# An Ontology-Based Automatic Layout Design for Cabin Hospitals

**A thesis submitted for the degree of Doctor of Philosophy**

**By**

**Zhen DANG**

**Department of Mechanical, Aerospace, and Civil Engineering**

**College of Engineering, Design, and Physical Sciences**

**Brunel University London**

**May 2025**

# Abstract

The integration of Information and Communication Technology (ICT) has significantly transformed the construction industry, with Building Information Modeling (BIM) emerging as a revolutionary advancement. BIM's shift from traditional 2D design methods to sophisticated 3D modeling offers a comprehensive digital representation of a building's physical and functional characteristics, promising enhanced efficiency, reduced errors, and improved collaboration among project stakeholders. However, the adoption of BIM, particularly in the context of designing cabin hospitals, presents unique challenges such as the need for precise coordination among diverse aspects, integration of complex medical requirements.

This research addresses these challenges by developing an ontology-based automatic layout design method aimed at enhancing the resilience and efficiency of cabin hospitals. The proposed framework leverages ontology to encapsulate the relationships and attributes of essential components within the BIM environment, facilitating a more robust, flexible, and efficient design methodology.

This project has demonstrated significant improvements in design efficiency, reduced trials and errors, and control construction and operational costs at the early stage compared to traditional methods. The research contributes to the broader field of construction management and healthcare facility design by providing a practical, ontology-based solution to the complex challenges of designing cabin hospitals.

This thesis provides valuable insights and practical solutions for the design of cabin hospitals, emphasizing the importance of integrating advanced ICT tools like BIM with innovative ontology-based frameworks. The proposed approach promises to set a novel method in healthcare facility design, ensuring efficiency of project delivery.

## Acknowledgements

First and foremost, I would like to show my deepest gratitude to my supervisor, Dr. Qing-ping Yang, a respectable, responsible and resourceful scholar, who has provided me with valuable guidance in every stage of the writing of this thesis. Without his enlightening instruction, impressive kindness and patience, I could not have completed my thesis. His keen and vigorous academic observation enlightens me not only in this thesis but also in my future study.

I shall extend my thanks to Dr Tao Zhang all his kindness and help, especially during the period of learning computer programming, comprehensive and detailed explanations he give. I would also like to thank Dr Yuanhong Zhao. She made me feel the warmth of family-like even in this far foreign country.

Last but not least, I' d like to thank Mr. Jeremy Glen Atony Thomas, my partner, his stable mood and special sense of humour always help me get out of the depression and meaningless self-loathing quickly.

# Contents page

## List of Figures

# List of Tables

# Acronyms

**ICT** Information and Communication Technology

**BIM** Building Information Modeling

**CAD** Computer-Aided Design

**CSP** Constraint Satisfaction Problems

**IFC** Industry Foundation Classes

**OWL** Web Ontology Language

**PM** Project Management

**IOIS** Inter-organizational information system

**WCEs** Web-Collaborative Extranets

**DMS** Document Management Systems

**AEC-FM** Architecture, Engineering, Construction, and Facility Management

**IDM** Information Delivery Manual

**MVD** Model View Definition

**XML** Extensible Markup Language

**HTML** Hypertext Markup Language

**RDF** Resource Description Framework

**KSL** Knowledge Systems Laboratory

**SKME** Simple-Knowledge Engineering Methodology

**SW** Semantic Web

**SPARQL** SPARQL Protocol and RDF Query Language

**UMO** Urban Morphology Ontology

**SEMANCO** Semantic Tools for Carbon Reduction in Urban Planning ontology

**UML** Unified Modeling Language

**ICU** Intensive Care Unit

**EIS** Enterprise Information System

**GO-RRB** Gene Ontology

**DO** Disease Ontology

**BOT** Building Topology Ontology

**OWL-DL** Web Ontology Language Description Logic

**CQs** Competency Questions

**UUID** Universally Unique Identifier

**GUID** Globally Unique Identifier

# Chapter 1 Introduction

## 1.1 Background

Conventional layout design within the construction sector is typically performed manually by architects, with computers primarily functioning as tools for modeling, printing, and similar tasks. Nonetheless, the exploration of computer-assisted layout design dates back to the 1960s (Whitehead & Eldars, 1965) leading to the development of numerous methods and applications grounded in diverse design mechanisms.

The incorporation of information and communication technology (ICT) has inaugurated a new era in project management, characterized by the implementation of advanced management tools and technologies (Taxén & Lilliesköld, 2008). The construction industry, in particular, has seen a transformation over the past two decades through the integration of ICT. One such innovation is building information modeling (BIM), a revolutionary computer-aided design (CAD) paradigm that has gained significant traction in both industry and academic circles (Succar, 2009).

The genesis of 3D modeling can be traced back to the 1970s, building upon the early CAD successes in various industries. Numerous industries have since created integrated analysis tools and object-oriented parametric modeling, which are fundamental to the concept of BIM. Nevertheless, the construction sector has been comparatively slow to embrace these innovations, continuing to depend on conventional 2D design techniques. (Eastman, Charles M. et al., 2011).

In the realm of building layout design, various factors must be concurrently considered, leading to heightened computational complexity. These factors encompass the geometry and topology information of the building's internal spaces.

Building Information Modeling (BIM) presents a viable solution to these challenges by offering a digital representation of a facility's physical and functional characteristics (Kang & Choi, 2015a). This digital model serves as a centralized platform for project planning, design, construction, and operation, facilitating collaboration and streamlining processes. The adoption of BIM in the construction industry has the potential to revolutionize project management, improving efficiency, reducing errors, and enhancing overall project outcomes (Maunula & Smeds, 2008).

Despite its advantages, the adoption of BIM, particularly in the context of designing resilient cabin hospitals, presents unique challenges. These include the need for precise coordination among diverse stakeholders, the integration of complex medical requirements, and the extremely compressed time from design and construction to use. Existing design methodologies often fall short in addressing these complexities, leading to inefficiencies, increased costs, and suboptimal patient care environments. Therefore, there is an urgent need for a more flexible, and efficient design methodology. This research aims to develop an ontology-based automatic layout design approach to enhance the efficiency of resilient cabin hospitals.

## 1.2 Research problems statement

Problem 1: In current layout design processes for cabin hospitals is complex and inefficient.

Since the global outbreak of the novel coronavirus, many countries have renovated and built cabin hospitals(Chen, Chen et al., 2022). In the project preparation stage, stakeholders from relevant aspects were brought together, for example, design, construction, material transportation, and medical personnel mobilization were all carried out simultaneously(Luo, H. et al., 2020). This process is very complicated and requires strong and precise coordination and arrangement. Any problem in any link will have serious consequences. Although the application of BIM is facing these challenges, its implementation and promotion are still not smooth. This can be known from the open source drawings of cabin hospitals, most of which are still completed using Computer Aided Design (CAD). In the design stage, although the designers drew a lot from previous experience, which shortened the design time and reduced the design difficulty, the main way to contribute to timeliness was the division of labour and collaboration of a large number of designers to draw drawings. Such a short-term investment cannot be considered effective.

Problem 2: Difficulties in controlling costs and time in designing and building cabin hospitals.

Cabin hospital is a temporary medical facility, which means that the hospital needs to be demolished or abandoned after the operation ends(Liu et al., 2024). The notable feature of these projects is that the construction is very fast, but the rapid completion and commissioning is achieved by huge human, material and financial resources. In the early stages of design, the recycling and reuse of equipment and building components must be considered, and those cases that have been implemented just prove that the cost of the reconstruction or construction of cabin hospitals was huge. For example, ventilators are important facilities for maintaining vital signs of critically ill patients. During the epidemic, the demand for ventilators in various regions increased sharply. If they are obtained through requisition, it will affect patients with other diseases. If they are purchased, these ventilators will be hoarded after the demand for ventilators shrinks after the epidemic ends, which not only increases the storage cost but also causes waste of equipment loss.

Problem 3: Heterogeneity and fragmentation of knowledge data caused by multi-party collaboration.

In Problem 1, the involvement of multiple parties was mentioned as a cause of project complexity. Information originates from various sources such as different departments, teams, suppliers, and partners, each with its own unique format, including text documents, spreadsheets, databases, images, videos, etc., which greatly increases the difficulty of knowledge integration. Organizations may use multiple disparate software systems and platforms that lack effective interoperability and data sharing mechanisms(Liu et al., 2024). Information is stored in various independent systems and devices without centralized management and coordination, making it difficult to integrate and utilize information on a global scale. Multiple versions of the same information may be generated by different systems and personnel, resulting in redundancy and inconsistencies, further complicating integration efforts. Addressing information fragmentation requires a multi-faceted approach involving technology, management, and organizational culture. This includes establishing unified information management standards and data exchange mechanisms, enhancing interoperability between systems, and promoting centralized management and sharing of information.

## 1.3 Aims and objectives

The primary aim of this research is to develop an ontology-based automatic layout design approach for cabin hospitals. This approach seeks to enhance the efficiency, accuracy, and resilience of the design process, ensuring that the resulting layouts are both functionally effective.

The first object in this research involves identifying the key components and requirements for cabin hospital layouts. This encompasses a thorough examination of both functional needs, such as medical equipment placement, and staff accessibility, as well as spatial position relationship including health and safety standards. By clearly defining these essential components, the research aims to create a robust foundation to guide the layout design.

The second object, the development of the space ontology model is undertaken. An ontology, in this context, refers to a structured representation of knowledge that defines the relationships and attributes of the identified components. This involves mapping out the various elements of cabin hospital layouts, detailing how they interact and depend on each other. Many classic cabin hospital cases are referenced in this process. The space ontology serves as a comprehensive model that guides the automatic layout design, ensuring all necessary factors are considered and correctly integrated.

IFC-space ontology is a collection of relations and attributes, which cannot perform layout design by itself. It is more appropriate to describe this ontology with a design manual that contains various knowledge. Therefore, automatic layout design using mathematical models and Constraint Satisfaction Problems (CSP) is also one of the objects.

The proposed model is then validated and tested through format validation and case studies and simulations. These practical applications are crucial for assessing the model's effectiveness and real-world applicability. By simulating various scenarios and analysing the outcomes, researchers can identify potential improvements and ensure the model performs reliably under different conditions. This step is vital for demonstrating the practical benefits and robustness of the ontology-based approach.

Finally, the implement of the ontology-based design method is evaluated from both technical and financial perspectives. This involves comparing the new approach to traditional methods in terms of efficiency, accuracy, and cost-effectiveness. By analysing metrics such as time savings error reduction, and overall design quality, the research aims to highlight the advantages of the ontology-based approach.

## 1.4 Scope of the research

This research encompasses three critical aspects:

1) Theoretical Model: This involves the development of an ontology-based theoretical model tailored to the unique needs of cabin hospitals. The model will define the ontology's structure, components, and relationships.
2) Technological Integration: This aspect involves the integration of the ontology model with existing BIM technologies to enhance design efficiency and accuracy. It includes converting the BIM model into an ontology language model and integrating the two ontologies, as well as using algorithms to visualize the layout results.
3) Case Studies: The proposed approach will be applied in real-world scenarios to validate its practical utility and effectiveness. This will involve selecting representative projects, implementing the ontology-based design approach.

## 1.5 Organization of thesis

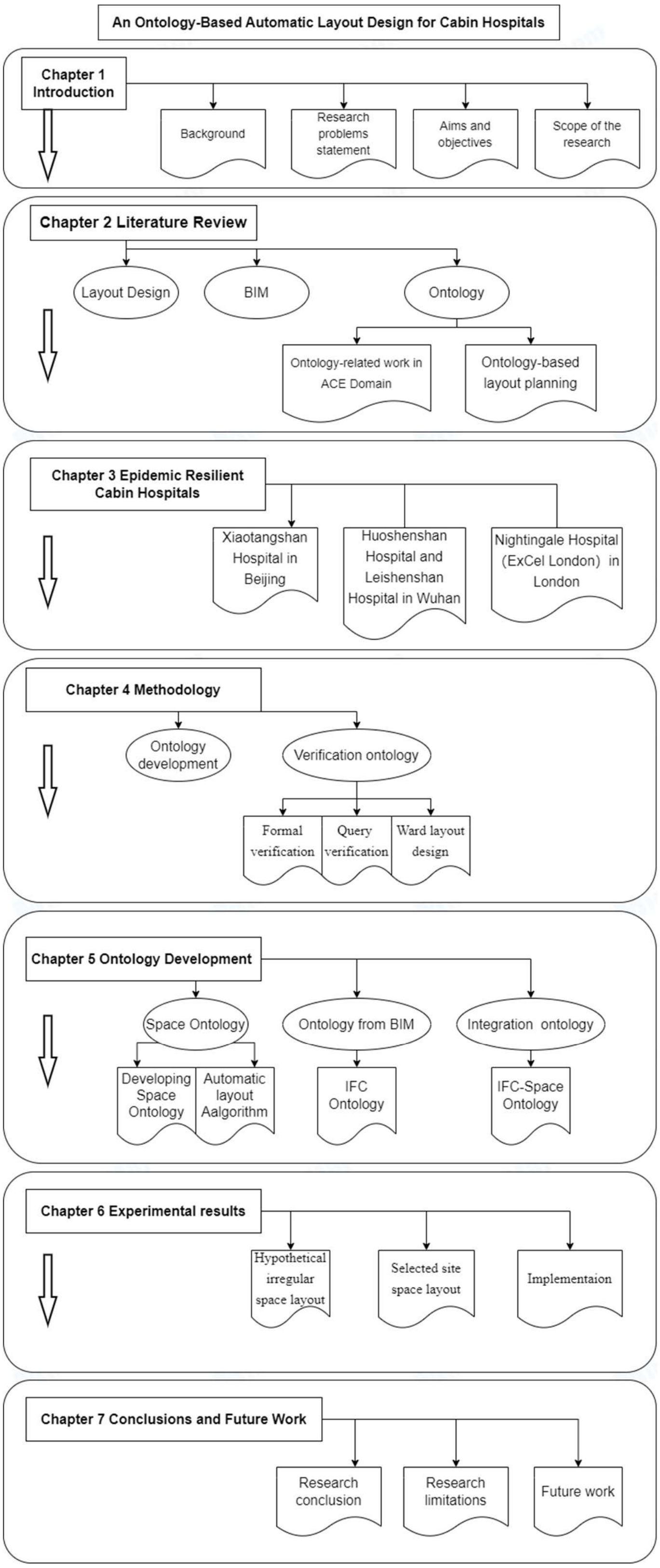



Figure 1.1 Thesis structure

Figure 1.1 is a structural representation of the present thesis.

Chapter 1 provides the background, problem formulation, aims and objectives, scope of the research, and organization of the thesis.

In Chapter 2, existing literature on BIM, ontology, and layout design methodologies are critically reviewed, highlighting gaps and challenges. It covers the evolution of BIM, the role of ontology in construction, and current practices in layout design.

Three types of epidemic resilience cabin hospitals that have been implemented are reviewed in Chapter 3. By analysing the construction and operation of these hospitals, we can deepen our understanding of the layout of cabin hospitals and provide a reliable basis for the subsequent establishment of the spatial entity of cabin hospitals.

It details the research methodology in Chapter 4, including the development of the ontology framework and its integration with BIM. This chapter covers the research design, data collection methods, and analytical techniques used to develop and validate the framework.

Ontology Development process is discussed in Chapter 5, including the creation, integration and validation of the ontology-based layout design framework. Firstly, a spatial ontology of a cabin hospital was established through the logical relationships between entities. Secondly, the cabin hospital BIM model with IFC file is converted into the ontology format which contains more instances. Thirdly, the IFC-Space ontology integrating the two ontologies was established, which not only represents the spatial position relationship but also includes the building components. Finaly, the mathematical model of the cabin hospital spatial layout using the CSP algorithm under the guidance of this space ontology is used for automatic layout design.

Chapter 6 focuses on the experimentation and implementation of the ontology-based automatic layout method proposed in this study.

Chapter 7 discusses the entire study in conjunction with the results of the case experiment and implementation, and points out the limitations of the research. Furthermore, future work is proposed to address these limitations.

# Chapter 2 Literature Review

## 2.1 Layout design

### 2.1.1 General building layout

In the disciplines of architecture and construction, the art of architectural layout design holds paramount significance. It is through the meticulous craftsmanship of architectural layouts that a sense of order and harmony is brought to the organization, ensuring that each element of the design serves its intended purpose with efficiency and elegance (Rahbar et al., 2022a). These layouts are the blueprints that dictate how individuals will engage with and move through a structure, enhancing the overall user experience and ensuring that the built environment is accessible and user-friendly.

Moreover, a robust architectural layout design is not merely about aesthetics or functionality; it is also a matter of life safety . A well-considered design incorporates comprehensive safety measures, including the strategic placement of emergency exits, the creation of unobstructed circulation pathways, and strict adherence to building codes and regulations. These considerations are vital for the protection of occupants and visitors, providing a sense of security and peace of mind in the event of fires, natural disasters, or other emergency scenarios.

The importance of architectural layout design extends beyond the immediate functionality and safety of a building (Wang et al., 2023). It also has profound implications for the overall sustainability and energy efficiency of the structure. By thoughtfully planning the layout, architects can optimize natural light penetration, improve ventilation, and reduce the need for artificial heating and cooling, thereby contributing to a more environmentally friendly and cost-effective building.

In conclusion, the craft of architectural layout design is a multifaceted discipline that encompasses aesthetics, functionality, safety, and sustainability. It is the cornerstone of successful architectural practice, ensuring that the built environment not only meets the needs of its users but also stands as a testament to the harmonious coexistence of form and function(Jia et al., 2023a). Effective architectural layouts minimize space wastage and optimize resource utilization. They can lead to reductions in energy consumption, construction costs, and maintenance expenses. Furthermore, the flexibility of architectural layout design should not be underestimated. A thoughtful layout can adapt to ever-evolving needs over time. Whether it involves reconfiguring office spaces, expanding healthcare facilities, or adapting residential spaces for growing families, flexibility remains of paramount importance.

