, and emergency departments of a real-world hospital. The hospital selected for this research is Panyu Central Hospital, located in Guangzhou, China. It is selected due to its available operational data of patient journeys as well as its representative layout complexities [29]. The Panyu Central Hospital has three main departments, namely, the outpatient department, inpatient department, and emergency department. Figures 1–3 illustrate the typical patient journeys in the outpatient, inpatient, and emergency departments of Panyu Central Hospital in the form of a flow chart. Based on the flow charts, an Activity Relations Chart can be formed as illustrated in Figure 4, where the rows and columns are labeled by space-related procedures in the flow charts, and entries indicate the relationships between any two pairs of the procedures. The relations range from 0 to 1. If there is no connection between two procedures, the relation is 0. If a connection between two procedures exists, the relation is larger than 0 and is calculated according to the frequency of transitions between procedures. The higher the number is, the more adjacent the two procedures need to be to each other.

It can be observed that the ARC model itself is a weighted graph, where each cell in the first row/column is a node, and the entries of the ARC model are the edges associated, with weights ranging from 0 to 1. This graph can be represented in a more visual way, namely, a metro network diagram, as illustrated in Figure 5, where each procedure is represented by a circle (node) and the connections between different procedures are represented by lines (edges). Pairs of procedures with stronger connections (i.e., higher numbers in the ARC model) are put close to each other in this diagram. By indicating and visualizing which procedures need to be adjacent to each other, these ARC models and the metro network diagram can aid in use case 1 of the HDSS, where hospital layouts need to be designed or optimized.

3.3. Hospital Configuration Model

According to the HDSS's use cases and functionalities introduced in Section 3.1, we can conclude what data are needed in a Hospital Configuration Model. As illustrated in Figure 6, a Hospital Configuration Model contains spatial and non-spatial information. The spatial information can be further divided into two types, namely, topological information and geometric information. The non-spatial information can also be further divided into semantic information and operational information.

A detailed technical exposition of the HCM's components is summarized as follows:

- **Geometric Information**

  The geometric data in the HCM represent the physical shapes of the hospital, encompassing the boundaries and 3D spaces of rooms and corridors. These are defined using mathematical constructs, such as:

  - **Vertices and Edges**: Each room is represented as a polygon defined by a set of vertices (3D coordinates) and edges connecting these vertices. The polygon data are extracted from BIM/IFC models using tools like Revit and Dynamo. For an example of the room polygon data, please see geometric information in Table 1.

  - **Mesh Representation**: The 3D space of the room is represented by the mesh geometry, and the mesh representation algorithm is developed using the COMPAS library in Python [35].

- **Topological Information**
  Topological information encodes the spatial relationships between different functional units of the hospital, represented as a network graph. The graph consists of:

  - **Nodes**: Each spatial unit (room or corridor) is a node.
  - **Edges**: An edge between two nodes signifies the adjacency relationship.
  - **Attributes**: Each node can carry attributes such as the room name or room capacity.

  The nodes and edges are extracted from BIM/IFC models using Rhino and Grasshopper, and the network graph is built in Python using the NetworkX library [36]. For a simplified example of a hospital graph, please see the topological information in Table 1.

- **Semantic Information**
  Semantic information provides meaning to the spatial units by linking them to their functional roles. Examples include:

  - **Room Names**: Identifying units such as operating rooms, waiting areas, and diagnostic labs.
  - **Organizational Hierarchy**: Associating rooms with departments to enable functional grouping.

  The algorithm for extracting semantic information from the BIM/IFC model is implemented in Python, and the extracted data are stored in the form of a Python dictionary. For an example of extracted semantic information, please see Semantic Information in Table 1.

- **Operational Information**
  Operational information captures patient journeys within the hospital. A patient journey is a detailed sequence of rooms visited during a medical procedure, e.g., see Operational Information in Table 1.
  Business Process Model and Notation (BPMN) diagrams are used to standardize and visualize the patient journey. The patient journey data are represented as a Python list, with each element of the Python list being a room the patient needs to visit during the patient journey.

Section 4 explains how we extract the four types of information from available sources, such as BIM/IFC models and BPMN models, to form the HCM. In the HCM, all types of information are connected logically; the dashed lines in Figure 6 show the relationships between different classes, e.g., the relationship named ‘Person uses rooms’ indicates specific rooms that a person uses. Ideally, we will achieve consistency among the spatial information, semantic information, and operational information, to make the operational management in a hospital straightforward. The following subsections introduce the available data models (i.e., BIM/IFC models, IndoorGML models, and BPMN models) from which the four types of data can be extracted.