For the end-user, the aesthetic appeal of a building is not just an incidental feature; it is a fundamental aspect that directly influences their experience within the space. The art of architectural layout design plays a pivotal role in enhancing the overall visual attraction of the structure, creating a sense of harmony and appeal that can elevate the occupant's emotional connection to the building.

Several key factors contribute to the aesthetic impact of a building's layout design. The spatial arrangement, for instance, is crucial in creating a visually pleasing environment. The flow of natural light, carefully planned through the placement of windows and skylights, can illuminate the space, creating a warm and inviting atmosphere. The integration of architectural features, such as ornate columns, graceful arches, or elegant balconies, can add a touch of elegance and sophistication to the design, making the building stand out from its counterparts.

In addition to aesthetics, sustainable architectural layout design is increasingly important in today's environmentally conscious world. A well-designed layout can contribute to the reduction of the building's environmental impact. This is achieved through various means, such as the correct orientation of the building for passive solar heating, which harnesses the sun's energy to warm the building naturally. Efficient utilization of water and energy resources, as well as the incorporation of eco-friendly materials, are also factors that contribute to a sustainable design.

Furthermore, the layout design significantly influences the health and well-being of the building occupants. Access to natural light, which has been shown to improve mood and productivity, is a crucial element in the design. Natural ventilation, which allows for the circulation of fresh air, is also important for the occupant's health. The provision of spaces for rest and relaxation, such as tranquil gardens or comfortable seating areas, contributes to the overall comfort of the occupants, enhancing their well-being and satisfaction with the space.

In summary, the architectural layout design is a multifaceted discipline that encompasses not only the visual appeal of a building but also its sustainability and the well-being of its occupants. A thoughtfully designed layout can create a building that is not only aesthetically pleasing but also environmentally friendly and conducive to the health and happiness of those who use it. Carefully planned layouts can also improve the economic viability of a building. Efficient space utilization, rental potential, and the capacity to meet market demands are all influenced by the layout.

A large amount of related research (Michalek et al., 2002; Naik & Kallurkar, 2016; Wong & Chan, 2009; Yong & Chibiao, 2022) suggests that traditional layout design methods have certain limitations or challenges.

Traditional design methods in architecture are often hampered by inefficiency, as they rely heavily on manual drawing and adjustments, leading to time-consuming and labor-intensive processes (Yang et al., 2024). This sluggishness becomes particularly problematic when architects aim to explore multiple design alternatives, as the effort required to iterate and refine ideas manually can stifle creativity and slow progress. Furthermore, conventional practices impose limitations on the scope of design exploration, restricting architects' ability to consider a broad range of layout options due to the inherent constraints of manual drafting techniques (Guo & Li, 2017). This rigidity is compounded by the repetitive nature of traditional workflows, where even minor design modifications may necessitate redrawing entire floor plans, a tedious and error-prone task that diverts time from more meaningful creative work.

The inflexibility of traditional approaches also creates challenges in adapting to evolving project demands. Fixed designs with limited adaptability hinder architects' capacity to respond dynamically to shifting client requirements, site conditions, or sustainability goals(Suter et al., 2014). This inflexibility is exacerbated by the difficulty of optimizing multiple design objectives simultaneously, such as balancing traffic flow, daylighting, and facade aesthetics. Traditional methods often prioritize singular goals, resulting in compromises that may overlook holistic efficiency or user experience (Yang et al., 2024). Additionally, managing complex(Knotten et al., 2015), non-standard architectural designs manually becomes increasingly unwieldy, and poor management of early design phases leads to document deficiencies, rework, and increased costs. Also, its mix of creative and structured tasks, involving four key interdependencies: pooled, sequential, reciprocal, and intensive. Early stages require iterative, flexible approaches for innovation, while later phases need structured planning for execution, managing this balance is challenging.

Another critical shortfall lies in the limited reusability of traditional designs. Once finalized, these plans are often locked into a static form, making them difficult to adapt for new contexts or repurpose for future projects(Michalek et al., 2002). This lack of modularity not only wastes resources but also stifles innovation by discouraging iterative improvement. Finally, traditional practices can inadvertently constrain creative freedom, as the labor-intensive nature of manual drafting and the pressure to avoid revisions may deter architects from pursuing bold, unconventional ideas. These cumulative limitations highlight the growing need for more adaptive, technology-driven approaches to overcome the barriers inherent in conventional architectural design.

### 2.1.2 Methods of layout design

There are various methods and approaches to building layout design that have evolved and adapted to contemporary needs and technological advancements.

Computer-Aided Design (CAD) has become an indispensable tool in modern building layout design, revolutionizing how architects and engineers conceptualize and develop structures. CAD software enables the creation of precise 2D drafts and detailed 3D models, providing a digital canvas for designers to experiment with spatial arrangements, structural elements, and aesthetic features. Unlike traditional hand-drawn blueprints, CAD allows for rapid modifications, reducing errors and saving time in the design process (Ibrahim & Pour Rahimian, 2010). Advanced features such as layer management, dynamic blocks, and real-time rendering help streamline workflows, ensuring accuracy in dimensions, materials, and construction details.

Popular CAD platforms like AutoCAD, Revit, and SketchUp offer specialized tools for architectural design, including automated floor plan generation, elevation modeling, and cross-sectional views(Leach et al., 2000). These programs also support interoperability with other digital tools, allowing seamless integration with Building Information Modeling (BIM) systems, structural analysis software, and energy simulation tools(Doukari & Greenwood, 2020; Yori et al., 2019). Additionally, cloud-based CAD solutions facilitate remote collaboration, enabling multiple stakeholders to review and edit designs in real time.

Beyond basic drafting, modern CAD systems incorporate parametric modeling, enabling designers to establish relationships between different components(Bhooshan, 2017). For example, changing a room's dimensions can automatically adjust wall placements, door openings, and even structural supports. Some CAD software also integrates AI-driven features, such as automated space optimization and generative design, which suggests layout alternatives based on predefined constraints like square footage, lighting, and circulation patterns (Wang et al., 2023).

Furthermore, advancements in computational design and automation are pushing CAD toward more intelligent, adaptive systems that can respond to environmental and user-specific needs. With its precision, efficiency, and expanding capabilities, CAD remains a cornerstone of contemporary architectural practice, shaping the future of building layout design.

As urbanization and environmental concerns intensify, flexibility and modular design will play an increasingly critical role in creating resilient, efficient, and future-proof buildings. Flexible design emphasizes spaces that can be easily reconfigured, often through open floor plans, movable partitions, and multi-functional areas that allow a single room to serve multiple purposes over time(Xu et al., 2020). This approach is particularly valuable in workplaces, educational facilities, and residential buildings, where shifting demographics, technological advancements, and evolving lifestyles demand adaptable environments. Modular design takes this concept further by incorporating prefabricated, standardized components that can be assembled in various configurations, offering scalability and efficiency. Prefabricated units, such as Mobile Cabin Hospital, are manufactured off-site and installed quickly, reducing construction waste and project timelines while allowing for future expansion or reconfiguration. The benefits of these approaches are far-reaching, including enhanced sustainability through reduced material waste, cost savings from minimized renovations, and improved user experience as spaces evolve alongside occupants' needs. Real-world applications range from co-living spaces with convertible layouts to modular healthcare facilities that can be rapidly adjusted for different medical demands

Building Information Modelling (BIM) is a digital model representing a building's physical and functional attributes (Eastman, Charles M., 2011). It improves collaboration among project stakeholders, facilitating enhanced planning, design, construction, and management of buildings. GreenBIM, which is combined with green buildings, is also a new popular trend (Lu et al., 2017), designers aim to maximize energy efficiency, use environmentally friendly materials, and reduce a building's carbon footprint.

Advanced simulation and analysis have revolutionized building layout design by enabling data-driven decision-making and performance optimization before construction begins. These sophisticated digital tools allow architects and engineers to rigorously evaluate various aspects of a building's design through virtual modeling and predictive analytics, such as computational fluid dynamics (CFD) models analyze airflow patterns to improve natural ventilation strategies and indoor air quality for hosptials (Tsang et al., 2023), patient flow analysis help to optimize hospital layout design (Chen, Xingren et al., 2024). These advanced simulations often integrate with BIM platforms, creating a comprehensive digital twin that allows for iterative refinement of designs based on multiple performance parameters. By leveraging these technologies, designers can predict and enhance a building's real-world performance, mitigate potential issues early in the design process, and create spaces that are not only aesthetically pleasing but also functionally superior, sustainable, and responsive to human needs.

Please note that the status of building layout design methods can continue to evolve rapidly due to technological advancements, changing social and environmental priorities, and architectural trends. It's important to stay updated on the latest developments and best practices in the field of architecture and design.

### 2.1.3 Healthcare facility layout design

In recent years, research on the spatial organization of healthcare facilities has gained increasing attention (Benitez et al., 2019). Healthcare facility, especially hospitals, rank among the most complexly designed buildings, meticulously planned to meet the multifaceted demands of healthcare. Their layout profoundly influences key operational factors, including efficiency, patient outcomes, staff satisfaction, and the overall standard of medical services provided. Ulrich and his team emphasized the pivotal role of physical design in healthcare settings, demonstrating that well-planned facilities can significantly enhance service quality (Ulrich et al., 2008). Infection control is another critical consideration, requiring layouts that minimize cross-contamination risks, ensure proper ventilation, and include isolation areas. Hospital layout directly impacts key factors such as nurse time allocation, wayfinding efficiency, patient flow, and the likelihood of overcrowding. As a result, effective hospital design is essential, not only for addressing operational challenges but also for maintaining high standards of patient care.

Designing healthcare facility layouts involves numerous complexities and challenges due to the unique requirements of the healthcare environment (Parsia & Tamyez, 2018). One of the primary difficulties is ensuring regulatory compliance, as healthcare facilities must adhere to strict building codes, accreditation standards, and infection control guidelines. Additionally, these facilities house a wide range of specialized departments, such as emergency rooms, surgical suites, radiology, and laboratories, each with distinct spatial and operational needs. Coordinating these diverse functions into a cohesive and efficient layout is a highly complex task.

Beyond its foundational importance, healthcare facility design entails a highly intricate and multifaceted process. As Hicks's work emphasize, this complexity extends far beyond the mere organization of physical spaces, it also encompasses the management of dynamic flows, including patients, staff, visitors, equipment, and information (Hicks et al., 2015). Spatial configuration plays a critical role in determining operational effectiveness, influencing movement patterns, accessibility, and even human behavior and interactions. Consequently, a major challenge in hospital design lies in analyzing how these spatial arrangements shape decision-making, mobility, and the broader efficiency of workflows and activities (Sopher et al., 2016).

Jia and his team  systematically reviewed decision support methods for hospital layout design, employing spatial network analysis and simulation modelling (Jia et al., 2023b). Their study identifies a key gap in connecting hospital design challenges directly to spatial configurations and calls for standardized layout representation and evaluation methods. The findings stress the need to align design challenges with suitable assessment techniques to enhance layout decision-making. Jamali's team conducted the first comprehensive architectural analysis of the topic, proposing a framework to help architects and healthcare designers critically evaluate Hospital Layout Problems and their associated ethical implications (Jamali et al., 2020).

Scalability and flexibility are also significant challenges, as healthcare facilities must adapt to changing patient volumes and technological advancements (Cubukcuoglu et al., 2021). Also, Fogliatto's case based on lean-oriented  hospital layout design, which pointed out that budget control often exacerbate these challenges, forcing difficult trade-offs between cost and quality of care (Fogliatto et al., 2019).

Their experience and understanding of the unique challenges and requirements of the healthcare sector are invaluable in creating functional, safe, and efficient healthcare environments.

In conclusion, the design of healthcare facility layouts is a multifaceted process that demands a comprehensive consideration of various factors. It is a field that requires expertise, experience, and a keen eye for detail to ensure the creation of spaces that meet the diverse needs of healthcare providers and patients alike.

# 2.2 BIM

## 2.2.1 BIM Background

The rapid advancement of information and communication technology (ICT) has brought about significant changes in project management practices, utilizing the latest developments in management tools and technologies. (Taxén & Lilliesköld, 2008). Over the past two decades, ICT in the construction industry has experienced substantial transformation. Building Information Modelling (BIM), as a new computer-aided design (CAD) paradigm, has been extensively adopted in both industry and academia. (Succar, 2009). The development of 3D modeling started in the 1970s, building on early computer-aided design (CAD) advancements across various industries. While many sectors have developed integrated analysis tools and object-based parametric modeling, which form the foundation of BIM, the construction industry has largely remained reliant on traditional 2D design for a long time. (Eastman, Charles M. et al., 2011; Gray et al., 2013).

As previously mentioned, projects are becoming increasingly complex and challenging to manage, particularly in the construction sproject. (Alshawi & Ingirige, 2003; Chan et al., 2004; Williams, 2013). One type of complexity stems from the intricate interdependence among a wide range of stakeholders. This includes financial institutions that provide the necessary funding, regulatory authorities that ensure compliance with laws and standards, and architects and engineers who design and plan the projects. Additionally, lawyers are involved in navigating legal aspects, while contractors and suppliers are responsible for the execution and provision of materials. Furthermore, related industries contribute specialized services and products, all of which must be seamlessly integrated to achieve successful project completion. The coordination and collaboration required among these diverse entities add significant layers of complexity to the project management process (Sears et al., 2008).

Complex construction projects necessitate inter-organizational collaboration. This involves forming alliances among various entities, such as contractors, subcontractors, suppliers, and consulting firms. These associations enable the pooling of resources, expertise, and technology, facilitating the efficient execution of multifaceted tasks. Effective inter-organizational cooperation ensures that each party's contributions are seamlessly integrated, addressing the intricate demands of construction projects. This collaborative approach is essential for managing the diverse aspects of project planning, design, execution, and completion, ultimately leading to more successful and streamlined project outcomes. (Maurer, 2010). To ensure the successful management of inter-organizational project risks, trust among various project partners is regarded as a critical success factor. Given the collaborative nature of these inter-organizational ventures, it is widely recognized that improved integration, cooperation, and coordination within construction project teams are essential. Effective collaboration among diverse stakeholders, such as contractors, subcontractors, suppliers, and consulting firms, hinges on establishing and maintaining trust. This trust facilitates open communication, reduces conflicts, and enhances the overall efficiency of project execution. Consequently, fostering a culture of trust and teamwork is imperative for achieving the seamless integration of efforts, ultimately contributing to the success of complex construction projects (Cicmil & Marshall, 2005). An Inter-Organizational Information System (IOIS) offers a viable solution to address the integration, cooperation, and coordination challenges prevalent in the construction industry. By enabling seamless communication and data sharing among various stakeholders, an IOIS enhances collaboration and ensures that all parties are aligned with project goals. This system facilitates the efficient exchange of information between contractors, subcontractors, suppliers, and consultants, thereby improving decision-making processes and reducing the likelihood of errors and misunderstandings. Implementing an IOIS can streamline project workflows, promote transparency, and foster a more cohesive project environment, ultimately leading to more successful and efficiently managed construction projects. (Maunula & Smeds, 2008). IOIS also known as a web-based Project Management (PM) system, serves as a comprehensive platform for enhancing collaboration and coordination in the construction industry. These systems leverage internet technologies to provide real-time access to project data, facilitating seamless communication among stakeholders such as contractors, subcontractors, suppliers, and consultants. By centralizing information and making it accessible from anywhere at any time, web-based PM systems improve transparency, streamline workflows, and support more effective decision-making. Adopting these systems can greatly minimize project delays, lower error rates, and enhance overall project efficiency, thereby leading to more successful project outcomes (Nitithamyong & Skibniewski, 2004). Examples of such systems include Web-Collaborative Extranets (WCEs) and Document Management Systems (DMS) (Ajam et al., 2010). Regardless of the terminology employed, these systems play a pivotal role in enabling the accurate and timely dissemination of various types of information. This capability is essential for the successful execution of projects, as it ensures that all stakeholders have access to up-to-date and reliable data. By facilitating seamless communication and information exchange, these systems help to coordinate efforts, streamline workflows, and mitigate risks. This, in turn, enhances decision-making processes and contributes to the overall efficiency and effectiveness of project management, ultimately leading to more successful project outcomes. (Anumba et al., 2008). Document-based work implies an "unstructured flow of text and graphic elements" throughout the project life cycle. This approach can lead to inefficiencies, as information is often scattered across various documents and formats, making it difficult to manage and retrieve. Consequently, the lack of structured data flow can hinder effective communication and coordination among project stakeholders. By relying on disparate and unorganized documents, the project is susceptible to errors, misunderstandings, and delays. Therefore, transitioning to a more structured and integrated information management system can significantly enhance project efficiency and success (Nisbet & Dinesen, 2010). This unstructured process poses a significant challenge to improved integration practices, as the information exchanged at the document level is frequently "ambiguous, unformatted, or difficult to interpret." Such disorganized communication can lead to misunderstandings, errors, and inefficiencies, impeding the smooth collaboration among project stakeholders. The lack of clarity and standardization in document-based information exchange makes it challenging to ensure that all parties have a consistent understanding of project details. Addressing this issue by adopting more structured and standardized information management practices can greatly enhance the accuracy, clarity, and efficiency of communication, ultimately leading to better project outcomes. (Ajam et al., 2010). Ajam et al. assert that the optimal use of IOIS involves transitioning from document-sharing practices to sharing information at the object or element level. Consequently, BIM could be a crucial approach for ensuring this integration, shifting from the document-based paradigm to an integrated database paradigm.