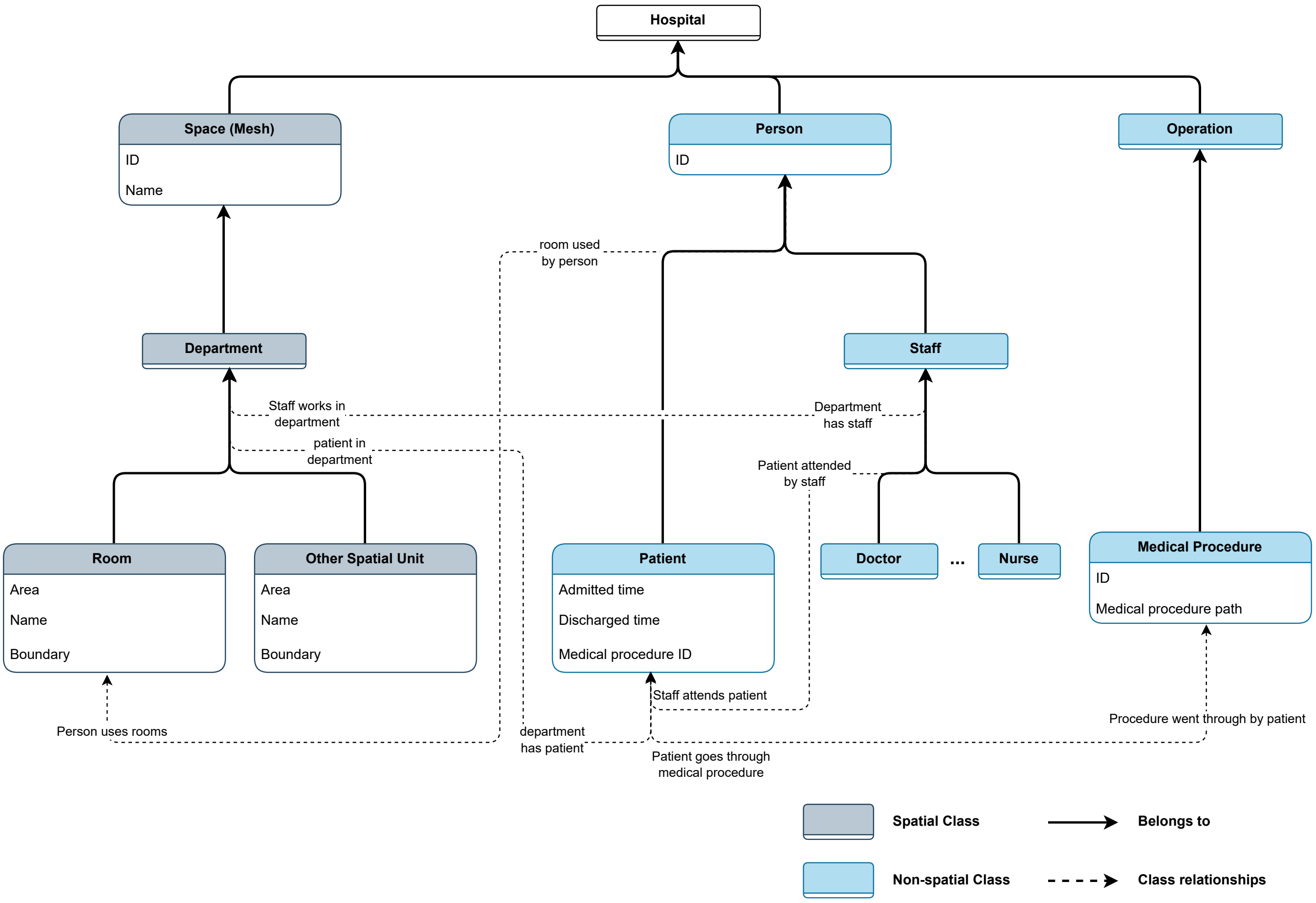


Figure 6. A UML diagram illustrating data included in a Hospital Configuration Model.

3.4. Hospital BIM and IFC Models

Building Information Modeling (BIM) consists of designing and using a digital 3D model of buildings to support architectural planning, design, construction, operation, maintenance, refurbishing and demolition [37]. BIM is widely used in the Architecture Engineering and Construction (AEC) domain for aiding the design and construction stages of an architectural project [37].

One typical challenge facing BIM is that different BIM software have different file formats that do not always support one another, which causes interoperability problems when exchanging files [37]. To solve this problem, the buildingSMART consortium has invented the industry foundation classes (IFC) as a common and open format for exchanging BIM models [37].

Due to the wide application of BIM/IFC in the AEC domain, most digital 3D models of hospitals are BIM/IFC models. Hence, we will use hospital BIM/IFC files as source files for extracting geometric, topological, and semantic information for the HCM.

3.5. IndoorGML

BIM/IFC models are very complex and contain much information that is irrelevant to geospatial applications and research [6]. They integrate geometric, spatial, structural, and material information across different stages of a building's life cycle. The information we need for the use cases and functionalities of the HDSS is only a small part of the entire information in the BIM/IFC model; the rest of the information is irrelevant to our research purpose. Hence, for simplicity and convenience reasons, we only need to extract relevant information from BIM/IFC models and abandon the rest.

We can extract IndoorGML files from BIM/IFC models. IndoorGML is an Open Geospatial Consortium (OGC) standard used for the description of 3D indoor spaces and facilitating applications such as indoor navigation [6]. IndoorGML files provide geometric, topological, and semantic information about indoor spaces, which suits the aim of our research [38].

IndoorGML models have two main parts: one is the *Primal Space Features* and the other is the *Node-Relation Graph* [38]. The *Primal Space Features* divide the indoor space of a building into cells; cells are representations of rooms and corridors. The *Node-Relation Graph* describes the relations among these cells (i.e., whether two cells are adjacent). The *Primal Space Features* are further divided into the *Cell Space* and *Cell Space Boundary*, where the *Cell Space* is the smallest spatial unit of a building, such as a room, corridor and staircase, etc., and the *Cell Space Boundary* is the door in a building. The *Node-Relation Graph* is also divided into two modules, i.e., *nodes* and *edges*. A *node* is the dual of the corresponding *Cell Space* (room) or *Cell Space Boundary* (door), and an edge connects two nodes if the two corresponding *Cell Spaces* (or *Cell Space Boundaries*) are adjacent. Figure 7 gives an illustration of the four modules of the IndoorGML, and Figure 8 is an example of the IndoorGML model extracted from the open-source hospital IFC model used in this research [39], where the red *Node-Relation Graph* is embedded in the transparent *Primal Space Features*.

Although IndoorGMLs seem suitable for our research, they present several challenges and drawbacks that are not conducive to this research as follows:

- There is a lack of available IndoorGML files in the industry because according to the literature study conducted in Section 1.2, there are no appropriate tools for generating correct IndooGML files. Furthermore, IndoorGMLs are encoded in XML (eXtensible Markup Language) format [40], which is complex, highly hierarchical, cumbersome to manage, and unpopular for web applications [37].
- While IndoorGML is designed to support applications in indoor navigation and facility management, effective execution of these tasks typically requires integration with additional data, such as operational information and enriched semantic information. However, IndoorGML files currently face the challenge of lacking this critical supplementary information.

Hence, IndoorGML itself cannot meet our research requirements. To deal with the above challenges, we first developed our tool for semi-automatically generating correct IndoorGML files from BIM/IFC models. The generated IndoorGML files are further parsed into JSON (JavaScript Object Notation) format [41], which is a more popular format due to its readability and edit-ability. Our tool can also integrate operational and semantic information into IndoorGML's spatial information to make the later simulation modeling feasible [42].

3.6. Hospital BPMN Models

Since hospital BIM/IFC and IndoorGML models do not contain operational information, such as medical processes or patient journeys, we need to collect this information from other sources. In this research, we extract the operational information for the HCM from business process model notation (BPMN) models. BPMN is an industrial standard for modeling business processes; it uses flow charts to visualize the steps included in the business process [34]. It is a common approach to model hospitals' medical processes [34,43]. Hospitals as complex systems have very complicated processes involving different users and multiple steps taking place in different places. Hence, BPMN is a perfect method to model these processes. We will use BPMN models as our source files for extracting operational information into the HCM.

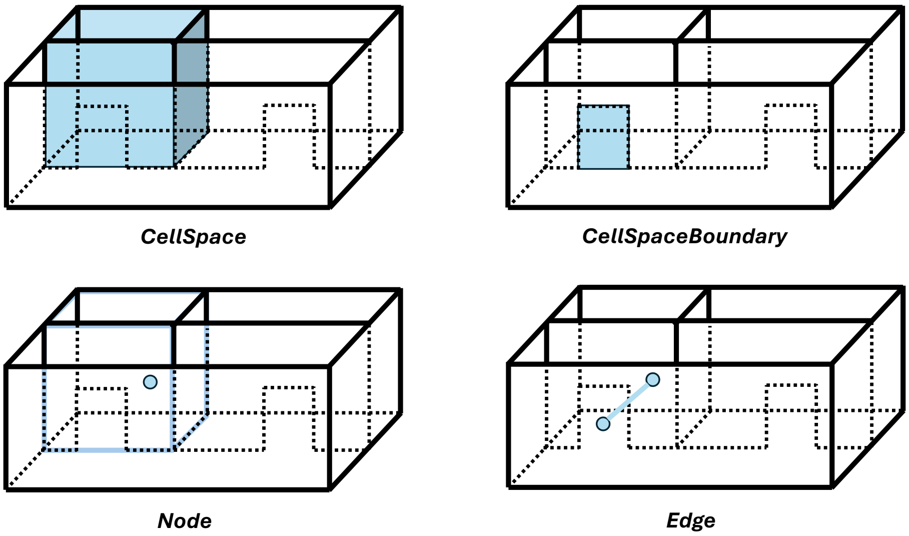


Figure 7. An illustration of IndoorGML's data structure.

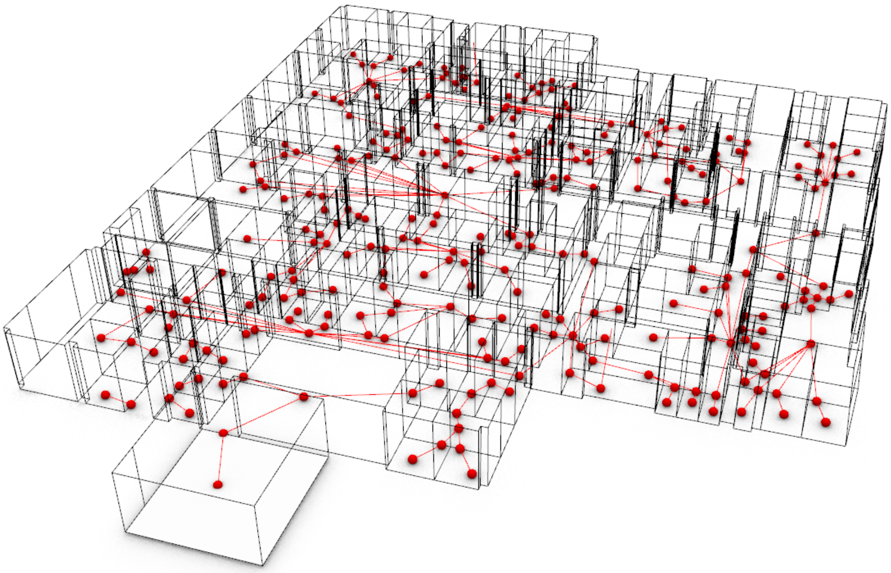


Figure 8. An example of a hospital IndoorGML model, image source: [42].

In this research, we selected representative hospital operational information about patient journeys from Peng's study [29] and manually modeled this information into BPMN models. The automatic generation of BPMN models is beyond the scope of this paper. However, we propose that an expert (such as an industrial engineer or someone familiar with operations research) should systematically extract such information from textual and visual documents concerning the operational management and service design of a hospital to construct multiple BPMN models to describe the main procedural workflows in a hospital. Figures 1–3 show BPMN models that we built for modeling the medical processes in outpatient, inpatient, and emergency departments in a real-world hospital. We used these three BPMN models as our source files for extracting the operational information into the HCM. Section 4 explains how we extract operational information from these BPMN models.

4. Research Results

In this section, we discuss how we extract the required information of our HCM from what is typically available as data and information models.

We introduce the main results of this research, which are our methodologies of extracting data from BIM/IFC models to build the HCM.

From Hospital IFC Model to HCM

This research developed software to automatically generate a configuration model from an IFC model for the evaluation of the hospital's functionality and efficiency. The generation workflow includes two main steps. The first step is converting the hospital IFC model to the IndoorGML model. The second step is to build the HCM from the IndoorGML file. Figure 9 depicts the workflow of converting the Hospital IFC Model to the HCM.

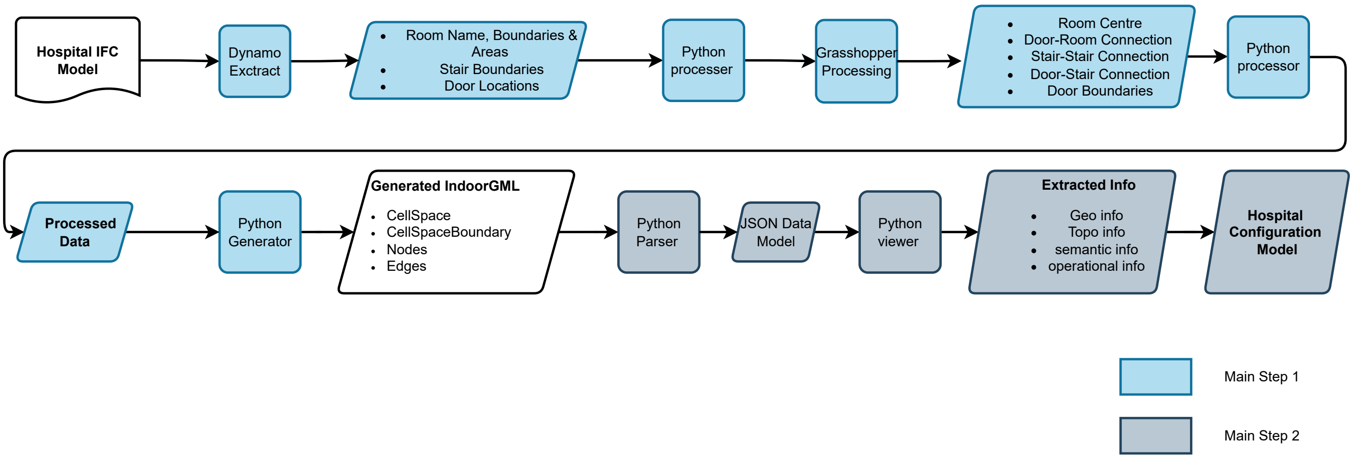


Figure 9. The workflow from Hospital IFC Model to Hospital Configuration Model, source: [42].

Our software for generating IndoorGML is inspired by Tong and Zheng's work but is more generalized and works for more complex buildings with rooms of irregular shapes. As illustrated in Figure 9, the workflow of generating IndoorGML files starts with extracting relevant information from the IFC model using Autodesk's Revit and Dynamo. Three groups of data (i.e., room names, boundaries and areas, stair boundaries, and door locations) are taken out of the IFC model for building IndoorGML's geometric and semantic information. The extracted data can be further used to generate IndoorGML's topological information (i.e., Node-Relation Graph), but because Dynamo and Grasshopper use different data structures to store data, the data from Dynamo needs to be first processed in Python so it is readable by Grasshopper. Once the data are imported into Grasshopper, the scripts in Grasshopper read the data and generate a Node-Relation Graph of the IndoorGML model. The data exported from Grasshopper again need to be processed in Python so that they can be read by other Python libraries to write an IndoorGML file. Since the IndoorGML file has an XML-based exchanged format [44], we use etree [45], an XML library for Python, to write the IndoorGML file. The processed geometric, topological, and semantic information are read by the Python generator and turned into XML-formatted output (i.e., IndoorGML). For more implementation details about the software, such as the scripts in Dynamo, Grasshopper, and Python, please refer to the study by Jia et al. [42].