To address the growing complexity and challenges of project management, BIM has rapidly evolved and become widely adopted. BIM is now the most prevalent term for embracing new approaches in building design, construction, and maintenance. It is defined as “a set of interacting policies, processes, and technologies that create a methodology to manage essential building design and project data in digital format throughout the building's life cycle.”(Succar, 2009). Additionally, BIM is utilized for various individual components of smaller projects. Implementing 3D BIM entails that all project and asset information, data, and documents be maintained in electronic form. Furthermore, both the public and private sectors in the United States are working together to encourage the adoption of BIM (Underwood & Isikdag, 2011)However, some argue that the case for BIM has not been entirely established, and the overall effectiveness of BIM utilization remains somewhat questionable. Despite its widespread adoption and the benefits it promises, there are still concerns regarding its practicality and efficiency in real-world applications. Critics point out that comprehensive data on BIM’s performance and return on investment is limited, and more empirical evidence is needed to fully validate its advantages. Consequently, while BIM has shown significant potential, its complete efficacy and justification are still under scrutiny (Jung & Joo, 2011). Succar's (Succar, 2009) definition of BIM emphasizes its comprehensive nature, encompassing not only software for geometric modeling and information input but also tools and processes associated with project management (PM). This holistic perspective firmly situates BIM within the realm of architectural project management. As a result, construction project managers can leverage BIM to enhance stakeholder collaboration, shorten the time needed for project documentation, and achieve more favorable project outcomes. By integrating BIM into the project management workflow, managers can streamline processes, improve accuracy, and foster a more efficient and cooperative project environment.

The advantages of BIM across various types of construction projects are numerous and widely acknowledged by relevant stakeholders (Gu & London, 2010). BIM offers a range of benefits, including improved collaboration and communication among project participants, enhanced accuracy and efficiency in design and construction processes, and better project visualization. Stakeholders, such as architects, engineers, contractors, and owners, recognize that BIM facilitates more informed decision-making, reduces errors and rework, and streamlines project timelines. Additionally, BIM's ability to integrate diverse data sources and provide comprehensive project insights makes it an invaluable tool for optimizing resource management and ensuring higher-quality outcomes. The widespread recognition of these benefits underscores BIM's transformative impact on the construction industry. Despite its significant technical advantages and potential value, the global adoption of BIM remains limited, falling short of realizing its full capabilities. BIM modeling was first introduced in pilot projects during the early 2000s (Penttilä et al., 2007) to aid architects and engineers in architectural design. While these early initiatives showcased the promise of BIM in enhancing design accuracy and collaboration, widespread implementation has been slow. Various challenges, including resistance to change, high initial costs, and the need for extensive training, have hindered its broader application. Consequently, current research trends are primarily focused on enhancing several key areas: pre-planning and design, conflict detection, visualization, quantification, cost estimation, and data management. Efforts are being made to refine pre-planning and design processes to ensure more accurate and efficient project setups. Advanced conflict detection techniques are being developed to identify and resolve issues early in the project lifecycle. Improved visualization tools are enabling better representation and understanding of project components. Quantification methods are being optimized to ensure precise measurement of materials and resources. Cost estimation practices are being enhanced to provide more accurate financial projections. Lastly, robust data management strategies are being implemented to streamline information handling and accessibility, ultimately aiming to maximize the benefits and effectiveness of BIM in construction projects (Eastman, Charles M. et al., 2011; Wassouf et al., 2006). Recently, specialized tools for design, architecture, and engineering have incorporated essential features such as energy analysis, structural analysis, scheduling, progress tracking, and site safety (Becerik-Gerber et al., 2012). The use of BIM has traditionally concentrated on pre-planning, design, construction, and the overall delivery of buildings and infrastructure projects. However, recent research has expanded its focus beyond the early life cycle (LC) stages to include maintenance, renovation, deconstruction, and decommissioning. This shift recognizes the importance of BIM throughout the entire lifespan of a building or infrastructure project. By incorporating considerations for maintenance and renovation, BIM aids in extending the useful life and optimizing the performance of structures. Furthermore, by addressing deconstruction and decommissioning, BIM contributes to more sustainable practices, ensuring that end-of-life processes are managed efficiently and with minimal environmental impact. This comprehensive approach enhances the long-term value and sustainability of construction projects. (Akbarnezhad et al., 2014; Becerik-Gerber & Rice, 2010; Becerik-Gerber et al., 2012; Eastman, Charles M. et al., 2011; Lucas et al., 2013; Nicolle & Cruz, 2010; Sabol, 2008)Many construction projects still overlook the implementation of BIM. Despite its proven advantages in enhancing design accuracy, improving collaboration, and optimizing project outcomes, numerous projects continue to rely on traditional methods. This neglect often stems from factors such as resistance to change, the perceived high cost of adoption, and a lack of awareness or understanding of BIM's full potential. As a result, these projects miss out on the significant benefits that BIM can offer, including better resource management, reduced errors, and increased efficiency. Addressing these barriers through education, demonstration of ROI, and gradual integration of BIM practices can help increase its adoption and unlock its full potential across the construction industry  (Cao et al., 2014). Although industry reports indicate that the number of practitioners using BIM has increased significantly in certain countries (Bernstein et al., 2012; Lee et al., 2012), its global adoption still falls short of its potential. Many projects remain hesitant or are only beginning to consider the adoption of BIM. This disparity suggests that while awareness and usage of BIM are growing in some regions, a substantial portion of the global construction industry has yet to fully embrace the technology. Challenges such as high initial costs, resistance to change, and a lack of standardized training continue to hinder widespread implementation. Consequently, despite the clear advantages and increasing local adoption rates, BIM's global utilization remains limited, with numerous projects still on the cusp of integrating this transformative technology (Lee et al., 2012; Waterhouse & Philp, 2013).

Buildings and structures vary widely in their use (e.g., residential, commercial, municipal, infrastructure), age (e.g., new, existing, heritage), and ownership (e.g., private owners, housing associations, authorities, universities). These diverse conditions, shaped by stakeholder requirements, influence the application of BIM, its level of detail, and the supporting functions related to design, construction, maintenance, and deconstruction processes. According to a recent survey, BIM is particularly suitable for larger and more complex buildings and is commonly applied to commercial, residential, educational, healthcare, and various other building types. The survey respondents highlighted that the adaptability and comprehensive capabilities of BIM make it an effective tool for managing the intricacies of different project requirements and stakeholder expectations (Becerik-Gerber & Rice, 2010; RICS, 2013). However, since less than 10% of the survey respondents are facility managers, owners, or involved in deconstruction, these trends may not accurately reflect the current use of BIM in existing buildings. This underrepresentation suggests that the data primarily highlights the application of BIM in new construction projects rather than its adoption in the ongoing management or renovation of existing structures. Consequently, while the survey indicates a growing trend in BIM usage for complex and large-scale projects, it does not fully capture the extent of BIM's integration into the lifecycle management, maintenance, and deconstruction of existing buildings. This gap underscores the need for more comprehensive research to understand BIM's utilization across all stages of a building's life.

Although BIM is widely accepted in the construction industry and offers clear benefits, many small companies are reluctant to adopt it. These companies often perceive BIM as relevant only to large construction firms, high-end architects, government projects, or environmentally-focused organizations. A significant barrier to BIM adoption is the lack of understanding and misinformation about its capabilities and benefits.

The most pressing challenge the construction industry faces in implementing BIM is reduced customer demand. Despite companies recognizing the advantages of this innovative approach, clients often do not accept or require BIM services. As a result, many small companies and even some larger firms have not yet integrated BIM into their operations. Companies cannot compel customers to adopt BIM services if they do not see the need or value.

Another issue is the unique nature of each construction project. The industry has traditionally focused on certain types of projects that reliably generate commissions. BIM technology is not universally applicable, especially for projects with unique characteristics where standardized techniques are not feasible. Small organizations encounter this problem more frequently than larger ones. Even though BIM can save time and costs for a wide range of projects, the cost-to-value ratio of implementing BIM may not be advantageous for smaller projects.

Once BIM is more widely adopted, it will be easier to implement these technologies across various project types, resulting in cost and time savings. However, for construction companies handling primarily small projects, the expense and effort of full-scale BIM implementation may not be justified. Consequently, many small organizations avoid BIM due to the limited scale of their projects.

The most significant hurdle for any business organization is the high cost of BIM implementation. The initial expense of upgrading equipment and technology is not the only concern; there are also substantial intangible costs. Implementing BIM involves more than just software upgrades; it requires integrating all elements of business operations, from compliance applications to supplier component specifications, into a single system. Additionally, comprehensive training is necessary. If internal project managers lack the required skills, organizations may need to hire specialized BIM managers, further increasing costs.

Most construction organizations do not have employees who are experts in BIM technology and principles, necessitating the assistance of external experts for issues like integration. This reliance on external expertise can also drive up implementation costs.

### 2.2.2 BIM Data

In a BIM environment, such as Autodesk Revit, users can digitally represent physical elements of a building project (e.g., walls, doors, columns, and slabs) and specify attributes for these elements, including type, location, and size. This detailed information can be used for various building design analyses, such as energy and structural analysis, which can be derived directly from the BIM design model. Consequently, the BIM model aids decision-making across numerous aspects of the Architecture, Engineering, and Construction (AEC) industry, while also boosting labour productivity by minimizing the need for rework when modeling or gathering building data for different purposes.Significant research has been dedicated to integrating BIM models with discipline-specific data and facilitating information transfer between BIM authoring software and specialized design tools. Despite these advancements, adapting BIM for construction management tasks remains challenging. For instance, quantity take-off in conjunction with workface planning, defined as "the process of organizing and delivering all the elements necessary before work starts, enabling craft persons to perform quality work safely, effectively, and efficiently", is one such area where BIM integration is still evolving. By improving the linkage between BIM models and construction management processes, the industry can better organize project elements and streamline the workflow, ultimately enhancing the efficiency and effectiveness of construction activities. However, this requires ongoing research and development to overcome existing barriers and fully realize BIM's potential in construction management. (*About Advanced Work Packaging (AWP)*, 2014).

The lack of information sharing can lead to delays, duplication of efforts, and increased construction costs. To address these challenges, various IT applications have been developed for the construction industry, targeting specific functions such as design (e.g., Autodesk, Bentley), project planning and scheduling (e.g., Microsoft Project, Primavera), cost estimation (e.g., WinEstimator), collaboration (e.g., Project Extranet), and field technologies (e.g., RFID, mobile phones) (O'Brien et al., 2008).While these tools are effective for their intended purposes, most are designed to serve a single business function and do not facilitate seamless information flow across different functions and organizational boundaries. This siloed approach can hinder overall project efficiency and coordination.For example, design software like Autodesk and Bentley excels in creating detailed architectural plans, but it may not integrate smoothly with scheduling tools like Microsoft Project or Primavera. Similarly, cost estimation software like WinEstimator operates independently of collaboration platforms such as Project Extranet, and field technologies like RFID and mobile phones often lack integration with other IT systems used in the project lifecycle.

To overcome these limitations, there is a growing need for integrated IT solutions that enable comprehensive information sharing and collaboration across all phases of a construction project. Such solutions would allow for real-time data exchange, enhance communication among stakeholders, and streamline processes from design through to field operations. Developing and adopting these integrated systems can significantly improve project management, reduce costs, and increase overall productivity in the construction industry.

A BIM platform user is restricted to the platform schema and is unable to define information that is not included in the platform schema (Pauwels et al., 2010).

Projects in architecture, engineering, construction, and facility management

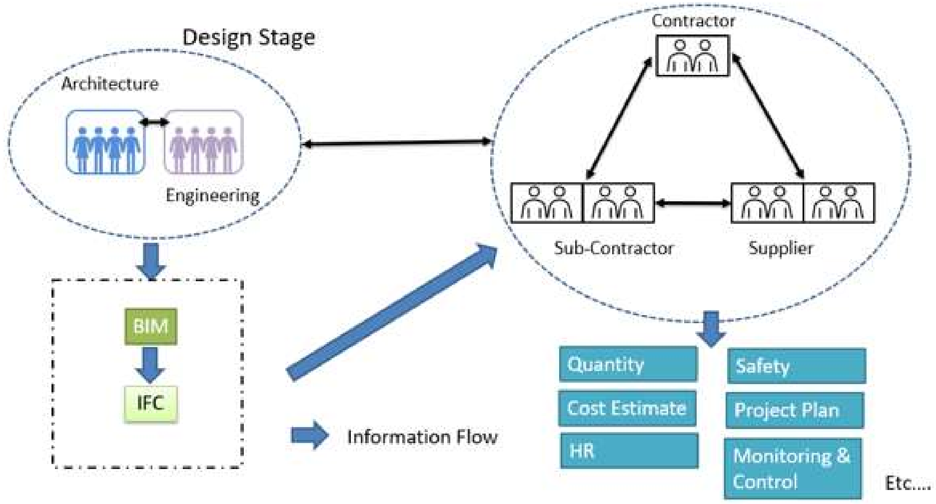



Figure 2.1 Information Flow in BIM

(AEC-FM) involve numerous specialists from various fields collaborating to complete a project. The Figure 2.1 illustrates the information flow in construction projects, highlighting how experts from different domains (e.g., estimating, scheduling, supply chain, and fabrication) interact with a BIM as a centralized 3D model of the project. The process begins with the Design Stage, where architects and engineers use BIM to create a detailed digital model of the project. This model, standardized through IFC, ensures interoperability and efficient data exchange across all stakeholders. Downstream, contractors and subcontractors leverage this information for on-site execution, while quantity surveyors and cost estimators derive accurate measurements and budgets directly from the model. The integrated data also supports HR allocation, supplier coordination, safety planning, and project scheduling, ensuring all teams work from a unified source. Finally, monitoring and control mechanisms use real-time BIM data to track progress, manage resources, and mitigate risks. This holistic approach demonstrates how BIM centralizes information, enhances collaboration, and drives efficiency throughout the project lifecycle, from design to completion. These specialists query the BIM to extract data relevant to their specific areas of expertise and use this information to generate their unique perspectives on the project.

When information about different project views is stored in diverse file formats, integrating data about a building element becomes challenging. It necessitates accessing multiple project documents and manually extracting and combining the required information, which can be time-consuming and prone to errors.

For example, an estimator might need cost data from financial documents, while a scheduler requires timeline information from planning files. Similarly, supply chain managers look for logistics details, and fabrication specialists need precise measurements and material specifications. Each of these professionals must navigate through different file formats and documentation systems, manually synthesizing the data to get a complete picture of the project.

To address these challenges, there is a critical need for integrated information systems that enable seamless data sharing and interoperability across all project domains. Such systems would streamline the process of extracting and consolidating information, reduce the risk of errors, and enhance the efficiency of project management. By leveraging advanced BIM capabilities and fostering greater collaboration among AEC-FM specialists, the construction industry can achieve more cohesive and effective project execution.

### 2.2.3 Industry Foundation Classes

Industry Foundation Classes (IFC) format of BIM data was created to let AEC-FM software programs communicate with one another. IFC is an open and standardized data model developed to facilitate interoperability in the building and construction industry (Froese, 2003). It allows different software applications to exchange and share data with ease, enhancing collaboration among architects, engineers, contractors, and other stakeholders involved in building projects. IFC provides a comprehensive framework for representing various building elements, including geometry, spatial relationships, material properties, and more. By adopting IFC, the industry can ensure that project information is consistent, reliable, and reusable throughout the lifecycle of a building, from design and construction to operation and maintenance. This leads to improved efficiency, reduced errors, and lower costs in managing building information across different platforms and stages.

The IFC schema is comprehensive, covering a wide array of building and infrastructure elements, including geometric data, spatial relationships, material properties, and even behavioural characteristics. This allows for a detailed and holistic representation of the built environment, supporting various stages of a building's lifecycle from design and construction to operation and maintenance. For instance, an IFC file can contain detailed information about walls, windows, doors, structural elements, HVAC systems, electrical systems, and much more, all integrated into a coherent digital model.

One of the primary advantages of using IFC is its vendor-neutrality. Unlike proprietary formats controlled by specific software vendors, IFC is an open standard, which means it is freely accessible and not tied to any particular company. This openness ensures that project data can be accessed and utilized by any compliant software, reducing the risks associated with data lock-in and ensuring long-term data integrity and accessibility.

IFC's role in enhancing collaboration cannot be overstated. By providing a common data environment, IFC enables different stakeholders, architects, engineers, contractors, and facility managers, to work from the same set of data. This reduces errors, omissions, and inconsistencies that often arise from using disparate systems. For example, changes made by an architect to a building design can be immediately reflected in the structural analysis performed by an engineer, ensuring that all parties are working with the most up-to-date information.

Furthermore, IFC supports sustainable practices in the construction industry. By providing detailed information about building components and their interactions, IFC facilitates more accurate energy modelling and simulation, which can lead to better-informed decisions regarding energy efficiency and sustainability. Additionally, the ability to accurately track and manage building data over its entire lifecycle supports more effective maintenance and operation, leading to reduced costs and improved performance over the long term.

The IFC standard is continually evolving, with new versions being released to incorporate advancements in technology and industry practices. These updates expand the range of building types and elements that can be modelled, as well as improve the accuracy and granularity of the information that can be represented. BuildingSMART International, along with its global network of partners, works tirelessly to refine and enhance the IFC standard, ensuring it remains relevant and capable of meeting the needs of the modern AEC/FM industry.

Overall, IFC is a cornerstone of effective BIM implementation, offering a robust, open, and flexible data model that supports interoperability, enhances collaboration, and promotes sustainability in the built environment. By leveraging IFC, the AEC/FM industry can achieve greater efficiency, accuracy, and innovation, ultimately leading to better-designed, constructed, and managed buildings and infrastructure.