The development of the HCM follows the design pattern of model-view-controller (MVC). MVC is the most common design pattern for developing software or user interfaces [46]. This pattern divides the program logic into three separate yet interconnected parts, i.e., data model, view, and controller [47]. Figure 10 presents the interrelationships among the three parts of the MVC. The **data model** component of the MVC is a data structure, which contains all the raw data of the project [48]. In the case of the HCM, the data model is derived from the IndoorGML file and carries all the geometric, topological, semantic and operational information of a hospital system. The **view** component of MVC presents the model's information to the users [48]. It contains functions to access the data model and organize the data more logically so that humans can easily read them. We have developed view functions of our HCM according to the use cases introduced in Section 3.1 for users to easily query relevant information. The **controller** component of MVC serves as an intermediary between the model and the view. It listens to the event triggered by the view and makes a response to the model (e.g., adding or changing information to the data model) [48]. In the following text, we demonstrate how we obtain the data model of the HCM, develop view codes to extract information from the data model and present them in a human-readable manner, and develop controller codes to add or edit contents in the data model.

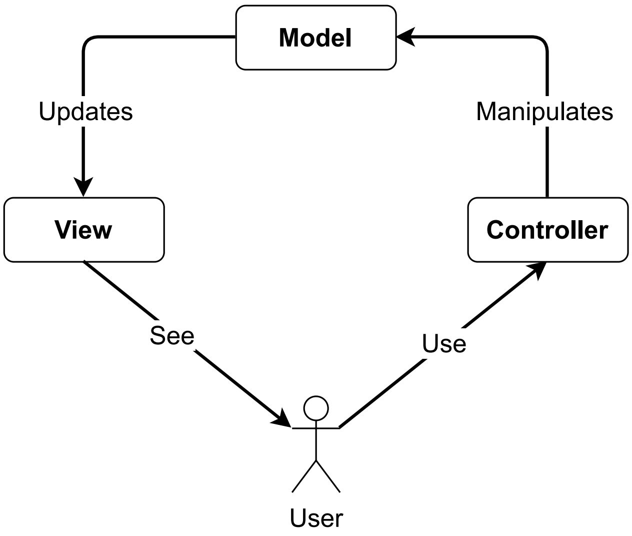


Figure 10. The basic architecture of MVC, image source [49].

- **Obtaining the Model part of the HCM**
  Figure 9 shows that the development of the data model for the HCM includes parsing the IndoorGML file into a JavaScript Object Notation (JSON) file [41]. The parser for IndoorGML used in this study is developed by Ledoux [50]. With this parser, all the information in the IndoorGML (e.g., cell spaces, cell space boundaries, nodes and edges) is parsed into a JSON file as the data model.
- **Developing the View part of the HCM**
  The information presented by the view codes should be a mathematical construct about sets and relations. These relations are graphs, so most of our view functions should output sets and graphs or hyper-graphs (a mesh is a hyper-graph and the edge in a face is considered to be a hyper-edge). According to the use cases demonstrated in Section 3.1, we developed view codes to extract the following information: a mesh (hyper-graph) describing the geometrical information (i.e., the *Primal Space Features*) in the IndoorGML model, a graph showing the topological information (i.e., the *Node-Relation Graph*) in the IndoorGML model, a set of room names and room areas showing the semantic information of the hospital in the IndoorGML model, another set of hospital departments and all the rooms within their respective departments, which shows the semantic information of the hospital's organizational structure, and lastly, a set of lists demonstrating the operational information of patient journeys in the hospital.
  The mesh output is obtained using the COMPAS library in Python. COMPAS is an open-source framework designed for computational research in the fields of architecture, engineering, digital fabrication and construction [35]. Users can use the

view code to generate and visualize the mesh geometry to obtain a view of what the IndoorGML model of the building looks like. The graph output is obtained by developing Python scripts using the NetworkX library, a Python package made for creating, manipulating, and analysing complex networks [36]. Users can use the view code to obtain the network graph which will be the base for running the simulation modeling. The simulation modeling is one of the core functionalities of HDSS, as mentioned in use cases 4, 5, and 6. Figure 8 is a visualization of the mesh output and graph output generated from the open-source hospital IFC model [39], where the red graph is embedded in the transparent mesh. The set output of room names and room areas is a Python dictionary. A Python dictionary is a data structure in Python that stores data in key-value pairs (e.g., {key: value}) [51]. In the Python dictionary of room names and areas, the room name is the key and the room area is the value. Users can use the view code to obtain all the room areas, which can aid in addressing use case 3 of assessing crowdingness in hospitals. Specifically, in the later simulation modeling step, once the room area and the number of people in the room are known, it is straightforward to assess the room's crowdingness by calculating the peoples' density in the room. The set output of departments and rooms is also a Python dictionary. In this dictionary, each department is represented as a key, and its associated rooms are grouped as the corresponding values.

In Appendix A, Figure A1 shows the Python code to implement this function of organizing all rooms in the hospital into their respective departments and Table A1 shows the resulting Python dictionary of hospital departments and rooms. The extracted operational information related to patient journeys in the hospital is in the form of Python lists. For extracting this information, we first turned the BPMN models (Figures 1–3) into multiple lists (e.g., see Input data list in Table 2), where each element in the list is a space-related procedure in the BPMN (i.e., rounded-corner rectangle in the flow Figures 1–3), and the entire list is a complete medical procedure in the BPMN flow chart (i.e., the procedure starts with patient entering the hospital and ends with patient leaving hospital). Subsequently, we developed view codes to identify the corresponding room names of list elements based on the extracted semantic information of hospital departments and rooms (Table A1). For example, the corresponding room name for the element 'registration' in the semantic information dictionary is 'RECEPTION1B13'. The view codes find the corresponding room names for each element in the list and put all corresponding room names into a new list (e.g., see Output data list in Table 2). The new lists contain extracted operational data on patient journeys, which can serve as input for HDSS simulation modeling, e.g., these data enable determining the shortest path for patients/agents in DES or ABM simulations. Table 2 provides an example of the input data list generated from the BPMN model and an example of the output data list of operational information generated from the input data list. It should be noted that one element in the input data list might have multiple corresponding room names in the output data list. This is because in a hospital, there can be multiple rooms for the same function. For example, the element 'diagnosis' in the input list has twelve corresponding room names ('INTERACTIONSTATION1D07', 'INTERACTIONSTATION1D08', etc.) in the output data list because, in the selected hospital BIM model, there are twelve rooms all serving the same function of diagnosis. Hence, the patient can have twelve options when choosing the diagnosis room and there will be twelve different potential paths for the patient to complete the same patient journey. For more implementation details of the view codes' Python scripts, please refer to [42] or the repository (https://github.com/ZhuoranJia/IFC2BCM, accessed on 25 November 2024).

Table 2. Source data and output data of HCM’s operational information, source: [42].

Input Data List	Output Data List
origianl_medical_path_1 = [‘registration’, ‘triage’, ‘waiting’, ‘diagnosis’, ‘medicine’]	medical_path_1 = [‘RECEPTION1B13’, ‘WTSandMEAS.ROOM1D15’, ‘WTSandMEAS.ROOM1D30’, ‘WAITING/ACTIVITYAREA1DC1’, ‘INTERACTIONSTATION1D11’, ‘INTERACTIONSTATION1D07’, ‘INTERACTIONSTATION1D32’, ‘INTERACTIONSTATION1D02’, ‘INTERACTIONSTATION1D13’, ‘INTERACTIONSTATION1D36’, ‘INTERACTIONSTATION1D10’, ‘INTERACTIONSTATION1D08’, ‘INTERACTIONSTATION1D09’, ‘INTERACTIONSTATION1D28’, ‘INTERACTIONSTATION1D34’, ‘INTERACTIONSTATION1D35’, ‘PHARM.DISP.1A16’]

- **The Controller part of the HCM**
  The controller codes we envisage are for updating the data model; in other words, adding/changing information to the data model. Once the simulation is complete, we need to update the data model by adding the dis-aggregated simulation results and aggregated evaluation results to the data model, so that users can easily view them. For example, once the simulation is finished and we know each room’s people density, we need to add this attribute to the dictionary that describes the room’s information. Table 3 provides an example for illustrating how one part of the data model has been changed before and after the controller code adds information to it. In addition to adding the dis-aggregated information, we also propose controller codes for adding aggregated information, such as the average people density, average patient walking distance, average patient waiting time, and average patient’s extra walking distance. Another example of the use of controller code is for updating the data model’s network graph. The original network graph only contains topological information; the controller codes can integrate semantic and geometric information into the graph, for example, assigning each node its corresponding room name, area, and 3D coordinate. By adding such information to the graph, the graph can aid in the simulation modeling, such as finding the shortest path in the graph according to the patient journey data list (output data list in Table 2), calculating the distance along the shortest path, and calculating the people density in a room.

Table 3. An example showing how controller codes add information to the data model (please note the data for people density in this example are hypothetical).

	Room Attributes
Before	{‘CENTRALWAITING’: {‘area’: ‘127’}, ‘WAITING/ACTIVITYARE’: {‘area’: ‘178’}, . . . }
After	{‘CENTRALWAITING’: {‘area’: ‘127’, ‘people density’: ‘0.9’}, ‘WAITING/ACTIVITYARE’: {‘area’: ‘178’, ‘people density’: ‘1.0’}, . . . }

5. Discussion

This study presents a novel Hospital Configuration Model (HCM) as the foundational component for a Hospital Design Support System (HDSS). By integrating geometric, topological, semantic, and operational data, the HCM enables the HDSS to utilize simulation modeling for assessing hospital layout efficiency and functionality. In this section, we summarize the key findings, the comparison with the academic literature, and the research limitations.

This study presents several key findings that contribute to advancing the field of hospital layout design and simulation modeling. First, a robust systematic methodology was developed for the semi-automatic generation of IndoorGML files from existing Building Information Models (BIM) and Industry Foundation Classes (IFC) data. This approach helps enhance the accessibility and availability of IndoorGML data for applications such as indoor navigation and indoor location-based services. The methodology further facilitates the automatic transformation of IndoorGML files into Hospital Configuration Models (HCMs). These HCMs, integrating geometric, topological, semantic, and operational data, enable spatial analysis and simulation modeling. To validate the methodology, it was successfully applied to a real-world hospital BIM model, resulting in an HCM which allows for a quantitative evaluation of hospital layout designs, focusing on key performance indicators, such as crowdingness, patient walking distance, patient waiting time, and difficulty in wayfinding.

This research bridges the gaps of other related studies. As discussed in Section 1.2, several gaps and limitations are evident in the existing body of related studies. Firstly, certain tools, such as those developed by Diakite et al. [6] and Intratech [7], generate IndoorGMLs that lack semantic information. Our methodology can generate IndoorGMLs integrated with semantic data. Secondly, Tong and Zheng's software [8] is incapable of dealing with complex building models with irregular shapes—our methodology overcomes this limitation. Thirdly, many existing tools do not support the generation of IndoorGML from other sources. These tools include the ones developed by Jeong et al. [9], Brincovean and Butean [10], Taehoon [11], and Claridades et al. [12]. Our methodology can generate IndoorGMLs from BIM/IFC files. Fourthly, several tools, such as those developed by Yuan and Schneider [13] and Teo and Cho [14], generate layout representation models in other formats, instead of the industry-standard IndoorGML format, which reduces the data interoperability. Our methodology can generate correct IndoorGML files, enhancing data interoperability. Additionally, the approach developed by Khan et al. [15] includes the extra step of converting IFC files into CityGMLs, and then extracting IndoorGMLs from CityGMLs, which is cumbersome. Our methodology can directly extract IndoorGMLs from IFC models. Lastly, Srivastava et al. [17]'s methodology and Hashim et al. [18]'s workflow can only generate IndoorGML files from CAD and Sketchup models instead of BIM/IFC models. The lack of 3D geometric information in CAD models and semantic information in SketchUp models makes them less suitable as source models for this research compared to BIM/IFC models. Our methodology is designed to be able to work with BIM/IFC inputs. Furthermore, none of the studies developed further steps for editing the IndoorGML files. Our methodology can subsequently convert the IndoorGML files into HCMs. The advantages of HSMs over IndoorGMLs can be found in Section 3.5.

While this research provides valuable insights, it is important to note that this research has several limitations. Firstly, the operational information discussed in this research focuses solely on the patient aspect. Specifically, when extracting operational data into the HCM, we concentrated only on the patient journey. However, operational information encompasses additional aspects, such as staff workflows, which include staff movement patterns and their interactions with both other staff and patients. This aspect is also integral to HCM's operational information and could provide further insights if considered. Secondly, the operational data integrated into the HCM were derived from pre-existing sources and may not comprehensively capture the dynamic variability of hospital workflows. The static nature of some operational inputs might limit the model's ability to simulate highly dynamic scenarios, such as those involving emergencies or sudden changes in patient flows. Incorporating real-time or stochastic operational data into the model could enhance its predictive capabilities and applicability in more complex scenarios. The third limitation of the developed methodology lies in its approach to representing spatial connections in the hospital layout graph. When generating the graph for the hospital layout, the methodology creates edges by connecting a room's node to the corresponding node of its door. Consequently, the methodology only establishes connections between two spatial units if they are linked by a door. For instance, if a room and a corridor share the same door, they will be connected through that door. However, if two corridors are directly connected without an intervening door, the methodology does not create an edge between them, leaving them unlinked. This approach introduces inaccuracies, as spatial units that are directly connected without doors should still be represented as connected by an edge in the graph. Additionally, the methodology is developed using a combination of Python scripts, Grasshopper scripts, and Dynamo scripts. While this multi-tool approach leverages the unique strengths of each platform, it also imposes significant challenges on users. Specifically, users must switch between these three tools to execute the software's functionality, resulting in a fragmented and cumbersome workflow. This lack of integration not only complicates the user experience but also introduces potential inefficiencies, such as increased learning curves, higher risks of user error, and reduced operational consistency. Lastly, the application and validation of the methodology were conducted on a single real-world hospital BIM model. While this case study demonstrates the feasibility and effectiveness of the proposed approach, the generalizability of the findings to other hospital layouts with varying levels of complexity and functional requirements remains to be further investigated.