### 2.2.4 IFC data conversion

A use case method has been developed to address the limitations of the IFC data model for information transmission (Eastman, C. M. et al., 2010). which needs domain experts to prepare an Information Delivery Manual (IDM) and a Model View Definition (MVD). This method requires domain experts to create an Information Delivery Manual (IDM) and a Model View Definition (MVD). Practitioners must develop an IDM and an MVD that specify the data to be transferred between two applications. While the IDM and MVD approaches provide a static description of the data that can be shared between applications, they do not support rule-based automatic extraction of data from multiple sources. (Kang & Choi, 2015b). To utilize this method effectively, practitioners must meticulously define the data exchange requirements in the IDM and MVD, ensuring that all necessary information is clearly outlined for seamless transmission. However, the static nature of these definitions means that any changes in data requirements or application updates necessitate manual revisions to the IDM and MVD. This lack of dynamic data handling can be a significant drawback, as it does not facilitate the automatic and rule-based extraction of data from various sources, which would otherwise enhance efficiency and reduce the potential for errors in data transmission. Therefore, while the IDM and MVD frameworks are valuable for standardizing data exchange, they fall short in providing the flexibility and automation needed for more complex and evolving project requirements.

Semantic Web technology enables individuals to express data about an entity in a manner that can be seamlessly integrated with information from other sources (Hendler et al., 2020). Consequently, applying semantic representation to the building information model allows anyone involved in a building project to articulate their information about building elements in a way that easily combines with data provided by others. This approach fosters enhanced collaboration and data integration, ensuring that all project stakeholders can contribute and access relevant information efficiently and coherently.

Currently, various sectors within the AEC-FM industry maintain project data in diverse formats (Kang & Choi, 2015b). For instance, BIM data is stored in object formats (such as IFC), XML formats (like ifcXML, gbXML), or relational databases (such as ODBC). In contrast, cost estimates and project progress data are typically housed in relational databases. Meanwhile, material suppliers often provide product data in text, HTML, XML, or relational formats. This heterogeneity in data formats presents challenges for seamless information exchange and integration across the different phases and stakeholders of construction projects.

When data from different sources is stored in heterogeneous formats, computers face significant challenges in integrating this data. This difficulty arises due to several factors:

- The data sources are often local and cannot be shared across computer applications on the Internet.
- Different data sources may use varied terminology to refer to the same entities, or the same words may have different meanings in separate databases.
- Dynamically modifying the database schema is challenging because the

data schema is closely tied to the object model of the application using it.

Each domain tends to develop its own schema to represent domain-specific attributes of the same object. For example, two different class hierarchies might be used in the estimation and scheduling domains to model the same building element, complicating the integration of scheduling and cost attributes. The W3C has extensively discussed the issues related to integrating data stored in relational or object-oriented databases, emphasizing the complexities involved in achieving seamless data integration across diverse systems (Knublauch et al., 2006).

Extensible Markup Language (XML) is a serialization format that addresses several issues associated with enabling different programs and computers to communicate with each other(*Extensible Markup Language (XML),* 2016)The XML Schema specifies the structure of XML documents. However, XML documents defined by this schema lack extensibility (*Introduction to: RDF vs XML ,* 2016). Adding simple attributes to an XML document necessitates rewriting all applications that use the document to ensure they can read the modified version.

Semantic Web technology offers a comprehensive framework that enables data to be shared and reused across application, enterprise, and community boundaries (*W3C SEMANTIC WEB,* 2016). On the World Wide Web, individuals can post opinions on any topic and publish them as web pages. The Semantic Web extends these principles from documents to data, allowing data to interrelate similarly to how documents are already connected (Hendler et al., 2020), (*W3C RDF Working Group, Resource Description Framework (RDF) ,* 2015). Thus, the semantic representation of building information will enable various disciplines involved in AEC-FM projects to express their data about project entities in a manner that can be seamlessly combined with data from other fields (Niknam & Karshenas, 2014, 2015). This approach facilitates better integration and collaboration, enhancing the efficiency and effectiveness of information sharing across different domains.

The Semantic Web employs formal ontologies (*W3C Standard, Ontologies ,* 2015) to define the relationships between concepts (classes) and other concepts. to define the relationship between concepts (classes) and concepts  (Gruber, 1993). Ontology helps define the organization of data distributed on the network by expressing the formal view shared between multiple parties. Ontology is a clear formal specification of concepts and their relationships in the domain  (Gruber, 1993). Ontologies are utilized to create domain knowledge bases, which are information libraries designed to collect, organize, and share information. The Semantic Web enables the creation of a distributed network of interconnected knowledge bases that can reference each other using Uniform Resource Identifiers (URI) (*W3C RDF Working Group, Resource Description Framework (RDF)* , 2015).

The Resource Description Framework (RDF) and Web Ontology Language (OWL) are used to create ontologies and knowledge bases. RDF/OWL data is represented by triples (Antoniou & Van Harmelen, 2004). A set of RDF/OWL triples can be expressed as a graph data structureA collection of RDF/OWL triples can be expressed as a graph data structure. This graphical data structure, combined with data URIs, creates a global information space of interconnected data distributed across the network. Figure 2.2 illustrates an example of how information from two different fields can be integrated.

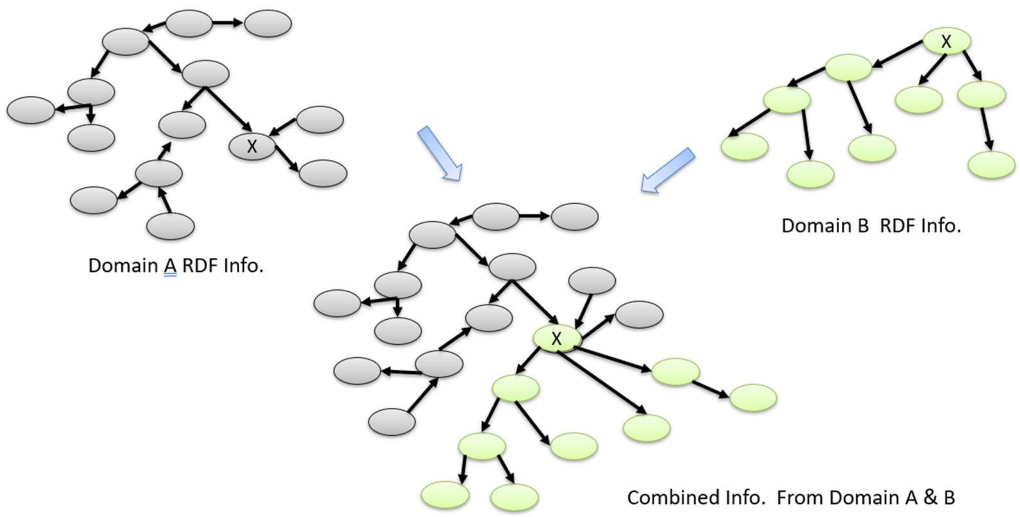



Figure 2.2 Merging process schematic

Domain A and Domain B, using the Resource Description Framework (RDF). Each domain contributes its own structured data in the form of RDF triples, which describe resources through subject-predicate-object relationships. By combining these datasets, the resulting unified information leverages the strengths of both domains, enabling a more comprehensive and interconnected knowledge base. This approach highlights the power of RDF in facilitating semantic interoperability, allowing data from different fields to be linked, queried, and analyzed together. Such integration is particularly valuable in applications like linked data, knowledge graphs, and cross-domain research, where combining diverse datasets enhances insights and decision-making.

## 2.3 Ontology

Ontology, a term derived from the Greek words "onto" meaning "being" and "logos" meaning "word" or "study," is a rich and nuanced concept that has found application in numerous domains. At its core, ontology in philosophy seeks to explore and answer some of the most fundamental questions about existence and reality: What is the nature of being? What constitutes reality? And how do we categorize and understand the world around us? These philosophical inquiries have laid the groundwork for the development of ontological systems that extend beyond academic discourse and into the practical realms of science, technology, and information management.

In the context of information science, artificial intelligence (AI), and the Semantic Web, ontology takes on a different but related meaning. It refers to a structured representation of knowledge that defines the types of entities within a particular domain of interest and the relationships between those entities(Noy, Natalya F. & McGuinness, 2001). This structured knowledge representation is akin to a language or a framework that allows for the precise encoding of concepts and their interconnections.

Ontology modelling is the process of creating these structures. It involves identifying the key concepts (classes), attributes (properties), and relationships within a domain(Sugumaran & Storey, 2006). These models serve as a foundational language for knowledge representation and are critical for data integration, as they provide a common vocabulary and conceptual framework that can be used to harmonize and combine information from disparate sources(Guizzardi, 2005).

The application of ontology in the Architecture, Construction, and Engineering (ACE) industry is transformative. Ontologies provide a structured framework for organizing and interpreting complex information, enabling more efficient data management and enhanced interoperability among various software systems. In the ACE industry, this translates to more cohesive and integrated workflows, as ontologies allow for the seamless exchange of data between different stakeholders and software platforms.

By defining a common set of terms and relationships, ontologies facilitate a shared understanding of project information, which is crucial for collaboration among architects, engineers, contractors, and facility managers. This shared understanding helps to reduce misunderstandings and errors, ensuring that all parties are on the same page throughout the project lifecycle. For instance, an ontology can help standardize the terminology used in BIM models, making it easier to integrate data from different sources and ensuring consistency across different stages of the project.

Ontologies also support advanced data analytics and decision-making processes. By organizing data into well-defined categories and relationships, they enable more sophisticated queries and analyses. This can lead to insights that improve design quality, enhance construction efficiency, and optimize maintenance operations. For example, an ontology-based system can help identify potential conflicts in a building design by understanding the spatial relationships between different components, thereby preventing costly rework during construction.

Moreover, ontologies play a crucial role in the automation of various tasks within the ACE industry. They provide the necessary structure for developing intelligent systems that can automate routine processes, such as code compliance checking, cost estimation, and project scheduling. By leveraging ontologies, these systems can interpret and act upon complex data with greater accuracy and efficiency, freeing up human resources for more strategic tasks.

In addition to improving current processes, the application of ontology paves the way for future innovations in the ACE industry. As the industry increasingly adopts digital twins and smart building technologies, ontologies will be essential for managing the vast amounts of data generated by these technologies. They will enable the integration of data from different sources, such as sensors and IoT devices, into a coherent model that can be used for real-time monitoring and predictive maintenance.

Furthermore, ontologies can enhance sustainability efforts within the ACE industry. By providing a comprehensive framework for analysing the environmental impact of building materials and construction methods, they support the development of greener practices and more sustainable designs. For example, an ontology can help assess the lifecycle impact of different materials, allowing designers to make more informed choices that reduce the carbon footprint of a building.

To sum up, the transformative application of ontology in the Architecture, Construction, and Engineering industry lies in its ability to enhance data interoperability, improve collaboration, support advanced analytics, enable automation, and drive innovation. By leveraging ontologies, the ACE industry can achieve greater efficiency, accuracy, and sustainability, ultimately leading to better-designed, constructed, and managed built environments.

In healthcare, for instance, medical ontologies are used to model diseases, symptoms, treatments, and other medical concepts. These ontologies not only facilitate the organization and retrieval of medical information but also support the development of intelligent systems that can assist in diagnosing diseases and suggesting appropriate treatments (Cubukcuoglu et al., 2021).

In summary, the development of ontology is a multifaceted endeavor that bridges the gap between philosophical inquiry and practical application in fields such as information science, AI, and the Semantic Web. Ontology modeling provides a structured representation of knowledge, which is essential for data integration and enabling AI systems to work with domain-specific information effectively. The application of ontology in the ACE industry is driving innovation and efficiency in various complex domains, promising to revolutionize the way we interact with technology and process information.

## 2.3.1 Ontology Definition

The origin of ontology is rooted in the philosophical domain, where it serves as a systematic exploration of the nature of existence. Gruber's extensively referenced definition posits that ontology is a precise description of a collectively held conceptual framework (Gruber, 1993, 1995). Within the realm of technology, ontological methods provide a structured approach for the formal representation of knowledge by categorizing entities, defining their attributes, and establishing logical interconnections within a defined domain (Wang et al., 2023). This, in turn, facilitates the integration, retrieval, and reutilization of information. There is no broadly defined definition of ontology, and researchers in different fields have different interpretations of it. What follows is a brief overview of several widely recognised ontology definitions:

Grüninger and Fox (1995) defined ontology as a formal method to describe entities and the relationships, attributes, and constraints between them. Uschold and Grüninger (1996) emphasized in their research on ontology development that the lack of a shared understanding often leads to poor communication between individuals and organizations, as well as difficulties in identifying requirements when building IT systems. Building on Gruber's (1993, 1995) definition of ontology, they redefined it as a term that denotes a shared understanding of a given domain, which can serve as a unifying problem-solving framework. Swartout et al. (1997) described ontology as a hierarchical set of terms used to describe a domain, providing a foundational framework for a knowledge base.

Studer, Benjamins, and Fensel (1998) provided an overview of ontology methods in the field of knowledge engineering (KE). They noted that ontology offers a vocabulary of terms and relationships used to model a domain, playing a crucial role in analyzing, modeling, and processing domain knowledge. They expanded on Gruber's (1993, 1995) definition of ontology by clarifying that 'Conceptualization' refers to an abstract model formed by identifying concepts related to a particular phenomenon. 'Explicit' means that the types and constraints of terms used are clearly defined. ‘Formal’ indicates that the ontology should be machine-readable, excluding natural language. ‘Shared’ reflects that an ontology is not private but can be shared and accepted by others.

Stevens, Goble, and Bechhofer (2000) reviewed ontology applications in bioinformatics and molecular biology, illustrating the ontology-building process, techniques, and methods of the time. They introduced the use of ontologies in bioinformatics with specific examples from the domain and interpreted Gruber’s definition as stating that an ontology is a concrete form of the conceptualization of domain knowledge, where the conceptualization involves knowledge about entities and their relationships, and the specification is the expression of this conceptualization in a concrete form.

Noy and McGuinness characterized ontology as a common domain vocabulary that defines domain knowledge or information concepts and clarifies their relationships to facilitate communication among domain experts and between experts and knowledge-based systems. They developed the “7 Steps” ontology-building guide, widely adopted across various domains. The "7 Steps" method, also known as the SKME (Simple Knowledge Engineering Methodology), includes determining the domain and scope, considering reusing existing ontologies, enumerating important terms, defining classes, defining slots, defining the facets of the slots, and creating instances (Zhou, Goh, and Shen, 2016).

Stanford University’s Knowledge Systems Laboratory (KSL) explained ontology as a formal and declarative knowledge representation system, where terms related to the subject domain and the logical relationship statements between these terms are declared. Based on this understanding, Darlington and Culley (2008) considered ontology a useful vocabulary to represent and share knowledge about a specific subject area and its relations.

According to Tserng et al. (2009), ontology is seen as an explicit formal specification of concepts within a specific domain and their relationships. The two vital components of ontology are domain concepts and their relations.

Hitzler, Krötzsch, and Rudolph (2010) described ontology as a description of knowledge about a domain of interest, with its core being a formally defined machine-processable specification. This specification enables the structured representation and sharing of knowledge across various systems and applications.

The Table 2-1 shows a summary from some of the definitions that are accepted in different fields.

Table 2-1 Definitions summary

| Phrase | Definition |
|---|---|
| **Philosophy Domain** | Describes the nature of being and relationships. |
| **Computer & Information Science Domain** | Knowledge management, representation, and sharing. |
| **Knowledge Representation** | Structured knowledge presentation for machines. |
| **Interoperability** | Achieving common understanding for data exchange. |
| **Problem-Solving Framework** | A unified framework for collaborative understanding. |
| **Vocabulary** | Defines terms and relations to model a domain. |
| **Formal Specification** | Machine-readable format for precise representation. |
| **Shared Understanding** | Ontology is shared, not private, for common use. |
| **Ontology Building Process** | Steps to create structured domain knowledge. |
| **Knowledge Clarification** | Making concepts and relationships explicit. |
| **Common Vocabulary** | Shared terms and relations for effective communication. |
| **Interdisciplinary Use** | Application in various domains for knowledge organization. |
| **Machine-Processable** | Knowledge is expressed in a |

| | |
|---|---|
| **Specification** | machine-readable format. |

## 2.3.2 Establishing Ontology

Since the 1990s, there has been a significant growth in the development and application of ontologies, with a variety of methods being established to facilitate this process. These methods have been refined and expanded upon over the years, with researchers and practitioners in various fields contributing to the development of robust and effective ontology construction techniques.

In the academic literature, numerous studies have explored and documented the different methods for building ontologies. Zhipeng Goh and his colleagues (Zhipeng et al., 2016) conducted a comprehensive review of these methods, highlighting the various approaches and tools that have been developed. They emphasized the importance of selecting an appropriate method based on the specific needs and requirements of the ontology being constructed.

The construction methods for ontology can vary significantly depending on the field or domain in question. Different domains have unique requirements and perspectives, which influence the choice of method and the way in which the ontology is structured and populated. For example, Zhang and his colleagues (Zhang, Daxin et al., 2019) explored the construction of ontologies for the field of biology, focusing on the specific needs and challenges of this domain. They proposed a method that combines traditional ontology construction techniques with advanced machine learning algorithms to improve the accuracy and efficiency of the process.

Similarly, Zhang and his colleagues (Zhang, Sijie et al., 2013) investigated the construction of ontologies for the field of computer science, identifying the importance of incorporating domain-specific knowledge and expertise during the construction process. They proposed a method that involves collaboration between domain experts and ontology builders to ensure that the ontology accurately represents the concepts and relationships within the domain.

Noy and McGuinness (2001) presented a method for building ontologies that is based on the use of formal languages and logic. This method ensures that the ontology is rigorous and consistent, making it suitable for applications in fields such as artificial intelligence and the semantic web.