6. Conclusions

The aim of this study was to develop a methodology for building a Hospital Configuration Model as the foundational component of a Hospital Design Support that applies simulation modeling for providing evaluation mechanisms on hospital layout performances in terms of operational efficiencies and functionalities. In practical application, the methodology can be used at the layout design stage of a new building project; it can also be used at the operation and maintenance stage of an existing hospital. The research results include the methodology of semi-automatically generating configuration models for hospitals from BIM/IFC models. The methodology has two parts: the first part includes the conversion of hospital BIM/IFC models to IndoorGML models, and the second part pertains to the automatic generation of HCM from the IndoorGML model. The following sub-sections discuss the future research directions and implications of this study.

6.1. Future Research

Our future research and development should, in particular, focus on the development of HDSS prototypes, which use an HCM as input and implement operational simulation models for assessing the accessibility of services and the efficiency of mobility. The HDSS development process further involves establishing methodologies for calculating key performance indicators, including average people density, average patient waiting time, average patient walking distance, and average difficulty in wayfinding. Another potential direction for future research is to establish a standard way of extracting hospital operational information (e.g., medical procedures) into a data model using the methods of BPMN.

6.2. Implications

The HCM's implications for policy, industry, and economy are summarized as follows:

- The HCM can help policymakers in establishing guidelines that ensure new hospital layout designs prioritize patient outcomes and operational efficiency. By mandating early-stage evaluations of layout designs against operational requirements, regulatory bodies can minimize hospital inefficiencies and operational expenditures.
- An HCM can aid in the application of **space optimization** by providing the basis to study relationships and flows between different spatial units.
- Together with the operational information, an HCM can be used as a digital twin for simulating and monitoring the daily operations of a hospital, e.g., in **operational management** and in **facility management**.
- An HCM can help improve the safety of a building by optimizing the placement of guards or cameras to ensure maximum coverage while keeping the lowest number of guards/cameras within the building.
- An HCM can be augmented with 3D information (after the hospital is realized) to help build a model for indoor navigation and way-finding.
- By optimizing hospital layouts and operational flows, the HCM can help reduce costs related to inefficiencies, such as prolonged patient waiting times and excessive patient walking distances. For hospitals, these improvements can translate into lower operational costs, and enhanced capacity to serve more patients without increasing physical space or workforce.
- For the construction and architecture sectors, the integration of HCM into hospital design processes promotes cost-effective planning, reducing redesign expenses and construction overruns.

We can conclude that a specific type of information model, dubbed a Hospital Configuration Model (HCM) is needed to collate spatial and operational information concerning the planned procedures in a hospital as a core of a class of information systems called Hospital Design Support Systems (HDSSs). In this paper, we introduced and proposed some constructs that need to be embodied into an HCM with an outlook towards the envisaged use cases for an HDSS. It is only natural that when prototyping an HDSS, we might realize that we need to revise our HCM, but such revisions are quite common and necessary in design science research [52,53]. For now, given the outlook of usage of the HDSS, the most essential types of information to be squeezed into an HCM are the spatial, configuration, semantic, and operational pieces of information as introduced in the paper.

Author Contributions: Conceptualization, Z.J. and P.N.; methodology, Z.J. and P.N.; software, Z.J. and P.N.; validation, Z.J.; formal analysis, Z.J. and P.N.; investigation, Z.J.; resources, Z.J. and P.N.; data curation, Z.J.; writing—original draft preparation, Z.J.; writing—review and editing, P.N.; visualization, Z.J.; supervision, P.N., P.L. and C.W.; project administration, P.L. and C.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original data presented in the study are openly available in IFC2BCM (https://github.com/ZhuoranJia/IFC2BCM, accessed on 25 November 2024).

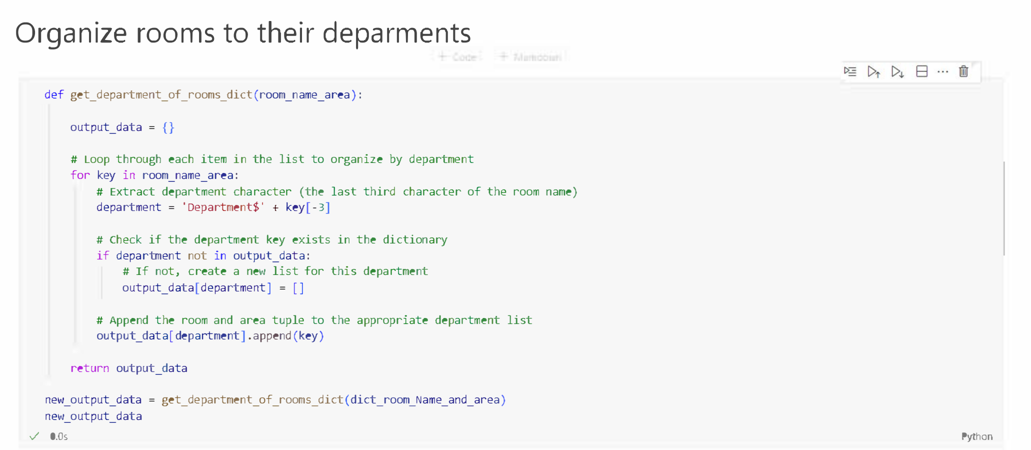
Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

HDSS	Hospital Design Support System
HCM	Hospital Configuration Model
DES	Discrete-Event Simulation
ABM	Agent-Based Modeling
DAG	Directed Acyclic Graph
ARC	Activity Relations Charts
BPMN	Business Process Model Notation
MVC	Model-View-Controller
JSON	JavaScript Object Notation

Appendix A



Organize rooms to their deparments

```python
def get_department_of_rooms_dict(room_name_area):

    output_data = {}

    # Loop through each item in the list to organize by department
    for key in room_name_area:
        # Extract department character (the last third character of the room name)
        department = 'Department$' + key[-3]

        # Check if the department key exists in the dictionary
        if department not in output_data:
            # If not, create a new list for this department
            output_data[department] = []

        # Append the room and area tuple to the appropriate department list
        output_data[department].append(key)

    return output_data

new_output_data = get_department_of_rooms_dict(dict_room_Name_and_area)
new_output_data
```

Figure A1. View function for extracting semantic information, source: [42].