Overall, the construction of ontologies is a complex and multifaceted process that requires careful consideration of the specific needs and requirements of the domain in question. The development of a wide range of methods and techniques has enabled researchers and practitioners to build ontologies that accurately represent the concepts and relationships within their respective domains. These ontologies play a crucial role in knowledge representation, data integration, and enabling machines to understand and work with domain-specific information.

This section briefly reviews several methods of establishing ontology.

#### 2.3.2.1 Uschold and King's Approach

Uschold and King's building ontology methodology is a structured approach to creating ontologies, which are formal representations of knowledge that help in modeling and organizing information in a specific domain (Uschold, Michael & King, 1995). This methodology is widely recognized in the field of ontology engineering.

The methodology typically involves several key steps to ensure a well-structured and functional ontology. First, the scope of the ontology must be determined by defining the domain and specifying the concepts, relationships, and constraints it will encompass. This step establishes clear boundaries for the ontology, ensuring it remains focused and manageable.

Next, knowledge is collected from existing resources, documents, and domain experts to provide a solid foundation for ontology construction. This information helps identify relevant concepts and relationships within the domain. Following this, conceptualization takes place, where key concepts are identified and defined, forming the building blocks of the ontology. These concepts are then organized hierarchically to reflect their relationships.

The next step is specification, where the relationships between concepts are defined. This includes establishing hierarchical ("is-a") relationships, part-whole ("part-of") relationships, and other relevant connections. Once the conceptual model is in place, it is formalized using a structured ontology language such as OWL (Web Ontology Language), making it machine-readable and suitable for computational use.

After formalization, the ontology is implemented by integrating it into a knowledge system or application, where it can be used to organize and reason about data. The ontology then undergoes evaluation and validation to ensure it accurately represents the domain and meets its intended objectives. Comprehensive documentation is also created, detailing the ontology's concepts, relationships, and usage guidelines.

Finally, since ontologies are dynamic, ongoing maintenance is required to update them as the domain evolves or new knowledge becomes available. This ensures the ontology remains relevant and useful over time.

Figure 2.3 briefly shows the steps of this methodology.

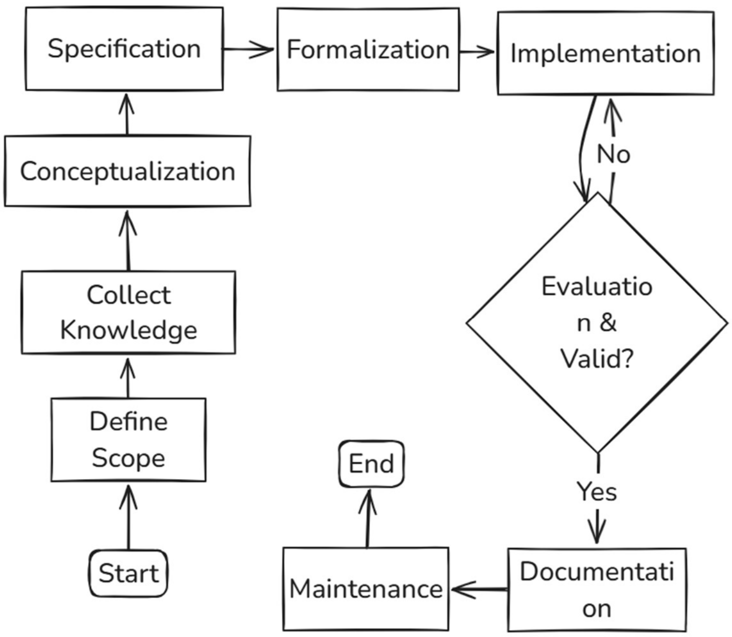



Figure 2.3 Steps of  Uschold and King's methodology

Uschold and King's methodology for building ontologies is a comprehensive and structured framework that offers a clear and systematic approach to the construction of these knowledge models. It provides a roadmap that guides ontology developers through the various stages of ontology creation, ensuring that the resulting ontology is well-defined, robust, and adaptable to a wide range of applications.

One of the key strengths of Uschold and King's methodology is its emphasis on the importance of understanding the domain for which the ontology is being constructed. This involves engaging with domain experts and stakeholders to gain a deep insights into the key concepts, terms, and relationships that are relevant to the domain. This initial analysis helps to ensure that the ontology is aligned with the needs and requirements of the domain, increasing its relevance and usefulness in practical settings.

The methodology also places a strong emphasis on the iterative and incremental development of ontologies. This means that the ontology development process is broken down into smaller, manageable tasks, which are addressed in a iterative manner. This approach allows for the continuous refinement and improvement of the ontology, as feedback and insights from users and domain experts can be incorporated into the ontology at each stage of its development.

Another important aspect of Uschold and King's methodology is its focus on the reuse of existing ontologies and knowledge resources. This is achieved through the use of standardized modelling languages and tools, which facilitate the integration and extension of existing ontologies. By leveraging existing resources, ontology developers can save time and effort, as they do not need to start from scratch when building new ontologies. Instead, they can build upon the work that has already been done, increasing the efficiency and effectiveness of the ontology development process.

Furthermore, Uschold and King's methodology also addresses the issue of scalability, ensuring that the ontologies that are developed can be easily extended and adapted as the domain evolves and new information becomes available(Sun H ,2023). This is achieved through the use of modular and flexible design principles, which allow for the easy addition of new concepts and relationships to the ontology without disrupting the existing structure.

In summary, Uschold and King's methodology for building ontologies provides a robust and adaptable framework that enables the creation of well-defined and reusable knowledge models. These models can be used to support a wide range of applications, including information retrieval, data integration, and knowledge representation in various domains. By following this methodology, ontology developers can ensure that their ontologies are aligned with the needs of the domain, efficient and effective in practical settings, and capable of evolving and adapting as the domain grows and changes.

Uschold and King's methodology for building ontologies is widely recognized and respected in the field of ontology engineering, but it also has several disadvantages and limitations. One major drawback is its complexity, as the methodology can be time-consuming, particularly for large and intricate domains, posing a barrier for those with limited experience. Additionally, it demands a high level of expertise in both the domain and ontology engineering principles, making it difficult for novices to produce high-quality ontologies. The process is also resource-intensive, requiring access to subject matter experts and extensive domain-specific information. Another limitation is its static nature, which makes it less suitable for rapidly evolving domains, as ontologies created using this approach may quickly become outdated and require frequent updates. The methodology also lacks specific guidance on tool selection, leaving newcomers uncertain about which software to use. There is also a risk of over-engineering, resulting in unnecessarily complex ontologies that are difficult to use. Furthermore, ontologies developed with this method may not be immediately interoperable with other systems, requiring additional effort for alignment. Finally, maintaining the ontology over time can be challenging, as ongoing effort is needed to ensure its accuracy and relevance as the domain evolves.

When utilizing Uschold and King's ontology engineering methodology, it is crucial to take into account its potential drawbacks. This introspection ensures that the chosen approach is the most appropriate for the specific ontology development project at hand. Different domains and project specifications might call for alternative ontology engineering methodologies that better cater to their unique needs. It is therefore imperative to carefully evaluate and select the methodology that aligns with the project's goals, domain complexities, and resource constraints. This evaluation process may involve considering factors such as the project's timeframe, budget, the expertise of the development team, and the level of domain expertise required. By doing so, one can optimize the ontology development process and enhance the likelihood of creating a robust, scalable, and domain-relevant ontology.

#### 2.3.2.2 Grüninger and Fox's Approach

Grüninger and Fox's ontology building methodology, renowned for its comprehensive and structured approach, finds extensive application within the Enterprise Integration Laboratory(Uschold, Mike & Gruninger, 1996). This systematic technique is pivotal in the generation of ontologies, which are essentially formalized depictions of knowledge. These ontologies serve the critical function of structuring and categorizing information across a multitude of domains.

In the realm of ontology engineering, Grüninger and Fox's methodology holds significant cachet, offering a dependable framework for the creation of ontological artifacts. The process is typically broken down into several key steps, each designed to facilitate the meticulous development and refinement of ontologies.

The process of developing an ontology involves several key steps to ensure its effectiveness and usability. First, it is essential to determine the purpose of the ontology by clarifying its objectives and intended use cases, as this guides the entire development process. Next, relevant data and domain knowledge must be gathered, including existing resources and expert input, to provide a solid foundation for the ontology. Defining the scope is another critical step, as it establishes the boundaries of the ontology by specifying which concepts, relationships, and constraints will be included. Once the scope is set, the key concepts and their relationships are identified and conceptualized, often using informal models or diagrams for clarity. The next phase involves formalizing the conceptual model into a structured ontology using standardized languages such as OWL or RDF. After formalization, the ontology is implemented in a knowledge system or application and tested to verify that it functions as intended and meets the defined requirements. Continuous evaluation and

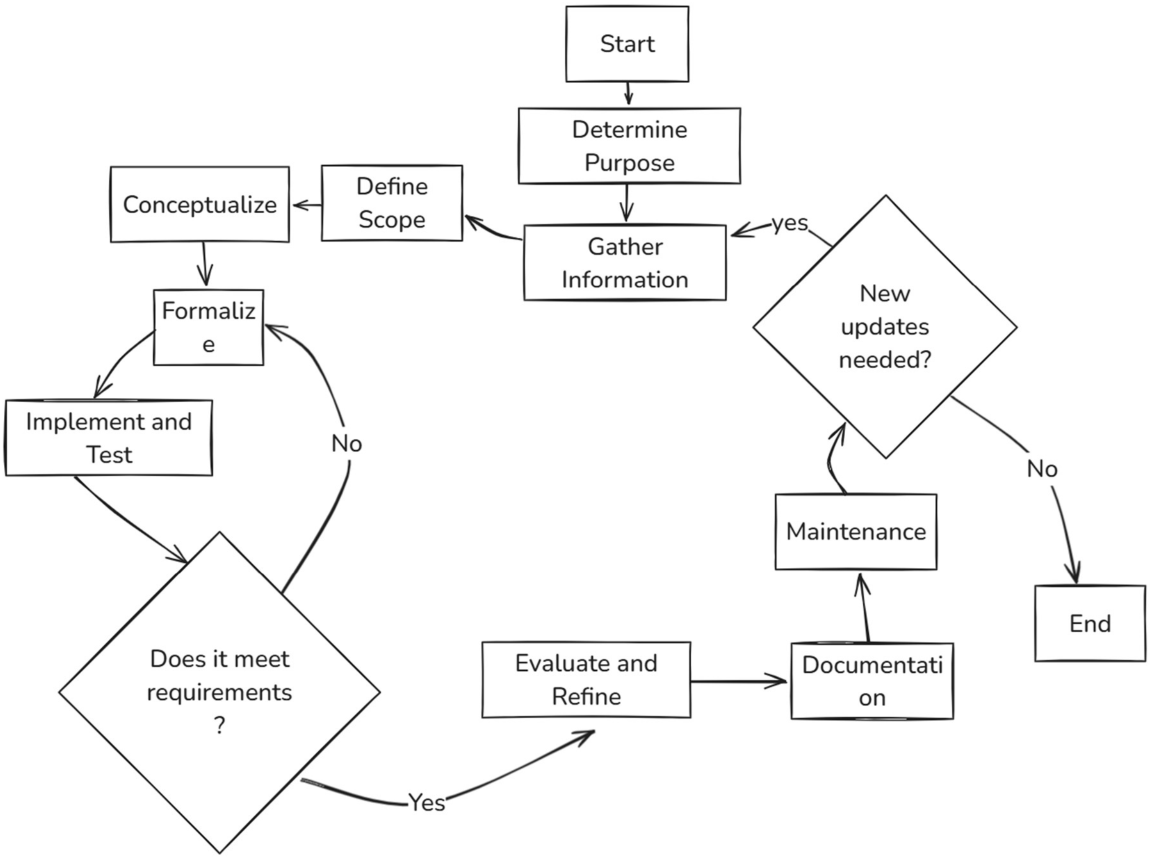



Figure 2.4 The steps of Grüninger and Fox's method

refinement follow, incorporating feedback from domain experts and users to improve accuracy and completeness. Thorough documentation is then created to explain the ontology's structure, concepts, and relationships, ensuring ease of understanding and adoption. Finally, ongoing maintenance is necessary to keep the ontology up to date, adapting it to changes in domain knowledge and evolving requirements over time. Figure 2.4 shows this process steps of Grüninger and Fox's method

Grüninger and Fox's methodology in ontology building is characterized by its emphasis on three key principles: understanding the ontology's purpose, engaging with domain experts, and maintaining a process of continuous refinement. By placing a strong focus on these principles, the methodology ensures that the resulting ontologies are not only robust and theoretically sound but also highly relevant and practical for their intended applications.

At the core of Grüninger and Fox's approach is a deep understanding of the ontology's purpose. This involves a comprehensive analysis of the needs and requirements of the domain in which the ontology will be applied. By clearly defining the goals and objectives of the ontology, the methodology ensures that the resulting artifact is tailored to meet the specific needs of the domain, thereby enhancing its utility and effectiveness.

Another key aspect of Grüninger and Fox's methodology is the active involvement of domain experts. These individuals, who possess deep knowledge and understanding of the domain, play a crucial role in the ontology building process. Their insights and expertise are invaluable in ensuring that the ontology accurately represents the domain's concepts, relationships, and constraints. By collaborating with domain experts, the methodology ensures that the ontology is grounded in the reality of the domain, making it more reliable and relevant for its intended applications.

Grüninger and Fox's methodology also places a strong emphasis on the iterative refinement of the ontology. This involves an ongoing process of evaluating, testing, and improving the ontology to ensure its continued relevance and usefulness. By adopting an iterative approach, the methodology allows for the identification and correction of potential issues and limitations, thereby enhancing the quality and effectiveness of the ontology. This continuous refinement process is essential in maintaining the ontology's alignment with the evolving needs and requirements of the domain, ensuring its long-term usefulness and adaptability.

The structured and iterative nature of Grüninger and Fox's methodology makes it a valuable tool for knowledge representation and data integration in a wide range of applications and domains. By providing a clear and systematic framework for ontology building, the methodology enables the creation of ontologies that are not only theoretically sound but also practical and applicable in real-world contexts. As a result, Grüninger and Fox's methodology has become widely recognized and adopted in the field of ontology engineering, playing a significant role in advancing the development and application of ontologies for knowledge representation and data integration purposes.

Grüninger and Fox's ontology building methodology is a well-established and comprehensive approach to ontology development, but it also has several disadvantages and limitations. One key challenge is its complexity, which can be daunting for novice ontology engineers, as it demands a strong grasp of both the domain being modeled and ontology engineering principles. Additionally, the methodology is resource-intensive, often requiring collaboration with domain experts, extensive data collection, and the use of formal ontology languages, which may involve specialized tools and expertise. The process can also be time-consuming due to its iterative nature, making it less suitable for projects with tight deadlines. Another issue is subjectivity, as the reliance on expert judgment in concept modeling and relationship identification can lead to varying interpretations of the same domain. The methodology also lacks prescribed tools, leaving developers to identify suitable software, which can slow progress. Maintenance presents further difficulties, as keeping the ontology updated with evolving domain knowledge requires ongoing effort. For highly specialized or complex domains, the methodology may need adaptation, adding another layer of complexity. Finally, the steep learning curve associated with the approach can make it less accessible to beginners in ontology engineering.

While Grüninger and Fox's methodology provides a robust and systematic framework for ontology development, it is crucial to thoughtfully evaluate whether this approach is the most suitable for a particular project. By considering the potential disadvantages and limitations of this methodology, stakeholders can make a more informed decision about whether to adopt it or explore alternative approaches that may better align with the project's unique requirements, constraints, and goals.

One potential disadvantage of Grüninger and Fox's methodology is that it can be resource-intensive, requiring significant time, expertise, and financial investment. The iterative and collaborative nature of the approach often necessitates ongoing engagement with domain experts, which can be challenging to sustain, especially in projects with limited resources. Additionally, the process of requirements analysis and domain modelling can be complex and time-consuming, potentially delaying the overall ontology development timeline.

Another consideration is that Grüninger and Fox's methodology may prioritize theoretical rigor over practicality in some cases. The emphasis on a deep understanding of the ontology's purpose and the involvement of domain experts can sometimes lead to ontologies that are overly complex or difficult to implement in real-world applications. It is essential to strike a balance between a robust theoretical foundation and the practical needs of the project to ensure that the resulting ontology can be effectively integrated into the intended system or platform.

Furthermore, the iterative refinement process inherent in Grüninger and Fox's methodology may not always be well-suited to projects with fixed deadlines or rapidly evolving domains. The continuous evaluation, testing, and improvement of the ontology can be challenging to manage in such contexts, potentially leading to delays or difficulties in meeting project milestones.

Lastly, the reliance on domain experts in Grüninger and Fox's methodology can introduce biases or limitations in the ontology development process. The perspectives and knowledge of domain experts are invaluable, but they may not always encompass the full range of perspectives or needs within a domain. It is important to critically evaluate the inputs provided by domain experts and consider alternative viewpoints to ensure the ontology's broad applicability and relevance.

In conclusion, while Grüninger and Fox's methodology offers a structured and iterative approach to ontology development, it is essential to carefully consider its potential disadvantages in the context of a specific project. By doing so, stakeholders can determine whether this methodology aligns with the project's resources, timeline, and goals or if an alternative approach would be more appropriate.

#### 2.3.2.3 "METHONTOLOGY" Approach

Fernández-López (1997b) outlined a comprehensive set of activities that constitute the ontology development process, introducing a life cycle approach for building ontologies through evolving prototypes. He also developed METHONTOLOGY, a well-structured methodology designed for constructing ontologies from the ground up. This methodology was formulated based on his extensive experience in creating an ontology within the domain of chemicals. METHONTOLOGY provides a systematic framework for ontology development, encompassing stages such as specification, conceptualization, formalization, implementation, and maintenance, ensuring a thorough and iterative approach to building robust and scalable ontologies. The ontology development process follows a structured sequence of key steps. It begins with knowledge acquisition, where domain experts and existing sources provide the necessary information and data. Next, the specification phase defines the ontology's scope, purpose, and the concepts and relationships it will cover. This is followed by conceptualization, where the main domain concepts are identified and modeled, often using informal diagrams or frameworks. The formalization stage then converts this conceptual model into a structured ontology using formal languages like OWL. Once formalized, the implementation phase integrates the ontology into knowledge-based systems or applications for practical use. Finally, maintenance ensures the ontology remains up to date through continuous refinement and adaptation as the domain evolves. This iterative process helps maintain accuracy and relevance over time. Figuer 2.5

illustrates process above.

METHONTOLOGY encourages the use of evolving prototypes to iteratively improve the ontology, and it emphasizes the involvement of domain experts and thorough documentation. This methodology aims to create ontologies that are well-structured, maintainable, and adaptable to changes in the domain over time. Thorough documentation is vital to ensure the ontology's long-term accuracy and usefulness.

#### 2.3.2.4 SKEM Approach

Noy and McGuinness's Simple-Knowledge Engineering Methodology (SKEM) is a streamlined approach to ontology development that emphasizes practicality and simplicity (Noy, Natalya F. & McGuinness, 2001). Unlike more complex methodologies, SKEM is designed to be accessible to a wide range of users, including those without extensive expertise in ontology engineering.

The motivation behind SKEM is reflected in five key aspects. Firstly, it aims to establish a shared understanding of information structure within a specific domain, enabling seamless communication and information exchange among different

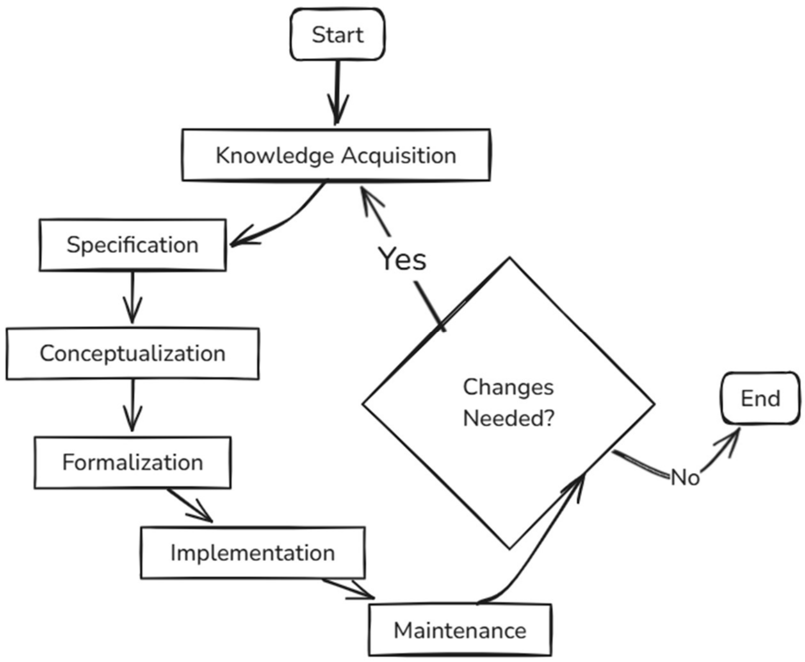



Figure 2.5 Steps of "METHONTOLOGY" Approach

stakeholders, including people and software agents. Secondly, SKEM facilitates the reuse of domain knowledge by explicitly capturing and representing it, allowing researchers and developers to leverage existing ontologies rather than starting from scratch. Thirdly, it promotes transparency by making domain assumptions explicit, documenting them within the ontology instead of embedding them in code, which simplifies adaptation as the domain evolves. Fourthly, SKEM supports the separation of domain knowledge from operational knowledge, ensuring that domain-specific information is clearly defined and can be modified independently of system operations. Lastly, it enables formal analysis of domain terms and relationships, enhancing the ability to reuse and extend existing ontologies for further research and development.

The motivations for building ontologies revolve around creating a shared understanding of domain information, promoting knowledge reuse, making assumptions explicit, separating domain and operational knowledge, and enabling in-depth analysis of domain knowledge. These motivations are essential for various applications, including data integration, knowledge management, and information retrieval.

SKEM has 7 steps, Noy and McGuinness (2001) practiced particularly in the context of wine and food. It emphasizes the importance of defining the domain and scope of the ontology at the beginning. The document discusses the creation of competency questions to determine the ontology's scope and the potential for reusing existing ontologies. The steps in ontology development show in Table 2-2:

Table 2-2 “7-Step” Methodology

| Steps | Content |
|---|---|
| **Determine the domain and scope of the ontology.** | ▪ Define the ontology's domain and purpose.<br>▪ Consider potential users and their needs. |
| **Consider reusing existing ontologies.** | ▪ Explore the possibility of using pre-existing ontologies.<br>▪ Mention resources for finding reusable ontologies. |

| | |
|---|---|
| **Enumerate important terms in the ontology.** | ▪ Create a list of terms that need representation.<br>▪ This list includes terms related to wine, food, and other relevant concepts. |
| **Define the classes and the class hierarchy.** | ▪ Discuss different approaches to creating a class hierarchy, such as top-down, bottom-up, and a combination of both. |
| **Define the properties of classes—slots.** | ▪ Discuss the various facets of slots, including slot cardinality, slot-value type, and slot domain and range. |
| **Define the facets of the slots.** | ▪ Describe the different value types, such as strings, numbers, booleans, and enumerated values.<br>▪ Discuss relationships between classes and how to define slots. |
| **Create instances.** | ▪ Develop individual instances of classes by filling in slot values.<br>▪ Provides an example of an instance for a specific type of wine. |

Noy and McGuinness's guide also touches on common pitfalls and considerations when defining class hierarchies, such as avoiding redundant subclassing (e.g., singular and plural versions of the same concept) and ensuring that the hierarchy follows an "is-a" relation, where a subclass represents a concept that is a "kind of" the superclass.

It provides a comprehensive overview of the steps involved in developing an ontology and offers guidance on best practices and potential issues to avoid.

SKEM's strength lies in its simplicity and practicality, making it a useful methodology for a wide range of applications and for individuals who are new to ontology development. It encourages the rapid creation of ontologies that are easy to understand and maintain.

## 2.4 IFC TO OWL

The widespread use of the BIM paradigm in the AEC/FM business has yielded undeniable advantages over earlier methods: cost savings during design and construction, as well as other advantages, e.g. interoperability between stakeholders is improved, and the building's life cycle is better managed (planning, construction, maintenance, demolition) (Pauwels et al., 2017c). Furthermore, it increases the likelihood that relevant actors would exchange the building's "semantic" information, implying a collective development in knowledge. In this scenario, it is required to specify a formal representation in the form of an open standard, the. Based on this, Zhong et al. (Zhong et al., 2019) conducted a scientometric analysis and critical review of architecture-related ontology research, examining 199 references published between 2007 and 2017. Their analysis revealed a shift in keyword trends from ‘project management’ and ‘knowledge management’ to ‘building information modeling’ (BIM) and ‘compliance control,’ highlighting the increasing prominence of BIM in the AEC field. Additionally, they identified three key future challenges: ontology-based information extraction, semantic enrichment of IFC schemas, and the automatic or semi-automatic generation of domain ontologies from documents.

To promote BIM in terms of data reuse and compatibility across heterogeneous applications, integrating it with the concept of the Semantic Web (SW) can be highly beneficial. Translating the IFC schema into OWL (Ontology Web Language) effectively generates an ontology from the IFC schema, offering significant advantages (Pauwels et al., 2015a):

- Communicating with external data sources.
- Reusing existing ontologies in AEC and other domains.
- Promoting the use of software tools, particularly those related to search, such as SPARQL.
- Utilizing inference techniques developed for the Semantic Web (Pauwels et al., 2015b);
- Providing a logical form that the EXPRESS languages (commonly used

for IFC schemes) lack.

- Leveraging a broader user base and more extensive use, leading to greater potential for technological development in this area.
- Discovering and utilizing resources available in the Semantic Web.
- Facilitating more intuitive operation of building information(De Farias et al., 2015).
- Supporting stakeholders in making faster decisions with more detailed global information.

Despite the inherently complex and condensed structure of IFC, these potential advantages have motivated some researchers to develop various functional IFC-OWL transformations. One notable implementation of this effort is ifcOWL. (Beetz et al., 2009a; Pauwels et al., 2015a; Pauwels & Terkaj, 2016; Pauwels & Roxin, 2017; Pauwels et al., 2017b; Terkaj & Šojić, 2015; Terkaj & Pauwels, 2017). The ontology preserves the robust structure of the original EXPRESS version (Pauwels et al., 2017a) though some expressiveness is lost during the conversion process (Bonsma et al., 2016). This loss is attributed to challenges such as converting the list data type. Furthermore, due to its large size, ifcOWL presents significant challenges when used in practical applications that require querying or reasoning (Terkaj & Pauwels, 2017)However, the benefits obtained from the transformation outweigh these drawbacks, making ifcOWL a reliable version of the IFC schema. Additionally, this ontology offers features that promote its adoption, including growing usage and support from expert organizations like the buildingSMART Linked Data Working Group. These advantages, coupled with the robust structure maintained from the original EXPRESS version, position ifcOWL as a valuable tool in the field of building information modeling, despite the inherent challenges posed by its size.

## 2.5 Ontology-related work in ACE Domain

Many industries have developed ontologies for efficient knowledge management, such as medicine, computer science, and biology. In the construction industry, ontology has also been widely introduced and studied because construction projects involve collaboration among various professionals (e.g., architectural design, plumbing, heating, ventilation, and air conditioning), stakeholders (e.g., designers, contractors, owners), and phases (e.g., design, construction, operation), which may lead to 'information silos.' The use of ontology enhances information sharing and reuse of structured data, thus improving the ability for different computer systems to interoperate effectively without misinterpreting or losing data.

Zhong's team conducted a comprehensive review of ontology research published in the Scopus database from 2007 to 2017 (Zhong et al., 2019). This review comprised a total of 199 articles and employed a combination of scientific measurement analysis and critical commentary. Scientific measurement analysis, such as examining common authors, shared keywords, co-citations, and clustering, was used to objectively visualize the current state of research. Additionally, critical commentary was employed to identify the research themes and challenges in the field of ontology studies in the construction industry.

Zhong’s research outlined four prominent research themes were identified chronologically: "Domain ontology," "Industry foundation classes," "Automated compliance checking," and "Building information modelling." Each theme was associated with specific challenges, such as automated ontology generation from documents, semantic enrichment of Industry Foundation Classes (IFC) files, and machine learning-based information extraction for automated compliance checking.

The discussion highlighted several key points regarding domain ontology and its applications in the construction industry. Domain ontology serves as a structured representation of knowledge within a specific field, encompassing concepts, relationships, and properties to facilitate shared understanding. In construction, researchers have developed various domain ontologies to improve knowledge management, though creating a single comprehensive ontology for an entire domain is often impractical. Instead, the focus has shifted toward developing structured, extendable ontologies that capture essential concepts, including function-specific ones for areas like highway construction and building environmental monitoring.

Developing domain ontologies in construction presents challenges, particularly due to differing stakeholder perspectives (e.g., occupants, designers, contractors), necessitating customized approaches. Most existing ontologies have been manually developed, which is labor-intensive, underscoring the need for more research into automated or semi-automated ontology generation from structured and unstructured data sources.

Ontology has been widely applied in BIM (Building Information Modeling) to enhance construction management and IFC (Industry Foundation Classes) interoperability. It supports risk knowledge management, defect management, and automated cost estimation by linking relevant information to building objects in a reusable manner. IFC, as a key BIM data schema, improves interoperability between tools, though research continues to enhance its semantic clarity for better data exchange. Semantic enrichment—such as classifying building objects, ensuring unique identification, and enabling aggregation—is critical for optimizing IFC's effectiveness. Integrating semantic web technologies with IFC can further improve extensibility and knowledge inference.

Automated compliance checking, another key application, addresses the inefficiencies of manual processes in construction. Ontology and semantic web technologies enable rule-based regulatory modeling, information extraction, and automated compliance verification. Ontological reasoning enhances the accuracy and completeness of rule retrieval, while ontology-based information extraction (OBIE) methods—whether rule-based or machine learning-based—improve performance in processing regulatory texts. Deep learning techniques can further refine machine learning approaches. Compliance checking implementations often rely on rule languages like SWRL, N3Logic, and RIF to formalize reasoning processes.

This study provides valuable insights into the development and trends in ontology research within the construction industry, aiding both researchers and practitioners in understanding the current landscape and potential future directions in this field.

## 2.6 Ontology-based layout planning

Section 2.1 provides an overview of the present state and difficulties encountered in the process of creating layout designs for buildings. In the context of modern BIM software, a significant challenge exists in the conversion of a model that is structured around individual rooms to one that is organized by zones, and this conversion currently necessitates manual intervention. This manual process can be time-consuming and prone to errors, especially as the complexity of the building design increases.

Advancements in science and technology have significantly propelled the field of architectural layout design, leading to increased integration with various other disciplines. This integration has allowed for the development of more sophisticated and efficient design methodologies, tools, and algorithms. For instance, computer-aided design (CAD) software has become an indispensable tool for architects, enabling them to create detailed and precise designs with greater ease . Additionally, BIM has revolutionized the design and construction industry by providing a digital representation of the physical and functional characteristics of a building project.

Within the realm of layout design, numerous methods have been proposed to automate and streamline the design process. These methods range from rule-based systems (Sydora & Stroulia, 2020) to genetic algorithms (Chen, Chen et al., 2021) and machine learning techniques (Rahbar et al., 2022b). However, despite these advancements, there is a notable lack of integration between layout design methodologies and ontology. Ontology, in the context of architecture, refers to the creation of a formal representation of knowledge about a building's components, their relationships, and their attributes. This representation can greatly enhance the semantic richness of a BIM model, allowing for more intelligent and meaningful interactions with the design data.

The incorporation of ontology into layout design methods has the potential to greatly enhance the automation and intelligence of the design(Zampetakis et al., 2012a). By leveraging the structured knowledge captured in an ontology, design software could potentially make more informed decisions during the layout generation process, leading to more efficient and effective designs. However, this integration is still in its infancy, and there is a need for further research and development to explore the potential benefits and challenges of combining ontology with layout design methodologies. This research could pave the way for a new generation of BIM software that is not only more automated but also more intelligent, providing architects with powerful new tools to aid in the creation of innovative and efficient building designs.

Luca's research, as documented by Caneparo (2022a), delves into the intricacies of knowledge management and its diverse applications within the fields of urban planning and layout generation. The study highlights the utility of formal ontologies as robust tools for encapsulating and organizing knowledge across a spectrum of disciplines, including but not limited to urban planning, construction, and renewable energies. These ontologies serve as conceptual frameworks that facilitate the standardization and sharing of information, thereby enhancing the coherence and accessibility of domain-specific knowledge.

Within the scope of Luca's research, particular emphasis is placed on the development and implementation of two distinct ontologies: the Urban Morphology Ontology (UMO) and the Semantic Tools for Carbon Reduction in Urban Planning (SEMANCO) ontology. The UMO is designed to address the complexities of urban morphology, providing a structured representation of the physical form and structure of urban areas. This ontology enables the capture and analysis of various urban characteristics, such as building configurations, street layouts, and land use patterns, which are crucial for informed urban planning and design decisions.

Concurrently, the SEMANCO ontology focuses on the integration of semantic knowledge with the aim of reducing carbon emissions within urban planning. By leveraging the principles of semantic web technologies, this ontology facilitates the representation and manipulation of knowledge related to carbon reduction strategies, such as the deployment of renewable energy sources, energy-efficient building designs, and sustainable transportation systems. The SEMANCO ontology, thus, plays a pivotal role in promoting the adoption of environmentally conscious practices within the urban planning domain.

Together, these ontologies exemplify the potential of knowledge management systems to structure and streamline the processes involved in urban planning and layout generation. They serve as powerful resources for professionals in the field, offering enhanced capabilities for knowledge organization, retrieval, and application. The research by Luca, through the exploration of these ontologies, underscores the transformative impact that knowledge management can have on urban planning practices, ultimately contributing to the development of more sustainable and efficient cities.

In his research, Luca goes beyond the realm of knowledge management and delves into the intricacies of layout generation, exploring various methodologies and approaches that are employed for this purpose. The process of generating layouts is a complex task that requires careful consideration of numerous factors, including functional requirements, aesthetic preferences, and contextual constraints. Luca's study provides a comprehensive overview of the different techniques and methodologies that have been proposed and utilized in the field of urban planning and design.

One of the methodologies discussed is the use of production rule systems. These systems involve the application of a set of predefined rules to generate layouts that meet specific criteria or objectives. Production rule systems are particularly useful in urban planning and design as they allow for the representation of complex relationships and constraints, enabling the generation of layouts that are both functional and aesthetically pleasing.

Another approach explored is the use of fractals. Fractals are mathematical patterns that exhibit self-similarity at various scales, and they have been widely used in urban planning and design to create layouts that are both efficient and aesthetically appealing. The use of fractals in layout generation can lead to the creation of intricate and complex urban environments that possess a sense of order and harmony.

Cellular automata is another methodology discussed in Luca's research. Cellular automata are mathematical models that consist of a grid of cells, each of which can be in one of several states. The behaviour of the cells is determined by a set of rules that are applied iteratively to generate complex patterns and structures. In the context of urban planning and design, cellular automata can be used to simulate the growth and development of urban areas, allowing for the generation of layouts that are both functional and contextually appropriate.