Table A1. Extracted semantic information of the HCM. Note: for simplicity reasons, the information shown here is incomplete; for a complete version of the information please see this repository (https://github.com/ZhuoranJia/IFC2BCM, accessed on 25 November 2024).

Departments & Rooms
{‘Department$A’: [‘CENTRALWAITING1AC1’, ‘CORRIDOR2AC3’, ‘PHARM.DISP.1A16’, ‘CORRIDOR2AC1’, ‘DENTALWAITING2A11’, . . . ‘X-RAYALCOVE2A12-A’]}
{‘Department$B’: [‘CORRIDOR1BC2’, ‘LAB1B04’, ‘CORRIDOR1BC4’, . . . ‘RECEPTION1B01’, ‘RECEPTION1B13’, ‘TECHOFFICE2B9’]}
{‘Department$D’: [‘WAITING/ACTIVITYAREA1DC1’, ‘MAINMECHANICALROOM2D05’, . . . ‘INTERACTIONSTATION1D11’, ‘INTERACTIONSTATION1D07’, ‘INTERACTIONSTATION1D08’, ‘INTERACTIONSTATION1D09’, ‘INTERACTIONSTATION1D28’, ‘INTERACTIONSTATION1D34’, ‘INTERACTIONSTATION1D35’, . . . ‘COMPUTERROOM2D04A’]}
. . .

References

1. Jia, Z.; Nourian, P.; Luscuere, P.; Wagenaar, C. Spatial decision support systems for hospital layout design: A review. *J. Build. Eng.* **2023**, *67*, 106042 . [CrossRef]
2. Chraibi, A.; Kharraja, S.; Osman, I.; El Beqqali, O. A Mixed Integer Programming formulation for solving Operating Theatre Layout Problem: A Multi-goal Approach. In Proceedings of the 2013 International Conference on Industrial Engineering and Systems Management (IESM), Rabat, Morocco, 28–30 October 2013.
3. Ulrich, R.; Quan, X.; Joseph, A.; Choudhary, R.; Zimring, C. *The Role of the Physical Environment in the Hospital of the 21 st Century: A Once-in-a-Lifetime Opportunity*; The Center for Health Design: Concord, CA, USA, 2004.
4. Burgio, L.D.; Engel, B.T.; Hawkins, A.; McCormick, K.; Scheve, A. A Descriptive Analysis of Nursing Staff Behaviors in a Teaching Nursing Home: Differences Among NAs, LPNs, and RNs. *Gerontologist* **1990**, *30*, 107–112. [CrossRef]
5. Peponis, J.; Zimring, C.; Choi, Y.K. Finding the Building in Wayfinding. *Environ. Behav.* **1990**, *22*, 555–590. [CrossRef]
6. Diakite, A. A.; Díaz-Vilariño, L.; Biljecki, F.; Isikdag, Ü.; Simmons, S.; Li, K.; Zlatanova, S. IFC2INDOORGML: An open-source tool for generating IndoorGML from IFC. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2022**, *43*, 295–301. [CrossRef]
7. Intratech. intratech/KICT-Autocad-RevitToIndoorGML. 2022. Available online: https://github.com/intratech/KICT-Autocad-RevitToIndoorGML (accessed on 22 November 2024).
8. Tong, Z.; Zheng, H. Generation Scheme of IndoorGML Model Based on Building Information Model. In *Proceedings of the Phygital Intelligence*; Yan, C., Chai, H., Sun, T., Yuan, P.F., Eds.; Springer Nature: Singapore, 2024; pp. 225–234. [CrossRef]
9. Jeong, H.; Ryoo, H.G. InFactory: A RESTful API server for easily creating IndoorGML. *ISPRS—Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2018**, *42*, 77–84. [CrossRef]
10. Brincovean, L.; Butean, A. I-Locate. A Comprehensive Solution for Indoor/Outdoor Localization. *Orașe Intel. Dezvoltare Reg.* **2017**, *1*, 61–70.
11. Taehoon, K. STEMLab/JIneditor. 2023. Available online: https://github.com/STEMLab/JIneditor (accessed on 21 December 2024).
12. Claridades, A.R.; Park, I.; Lee, J. Integrating IndoorGML and Indoor POI Data for Navigation Applications in Indoor Space. *J. Korean Soc. Surv. Geod. Photogramm. Cartogr.* **2019**, *2019*, 359–366. [CrossRef]
13. Yuan, W.; Schneider, M. iNav: An indoor navigation model supporting length-dependent optimal routing. In *Lecture Notes in Geoinformation and Cartography*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 299–313. [CrossRef]
14. Teo, T.A.; Cho, K.H. BIM-oriented indoor network model for indoor and outdoor combined route planning. *Adv. Eng. Inform.* **2016**, *30*, 268–282. [CrossRef]
15. Khan, A.; Donaubauer, A.; Kolbe, T. A multi-step transformation process for Automatically generating indoor routing graphs from semantic 3D building models. In Proceedings of the Journal Abbreviation: 9th 3DGeoInfo Conference 2014—Proceedings Publication Title: 9th 3DGeoInfo Conference 2014—Proceedings, Dubai, United Arab Emirates, 11–13 November 2014.
16. Open Geospatial Consortium . CityGML. Available online: https://www.ogc.org/publications/standard/citygml/ (accessed on 21 November 2024).
17. Srivastava, S.; Maheshwari, N.; Rajan, K. Towards generating semantically-rich indoorgml data from architectural plans. *ISPRS—Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2018**, *42*, 591–595. [CrossRef]
18. Hashim, M.N.; Hassan, M.I.; Abdul Rahman, A. Mobile indoor laser scanning For 3D strata registration purposes based on IndoorGML. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *42*, 241–245. [CrossRef]
19. Gozali, L.; Widodo, L.; Nasution, S.; Lim, N. Planning the New Factory Layout of PT Hartekprima Listrindo using Systematic Layout Planning (SLP) Method Planning the New Factory Layout of PT Hartekprima Listrindo using Systematic Layout Planning (SLP) Method. *Iop Conf. Ser. Mater. Sci. Eng.* **2020**, *847*, 012001. [CrossRef]
20. Foundation, C. Simulation Paradigms | CK-12 Foundation. 2014. Available online: https://www.ck12.org/book/ck-12-modeling-and-simulation-for-high-school-teachers%3a-principles-problems-and-lesson-plans/section/2.3/(accessed on 21 November 2024.
21. Hillier, F.S.; Lieberman, G.J. *Introduction to Operations Research*, 10th ed.; McGraw-Hill Education: New York, NY, USA, 2015; 1010p.
22. Leemis, L.M.; Park, S.K. *Discrete-Event Simulation: A First Course*; Pearson Prentice Hall: Saddle River, NJ, USA, 2006.
23. Elsenbroich, C.; Gilbert, N. (Eds.) Agent-Based Modelling. In *Modelling Norms*; Springer: Dordrecht, The Netherlands, 2014; pp. 65–84. [CrossRef]
24. Crooks, A.T.; Heppenstall, A.J. Introduction to Agent-Based Modelling. In *Agent-Based Models of Geographical Systems*; Heppenstall, A.J., Crooks, A.T., See, L.M., Batty, M., Eds.; Springer: Dordrecht, The Netherlands, 2012; pp. 85–105. [CrossRef]
25. Law, A.M. *Simulation Modeling and Analysis*; McGraw-Hill: New York, NY, USA, 2007.
26. Felix Brandl. Modelica vs. Simulink. 2023 . Available online: https://tlk-energy.de/blog-en/modelica-vs-simulink (accessed on 20 November 2024).

27. Documentation, S. Simulation and Model-Based Design. 2020. Available online: https://nl.mathworks.com/products/simulink.html (accessed on 20 November 2024).
28. Modelica Association. Modelica Language Specification. 2023. Available online: https://specification.modelica.org/maint/3.6/MLS.html (accessed on 20 November 2024).
29. Peng, Dejian. Improving the Performance of Hospitals: An Architectural Analysis of Patient Journeys in China. Ph.D. Thesis, Delft University of Technology, Delft, The Netherlands, 2022.
30. Semeraro, C.; Lezoche, M.; Panetto, H.; Dassisti, M. Digital twin paradigm: A systematic literature review. *Comput. Ind.* **2021**, *130*, 103469. [CrossRef]
31. Cubukcuoglu, C.; Nourian, P.; Tasgetiren, M.F.; Sariyildiz, I.S.; Azadi, S. Hospital layout design renovation as a Quadratic Assignment Problem with geodesic distances. *J. Build. Eng.* **2021**, *44*, 102952. [CrossRef]
32. çubukçuoğlu, C.; Nourian, P.; Sariyildiz, I.; Tasgetiren, M. Optimal Design of new Hospitals: A Computational Workflow for Stacking, Zoning, and Routing. *Autom. Constr.* **2022**, *134*, 104102. [CrossRef]
33. Nourian, P. Pirouz-Nourian/Spatial_Computing_Design_Studio18. 2022. Available online: https://github.com/Pirouz-Nourian/Spatial_Computing_Design_Studio18 (accessed on 3 November 2024).
34. Ruiz, F.; Garcia, F.; Calahorra, L.; Llorente, C.; Gonçalves, L.; Daniel, C.; Blobel, B. Business process modeling in healthcare. *Stud. Health Technol. Inform.* **2012**, *179*, 75–87.
35. Mele, T.V.; Casas, G.; Rust, R.; Li; beverlylytle; Kasirer, C.; Tetov, A.; Leung, V.; Rippmann; Lee, J.; et al. COMPAS: A Framework for Computational Research in Architecture and Structures. 2017. Available online: https://zenodo.org/records/14446252 (accessed on 20 November 2024).
36. Hagberg, A.; Swart, P.J.; Schult, D.A. Exploring network structure, dynamics, and function using NetworkX. In Proceedings of the 7th Python in Science Conference (SciPy2008), Pasadena, CA, USA, 19–24 August 2008; pp. 11–15.
37. Ohori, K.A.; Ledoux, H.; Peters, R. *3D Modelling of the Built Environment*; Delft University of Technology: Delft, The Netherlands, 2022.
38. OGC® IndoorGML 1.1. Available online: https://docs.ogc.org/is/19-011r4/19-011r4.html (accessed on 20 November 2024).
39. van Berlo, L . Sample-Test-Files/IFC 2x3/Medical-Dental Clinic at master · buildingSMART/Sample-Test-Files · GitHub. Available online: https://github.com/buildingSMART/Sample-Test-Files/tree/main/IFC%202x3/Medical-Dental%20Clinic (accessed on 12 November 2024).
40. W3C Recommendation. Extensible Markup Language (XML) 1.0 (Fifth Edition). 2008. Available online: https://www.w3.org/TR/xml/#sec-intro (accessed on 3 November 2024).
41. Python Software Foundation. json—JSON encoder and decoder. Available online: https://docs.python.org/3/library/json.html (accessed on 3 November 2024).
42. Jia, Z.; Nourian, P.; Luscuere, P.; Wagenaar, C. IFC2BCM: A Tool for Generating IndoorGML and Building Configuration Model from IFC. *SoftwareX* **2025**, *29*, 101975. [CrossRef]
43. Kassim, S.A.; Gartner, J.B.; Labbé, L.; Landa, P.; Paquet, C.; Bergeron, F.; Lemaire, C.; Côté, A. Benefits and limitations of business process model notation in modelling patient healthcare trajectory: A scoping review protocol. *BMJ Open* **2022**, *12*, e060357. [CrossRef]
44. Ledoux, H. Are your IndoorGML files valid? *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2020**, *6*, 109–118. [CrossRef]
45. Python Software Foundation. xml.etree.ElementTree — The ElementTree XML API. 2024. Available online: https://docs.python.org/3/library/xml.etree.elementtree.html (accessed on 5 November 2024).
46. Tutorials Point. Model View Controller Pattern. 2025. Available online: https://www.tutorialspoint.com/python_design_patterns/python_design_patterns_model_view_controller.htm (accessed on 5 November 2024).
47. Reenskaug, T.; O' Coplien, J. The DCI Architecture: A New Vision of Object-Oriented Programming. 2009. Available online: https://web.archive.org/web/20090323032904/https://www.artima.com/articles/dci_vision.html (accessed on 10 November 2024).
48. Geeks for Geeks. MVC Design Pattern, 2017. Section: Design Pattern. Available online: https://www.geeksforgeeks.org/mvc-design-pattern/ (accessed on 10 November 2024).
49. Mohd Nor, R.; Jalaldeen, M.; Razi, M.; Zakaria, A.; Safiuddin, A.; Fakhri, A.; Zulaiha, P.; Saat, A. Cloudemy: Step into the Cloud. *J. Telecommun. Electron. Comput. Eng.* **2018**, *9*, 3–5.
50. Ledoux, H. Indoorjson/Software/ig2ij at Master · Tudelft3d/Indoorjson · GitHub. 2020. Available online: https://github.com/tudelft3d/indoorjson/tree/master/software/ig2ij (accessed on 10 November 2024).
51. Python Software Foundation . Python Data Structures. Available online: https://docs.python.org/3/tutorial/datastructures.html (accessed on 10 November 2024).

52. March, S.T.; Smith, G.F. Design and natural science research on information technology. *Decis. Support Syst.* **1995**, *15*, 251–266. [CrossRef]
53. Peffers, K.; Tuunanen, T.; Rothenberger, M.A.; Chatterjee, S. A Design Science Research Methodology for Information Systems Research. *J. Manag. Inf. Syst.* **2007**, *24*, 45–77. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.