Lastly, Luca's research discusses declarative approaches for generating layouts. Declarative approaches involve the specification of the desired properties or characteristics of the layout, and the generation process is left to be determined by the underlying system. This approach allows for a high degree of flexibility and adaptability in the layout generation process, as the system can explore different possible solutions and select the one that best meets the specified criteria.

At last, Luca's research provides a comprehensive exploration of the different methodologies and techniques that are used for layout generation in urban planning and design. By discussing these approaches, Luca offers valuable insights into the complexities of the layout generation process and highlights the potential benefits and challenges of each methodology. This research can serve as a valuable resource for professionals in the field, enabling them to select and utilize the most appropriate approach for their specific project requirements.

Overall, Luca's work highlights the importance of formal ontologies for managing and sharing knowledge in complex domains like urban planning and how this knowledge can be used in the generation of layouts to aid decision-making and design processes.

Stamatis and his team have introduced a challenge and solution related to the visualization of large-scale ontologies, with a specific focus on RDF ontologies . They emphasize two main aspects as follows:

Stamatis and his team (Zampetakis et al., 2012b) have identified key challenges and proposed solutions for visualizing large-scale ontologies, particularly RDF-based ontologies. They highlight two major difficulties in working with such ontologies. First, comprehending large ontologies—especially those with numerous classes—can be highly time-consuming, particularly without prior expertise or guidance. Second, traditional visualization methods, such as ER diagrams, UML models, or standard RDF graphs, become ineffective as ontologies grow in size and complexity. To address these issues, the researchers outline essential requirements for effective visualization, including semi-automatic and interactive layout tools that allow users to adjust automated algorithms, as well as filtering techniques to reduce complexity and customize diagrams.

The study focuses on four key areas to improve ontology visualization. First, it introduces real-time exploration capabilities using star-shaped graphs with adjustable radii, enabling users to control the amount of information displayed based on their needs or screen constraints. Second, it examines the configuration of force-directed placement algorithms used in rendering these star-shaped graphs, proposing user-adjustable controls and an automatic layout optimization method to adapt to different ontology structures. Third, the research includes a quality assessment of the proposed visualization techniques, using various metrics to evaluate layout effectiveness. The results indicate that automated configuration methods improve layout quality, and a user study confirms the benefits of these enhancements. Finally, the study briefly addresses the visualization of multiple ontologies, recognizing that ontologies often incorporate elements from other ontologies. The proposed solution supports multiple visualization options for dependent ontologies, improving overall readability and usability.

In his research, Stamatis concentrates on overcoming the difficulties associated with visualizing extensive RDF ontologies. He introduces innovative methods designed to improve the quality and user-friendliness of ontology diagrams, which are essential for real-time exploration and comprehension. The research likely culminates in the development and evaluation of these methods within a dedicated system referred to as "StarLion." This system is likely designed to facilitate efficient visualization and interaction with RDF ontologies, enabling users to navigate and understand complex datasets more effectively.

Elsewhere, Georg Suter, along with co-authors Petrushevski and Šipetić (Suter et al., 2014), and in his 2022 work, delves into the modelling of multiple perspectives of spatial configurations within the realm of schematic building design. The focus is on leveraging space ontologies and implementing layout transformation operations to refine the conversion of room-centric building data into multi-view space models. This conversion is critical in the context of BIM systems, which are commonly used for authoring and managing architectural designs.

The transformation process from room-based to multi-view space models is particularly significant as it allows for a more nuanced understanding and representation of spatial relationships and functions. By utilizing space ontologies, which provide a structured framework for classifying and defining spatial entities and their attributes, the research aims to enhance the semantic richness of BIM data. This, in turn, supports more sophisticated queries, analysis, and simulation within the design process.

The layout transformation operations explored by Suter and his colleagues likely include methods for abstracting and representing spatial configurations in a way that captures the essence of the designed space while accommodating various perspectives. These operations may involve the application of computational algorithms that respect both semantic and spatial criteria, ensuring that the resulting multi-view space models are not only conceptually sound but also functionally viable.

The work of Suter and his collaborators underscores the importance of integrating ontological principles into the design and modelling process, particularly within the context of BIM. By doing so, they contribute to the development of more intelligent and flexible design tools that can support architects and designers in creating more meaningful and efficient spatial configurations.

The article examines several key challenges and findings related to space modeling and classification in architectural and building design. A major challenge discussed is the difficulty in transforming space data across different domains while accounting for both semantic meaning and spatial relationships. For instance, architectural design often requires grouping spaces with similar functions into zones to manage activity separation, noise control, or accessibility—a process that demands careful consideration of multiple criteria.

Existing methods for model transformation face notable limitations. Current zoning approaches in thermal domains and manual zone definitions in BIM (Building Information Modeling) authoring tools lack automation, highlighting the need for more advanced space classification systems. The article critiques conventional classification systems, such as OmniClass and Uniclass, which categorize spaces based on form or function but are often too rigid. A more flexible and extensible classification framework is needed to better support dynamic design requirements.

Semantic enrichment, a process that derives additional insights from existing building models, is explored as a potential solution. The article discusses how ontology-based models and rule languages can simplify data queries, connect information across domains, and facilitate compliance with standards and regulations. Semantic reasoning and query engines further enhance these models by enabling the reuse of existing ontologies to develop new ones, with semantic rules encoded in various ways for greater adaptability.

However, ontology-based methods have their own constraints, particularly in spatial reasoning and geometric processing—both essential for accurate building modeling. To address this gap, the article notes the development of domain-specific rule languages that better integrate spatial logic, improving the precision and applicability of semantic enrichment in architectural and construction contexts.

In summary, the research presented by Georg Suter, Petrushevski, and Šipetić (2014) and Suter (2022) introduces a novel method for converting room-based building data, typically generated in BIM authoring systems, into multi-view space models. This conversion is achieved by utilizing space ontologies and layout transformation operations. The method aims to overcome the limitations and challenges associated with existing transformation techniques, which often fail to fully capture the complexities of spatial relationships and functions within architectural designs.

The research emphasizes the importance of flexible classification systems and semantic enrichment in the context of building modeling. By incorporating space ontologies, the method enables the creation of multi-view space models that are not only conceptually sound but also functionally viable. These models provide a more nuanced representation of spatial configurations, allowing for more sophisticated queries, analysis, and simulation within the design process.

The study highlights the significance of improving the efficiency and accuracy of model transformation in the field of schematic building design. The proposed method offers a promising solution to this challenge, as it leverages the power of space ontologies and layout transformation operations to refine the conversion of room-based building data. By doing so, the research contributes to the development of more intelligent and flexible design tools that can support architects and designers in creating more meaningful and efficient spatial configurations.

Overall, the work of Suter and his collaborators underscores the potential of integrating ontological principles into the design and modelling process, particularly within the context of BIM. By addressing the limitations of existing methods and emphasizing the importance of flexible classification systems and semantic enrichment, their research paves the way for more advanced and effective approaches to building modelling in the future.

## 2.7 Summery for Literature review

The literature review suggests that ontology can significantly facilitate layout design in several ways. Ontologies provide a structured framework for representing knowledge about building components, their attributes, and relationships. This structured representation enhances data integration and interoperability across different systems and stakeholders in the Architecture, Construction, and Engineering

(ACE) industry. By converting data formats like IFC to OWL, ontologies enhance the semantic capabilities of BIM models, making it easier to share and utilize building information across various platforms.

Additionally, incorporating ontologies into layout design methodologies can automate and enhance the intelligence of the design process. Leveraging the structured knowledge captured in an ontology allows design software to make more informed decisions during the layout generation process, leading to more efficient and effective designs. Ontologies also facilitate the automated checking of regulatory compliance and safety standards. By encoding regulations and standards within an ontology, it becomes possible to automatically verify that a design adheres to required guidelines, reducing the risk of non-compliance and enhancing overall safety.

Ontologies support flexible and adaptable classification of spaces and functions within a building, allowing for more dynamic and responsive design processes where layouts can be easily adjusted to meet evolving requirements and constraints. They also improve the management and reuse of knowledge within the ACE industry by providing a common vocabulary and conceptual framework, enabling different stakeholders to share and reuse design knowledge more effectively, reducing redundancy, and improving overall efficiency.

Furthermore, ontologies enable more sophisticated queries and analyses of design data, supporting better decision-making. For example, they facilitate energy modeling and simulation, helping designers optimize layouts for energy efficiency and sustainability. By enhancing information sharing and reuse, ontologies reduce the prevalence of information silos within construction projects, leading to more cohesive and integrated workflows as all stakeholders can access and use the same set of structured information.

However, combining ontology and layout design presents several gaps and challenges that need to be addressed to fully leverage the potential benefits. One of the primary issues is the complexity of ontology development. Creating accurate and comprehensive ontologies that capture all necessary aspects of layout design requires a deep understanding of both the domain and ontology engineering principles. This process can be time-consuming and resource-intensive, posing a barrier to entry, particularly for firms without extensive experience in ontology development.

Integration with existing systems is another significant challenge. Many current BIM and CAD systems may not be designed to handle the semantic richness that ontologies provide, necessitating significant modifications or the development of new tools and platforms. Additionally, scalability issues arise as ontologies, particularly large and complex ones, can become unwieldy and difficult to manage. Ensuring that ontologies remain scalable and performant as the size and complexity of building projects increase is a critical concern.

While ontologies aim to improve interoperability, ensuring that different ontologies and systems can seamlessly interact is not straightforward. Differences in ontology structures, terminologies, and underlying technologies can hinder effective integration and data exchange. Furthermore, the dynamic nature of the building design and construction industry means that ontologies must be continuously updated to reflect new standards, regulations, and technologies. Keeping ontologies current and relevant requires ongoing effort and coordination among various stakeholders.

User acceptance and training also pose challenges. Encouraging the adoption of ontology-based approaches in layout design requires overcoming resistance to change. Users need to be convinced of the benefits, and adequate training must be provided to ensure that they can effectively use ontology-based tools and methodologies. Moreover, while ontologies enhance semantic reasoning, there are limitations in current technologies' ability to perform complex spatial and geometric reasoning required for layout design. Developing more advanced reasoning capabilities that can handle these complexities is essential.

Ensuring the quality and consistency of data within ontologies is crucial for their effectiveness. Inaccurate or inconsistent data can lead to incorrect design decisions and undermine the benefits of using ontologies. Additionally, the initial costs associated with developing and implementing ontology-based systems can be high. This includes the costs of software development, training, and ongoing maintenance, which can be a barrier for smaller firms or projects with limited budgets.

Successful ontology development and implementation require close collaboration among domain experts, IT professionals, and end-users. Coordinating this interdisciplinary effort can be complex and challenging. Addressing these gaps and challenges is essential for effectively combining ontology and layout design to realize their full potential in creating intelligent, efficient, and effective building layouts.

In conclusion, integrating ontology with layout design methodologies can lead to more intelligent, efficient, and effective design processes. Ontologies facilitate better data integration, automation, compliance, flexibility, knowledge management, and decision-making, ultimately contributing to the creation of high-quality building layouts.

# Chapter 3　Review of Epidemic Resilient Cabin Hospitals

Chapter 2 provides a comprehensive review of existing literature on layout design methodologies, Building Information Modeling (BIM), and ontology applications in the Architecture, Construction, and Engineering (ACE) domain. It highlights the limitations of traditional design approaches, such as inefficiency in manual processes, challenges in cost and time control, and fragmentation of knowledge due to multi-party collaboration. The chapter emphasizes the transformative potential of ontology-based frameworks to address these issues by enabling structured knowledge representation, interoperability, and automated reasoning. Key themes include the integration of BIM with semantic web technologies (e.g., IFC-to-OWL conversion) and the role of ontologies in optimizing spatial relationships and constraints for resilient healthcare facilities.

Chapter 3 shifts focus to real-world implementations of epidemic-resilient cabin hospitals, analyzing case studies like Xiaotangshan Hospital (Beijing), Huoshenshan/Leishenshan Hospitals (Wuhan), and Nightingale Hospital (London). These examples illustrate the urgent need for rapid, flexible, and cost-effective design solutions during public health crises. The case studies reveal common challenges: the complexity of coordinating multi-disciplinary teams, the demand for modular and scalable layouts, and the integration of medical workflows with spatial planning. Notably, Huoshenshan Hospital's modular design and Nightingale Hospital's repurposing of existing infrastructure demonstrate how innovative approaches can mitigate these challenges. At the same time, this chapter, as the knowledge about Cabin Hospital, will become a knowledge background and provide reference in the process of ontology establishment.

The transition from Chapter 2 to Chapter 3 bridges theoretical foundations with practical validation. The literature review identifies gaps—such as the lack of ontology-driven automation in layout design—while the case studies underscore the real-world consequences of these gaps (e.g., reliance on labor-intensive manual design during emergencies). The failures and successes of the reviewed hospitals (e.g., Huoshenshan's efficient modularity vs. Nightingale's underutilization due to inflexible zoning) directly align with the ontology-based solutions proposed in Chapter 2.

In the interoperability and knowledge integration aspect of Chapter 2, the heterogeneity of data in multi-stakeholder projects (Problem 3) mirrors the fragmented coordination observed in Wuhan's rapid construction (Chapter 3). An ontology-based BIM framework could streamline this by unifying spatial, functional, and medical requirements into a shared digital model.

For reusability and modularity aspect in Chapter 2, the modular units of Huoshenshan Hospital (Chapter 3) align with ontology's ability to standardize and reuse design components, reducing trial-and-error in future projects.

The understanding of reveal ad-hoc spatial arrangements (e.g., Nightingale's open wards risking cross-contamination) that could be systematically optimized using CSP algorithms (Chapter 5) guided by ontology-defined rules (e.g., the size of the ward unit, pathways, spatial physical location relationship, etc.).

By synthesizing these chapters, the thesis positions ontology as a critical tool to address the inefficiencies and adaptability demands exposed by the pandemic. The subsequent chapters (e.g., Methodology, Ontology Development) will operationalize this linkage, translating theoretical insights from Chapter 2 and empirical lessons from Chapter 3 into a cohesive ontology-based design approach for cabin hospitals.

From the outset of the COVID-19 pandemic in 2019 through to 2021, the world has faced an unprecedented public health crisis. The virus, which spreads with remarkable speed, has posed a significant threat to populations around the globe. The illness it causes can range from mild to severe, and in the most severe cases, patients can become critically ill, necessitating intensive care unit (ICU) monitoring and treatment. The demand for ICU beds and medical resources has skyrocketed, putting immense pressure on healthcare systems.

In response to the overwhelming surge in patients, countries worldwide have had to quickly adapt and find innovative solutions to ensure that those who needed critical care could receive it. One of the most visible responses has been the construction of modular or converted epidemiological hospitals(Zhou et al., 2020). These facilities are designed to provide additional beds and treatment spaces specifically for COVID-19 patients, helping to alleviate the burden on existing healthcare facilities that were often ill-prepared for the scale of the crisis. Zhou’s work also indicate that modular hospitals are prefabricated structures that can be quickly assembled onsite, offering a flexible and scalable solution to increase healthcare capacity. These units can be designed to include patient rooms, isolation areas, and even ICU units, all of which can be rapidly deployed in areas experiencing a high volume of COVID-19 cases.

Converted epidemiological hospitals, on the other hand, involve repurposing existing buildings such as convention centres, sports facilities, or other large structures into temporary healthcare facilities(Lawrence Dunhill, 2020). These buildings are often retrofitted with necessary medical equipment and staffed with healthcare professionals to treat patients with less severe symptoms while still providing a level of isolation that helps contain the spread of the virus.

Countries like China, Italy, Spain, and the United States, among others, have turned to both modular and converted hospitals to manage the crisis(Kraus et al., 2025). These temporary facilities have played a crucial role in containing the spread of the virus by providing dedicated spaces for treating infected individuals and isolating them from the general population.

The construction and operation of these facilities have not been without challenges, including the need for coordination with existing healthcare services, the training of staff for new types of facilities, and the logistical complexities of providing adequate medical supplies and equipment. Nonetheless, these temporary hospitals have been a vital component in the fight against COVID-19, saving countless lives and giving healthcare systems the breathing room they needed to manage the crisis more effectively.

## 3.1 Xiaotangshan Hospital in Beijing

During the outbreak of Severe Acute Respiratory Syndrome (SARS) in Beijing, China, in 2003, Xiaotangshan Hospital emerged as a shining example of rapid response and adaptability in the face of a pressing healthcare crisis (Luo, Z. et al., 2021). Amidst a severe shortage of hospital beds and medical facilities capable of treating the increasing number of SARS patients, the decision was made to transform the existing Xiaotangshan Sanatorium into a tertiary-level general hospital dedicated to combatting the disease.

Located in Xiaotangshan Township, Changping District, Beijing, China, the hospital occupies an expansive 33 hectares (82 acres) of land, providing ample space for the construction of additional wards and necessary healthcare facilities. The transformation of the sanatorium into a fully-functional hospital was a Herculean effort that began in the early hours of May 1st, 2003.

In a remarkable display of efficiency and coordination, the hospital welcomed and treated 680 SARS patients from across the country within a matter of days. This number represented approximately one-tenth of the global SARS cases and one-seventh of the cases in China at the time. The management of this influx of patients was entrusted to a dedicated team of 1,200 military medical staff, who tirelessly worked to provide the highest standard of care under challenging circumstances.

The renovation and expansion process was a logistical marvel, involving a workforce of 4,000 workers and approximately 500 machinery and equipment from six large construction groups based in Beijing. The project was completed in just eight days, showcasing the incredible speed at which the Chinese government and its people were able to mobilize resources and infrastructure in response to the health emergency.

The design capacity of the newly minted Xiaotangshan Hospital was impressive, with the facility capable of accommodating up to 4,000 people (CGTN Nwes, 2020). This included provisions for 1,000 SARS patients, 1,200 medical staff, and an additional 1,800 logistics support staff. The hospital was equipped with state-of-the-art medical facilities and isolation units, ensuring that patients received the care they needed while also minimizing the risk of cross-contamination.

The transformation of Xiaotangshan Sanatorium into Xiaotangshan Hospital during the SARS outbreak serves as a testament to the resilience and ingenuity of the

Chinese people in the face of a daunting challenge. It also highlights the importance of having flexible and scalable healthcare infrastructure in place to respond effectively to pandemics and other public health emergencies. The lessons learned from this experience have undoubtedly informed and influenced the construction and preparation of similar facilities in response to the COVID-19 pandemic, ensuring that the world is better prepared to tackle such crises in the future.

The Xiaotangshan SARS Hospital, a beacon of rapid response and adaptability in the face of a health crisis, was a testament to the innovative use of lightweight materials and the efficient allocation of space. Constructed primarily as a single-storey facility, the hospital was meticulously designed to maximize functionality while ensuring the safety of patients and healthcare professionals.

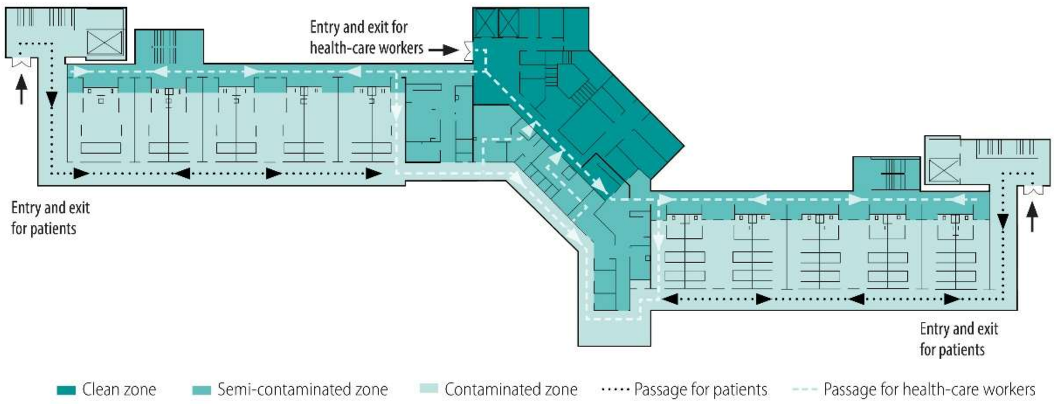



Figure 3.1 Partial layout of Xiaotangshan hospital (Luo, Z. et al., 2021)

The hospital was strategically divided into three distinct areas, each with its own purpose and level of security. The tightly controlled area was dedicated to the patients, ensuring that their conditions were closely monitored and properly managed. This area was strictly off-limits to anyone not involved in patient care, thus minimizing the risk of cross-contamination and maintaining a controlled environment. In Figure 3.1, depicts a zoning and pathway system designed for infection control in a healthcare setting. It divides the facility into three distinct zones: the Clear Zone, which is a safe, uncontaminated area for staff and non-infected individuals; the Semi-Contaminated Zone, a transitional space for putting on or removing personal protective equipment

(PPE); and the Contaminated Zone, a high-risk area where infected patients are treated. The diagram also highlights separate pathways for healthcare workers and patients, with designated entry and exit points for each group. This segregation ensures controlled movement between zones, minimizing cross-contamination and maintaining strict infection prevention protocols. The one-way flow of movement is typical in outbreak or isolation scenarios, emphasizing safety for both medical staff and patients. This system reflects standard practices for airborne or droplet precaution settings in hospitals.

The Xiaotangshan SARS Hospital stands as a remarkable example of how a swift and coordinated response can mitigate the impact of a pandemic. Its design and layout reflect a deep understanding of the importance of containment, separation, and the provision of a safe and comfortable environment for both patients and healthcare providers. The hospital's existence serves as a testament to the resilience and innovation of the healthcare sector in times of crisis, and its lessons continue to inform and inspire responses to subsequent health emergencies (Mu Fei, 2020).

In response to the unprecedented challenges posed by the COVID-19 epidemic, the Xiaotangshan SARS Hospital, a facility that had been decommissioned for 17 years, was brought back into service. This decision to reopen the hospital was a testament to the urgent need for additional healthcare infrastructure to combat the surging number of cases. Workers were mobilized to restore and rebuild the internal facilities, ensuring that the hospital would be ready to meet the demands of the ongoing health crisis.

The Xiaotangshan SARS Hospital, with its extensive capacity of more than 1,000 beds, was repurposed to serve as a vital hub for the screening and treatment of individuals in need. It played a crucial role in the containment strategy, focusing on the screening and care of suspected cases, as well as light and common confirmed patients among those returning to Beijing from overseas.

The hospital's large bed capacity allowed for the efficient processing of a significant number of patients, alleviating the burden on existing healthcare facilities that were quickly overwhelmed by the influx of cases. The facility was equipped with the necessary medical equipment and staffed with healthcare professionals who were trained to handle the specific challenges posed by COVID-19.

The Xiaotangshan SARS Hospital served as a model for rapid response and adaptability in the face of a rapidly evolving health crisis. Its reopening and reconfiguration highlighted the importance of having flexible and scalable healthcare infrastructure in place to address the needs of a pandemic. The hospital's role in screening and treating patients helped to slow the spread of the virus and protect the broader community, showcasing the effectiveness of targeted and decisive action in managing public health emergencies.

The experience gained from the operation of the Xiaotangshan SARS Hospital during the COVID-19 epidemic will undoubtedly inform and influence future preparedness and response efforts. It serves as a valuable lesson in the importance of planning and preparing for potential health crises, ensuring that societies are better equipped to handle similar challenges in the future.

## 3.2 Huoshenshan Hospital and Leishenshan Hospital in Wuhan

Huoshenshan Hospital stands as a remarkable testament to the swift and decisive action that can be taken in the face of a global health crisis. Constructed between January 23 and February 2, 2020, the hospital was specifically designed and purpose-built to cater to the unique needs of the COVID-19 pandemic, which was rapidly escalating in China at the time(Chen, C. et al., 2020). This emergency specialty field hospital was a response to the urgent demand for dedicated healthcare facilities that could effectively treat and manage cases of COVID-19, ensuring that regular hospitals were not overwhelmed and could continue to provide care for other health issues.

The design and construction of Huoshenshan Hospital were meticulously planned to maximize efficiency and functionality in a challenging environment. The hospital was staffed by a dedicated team of 1,400 medical personnel from the Chinese People's Liberation Army, who brought their expertise and military discipline to the task of managing and operating the facility. This military involvement highlighted the severity of the situation and the all-hands-on-deck approach that was required to combat the pandemic.

Following in the footsteps of Huoshenshan Hospital, Leishenshan Hospital was established on February 8, 2020. This second field hospital utilized the same design as its predecessor, which allowed for rapid replication and standardization of the healthcare infrastructure needed to fight the virus. The identical design ensured that the construction process could be expedited while maintaining the necessary quality and functionality of the facilities.

In addition to these two field hospitals, a further 16 temporary treatment facilities were established in Wuhan (Chen, Fei, 2020). These facilities were repurposed from existing buildings and were specifically designed to isolate and treat COVID-19 cases. This multipronged approach to healthcare infrastructure expansion provided a comprehensive network of care that could cater to the varying needs of patients affected by the virus.

The establishment of Huoshenshan Hospital and the subsequent Leishenshan Hospital, as well as the conversion of existing buildings into temporary treatment facilities, demonstrated the Chinese government's commitment to addressing the healthcare needs of its citizens in the face of the COVID-19 pandemic. These facilities not only provided much-needed care for those infected but also served as a critical component in containing the spread of the virus. The speed and efficiency with which these hospitals were built and operationalized represented a significant milestone in the global response to the pandemic and served as an inspiration for other countries facing similar challenges.

The experience gained from the operation of Huoshenshan Hospital and its counterparts has left a valuable legacy, offering lessons in rapid response, effective healthcare delivery during emergencies, and the importance of preparedness in the face of unforeseen health crises. The world continues to grapple with the impacts of COVID-19, and the example set by China in creating these field hospitals and temporary treatment facilities remains a powerful reminder of the potential for innovation and resilience in the face of adversity.

Huoshenshan Hospital, a beacon of emergency medical response and innovation, was designed and constructed with the urgency of the COVID-19 pandemic in mind. Drawing inspiration from the successful model set by the Xiaotangshan Hospital during the 2003 SARS epidemic, the design team was able to create a blueprint that would be both efficient and effective in addressing the current crisis. The design process, which took a mere 60 hours, was a testament to the rapid mobilization of resources and the collective ingenuity of the architectural and medical communities.

The hospital, spread over an impressive area of approximately 60,000 square meters, comprises two floors and is capable of accommodating up to 1,000 beds. Its infrastructure is robust, featuring 30 intensive care units, dedicated medical equipment rooms, and quarantine wards. This comprehensive layout ensures that patients receive specialized care while maintaining the necessary isolation to prevent the spread of the virus.

The final design of the hospital was finalized at 33,900 square meters (Chen, Fei, 2020), with a functional layout that includes wards, a reception room, an intensive care unit (ICU), a medical technology department, a network computer room, a supply warehouse, a garbage temporary storage room, and an ambulance washing room, among other essential facilities. This design was crafted to support the efficient flow of patients, medical staff, and resources, optimizing the hospital's functionality in a high-stress environment.

Adopting the fishbone layout from the Xiaotangshan Hospital design, the architects created a temporary board room that facilitated communication and coordination among the various departments. The building's structure is characterized by standardization, modularization, and extensibility, allowing for quick adjustments and scalability in response to the evolving needs of the pandemic.

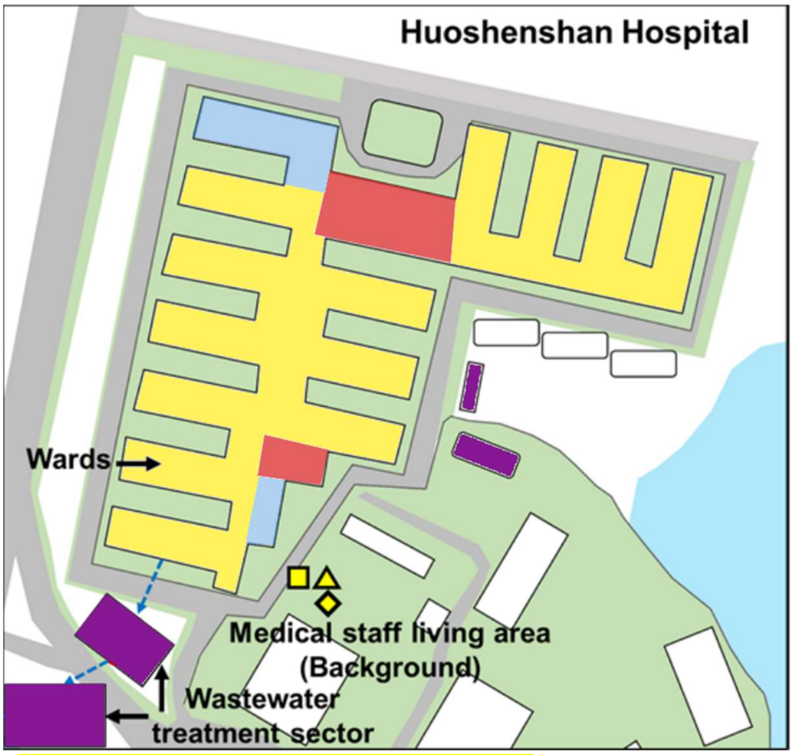



Figure 3.2 Huoshenshan overall Layout (Zhang, Dayi et al., 2020)

Facilities include isolation wards, an intensive care unit (ICU), laboratories and medical support facilities. The development of the hospital demonstrates the innovation of modular construction and the use of prefabricated units for rapid deployment. This Figure 3.2 shows the layout of Huoshenshan Hospital.

The yellow area, which is the most prominent and covers the most extensive area in the diagram, is the ward building. In red is the ICU treatment area, whose main function is to receive critically ill patients. The blue area, connected to the wards, is the medical technology area, the main function of which is to prepare the necessary medicines and instruments for the medical staff, and also to take care of the cleaning function of the medical staff after their work. This layout ensures that the virus does not spread and that critically ill patients receive timely and appropriate treatment. The purple area is the medical equipment spare parts store, while instrument sterilization is also carried out here.

The Figure 3.3 is the general layout of patient area in Huoshenshan hospital, is composed of multiple "H"-shaped modules, each containing a central module and four corresponding nursing units. This configuration allows for an efficient distribution of resources and a streamlined patient care process. Each nursing unit features two rows of beds, with patients entering from the periphery of the ward to minimize the risk of contamination.

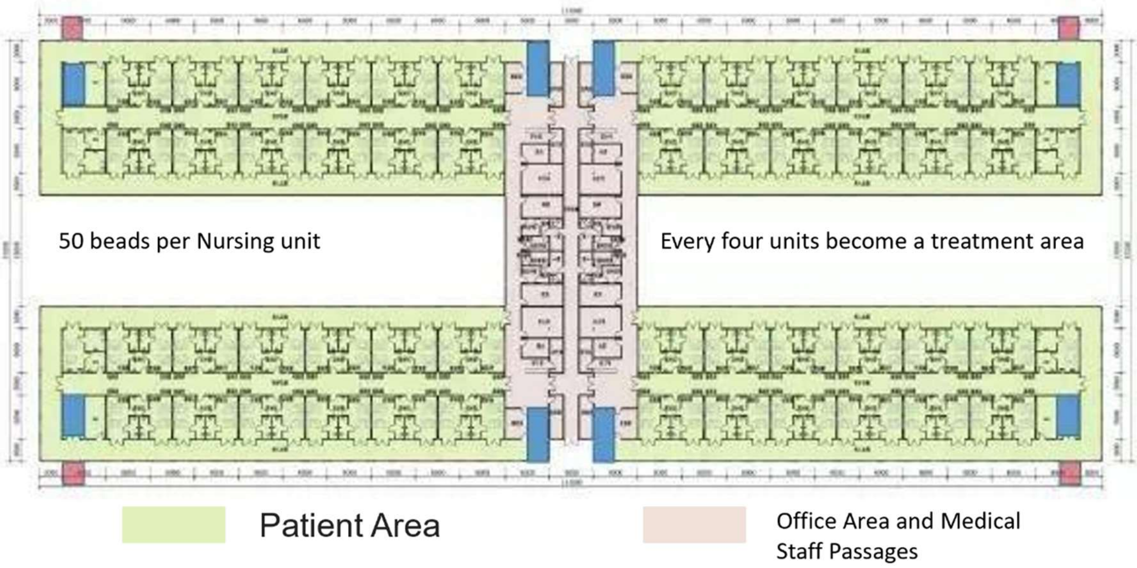



Figure 3.3 Patient area layout (Chen, C. et al., 2020)

The office area and medical staff passages are strategically positioned along the central axis of the building, ensuring a clear and efficient path for medical staff as they move between wards. To maintain air cleanliness and protect the health of medical personnel, staff must undergo thorough procedures such as changing clothes before entering the ward. This careful planning not only enhances safety but also improves the overall efficiency of patient treatment and care.

From Figure 3.3, it can be seen that the ward building is connected to both sides of the building by a central corridor, and that a ward 'branch' is designed as a therapeutic unit, with patients with the same symptomatic stage generally being placed in the same therapeutic unit, which is conducive to real-time monitoring of the patient's status by the medical staff. The diagram below shows the internal layout of the ward building.

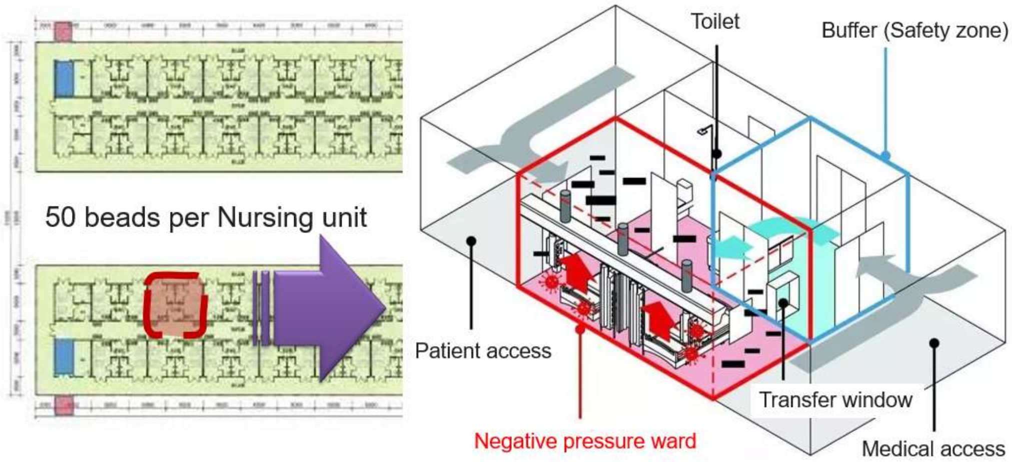



Figure 3.4 A ward functional space (Chen, C. et al., 2020)

To enable rapid construction, the Huoshenshan Hospital uses a modular assembly building with standard container dimensions of 3 meter by 9 meter, along with seven different split walls components for different functional zoning, those walls are shown in Figure . Unit A is the ward, Unit B is the restroom and buffer area, Unit C makes up the corridor, and two Unit C's and one each of Unit A and Unit B make up a cluster of wards in an effective epidemiologic hospital (Zhou et al., 2020).

In the face of the COVID-19 pandemic, the establishment of field hospitals became a critical strategy for many countries and regions around the world. These temporary healthcare facilities were designed to provide immediate medical services and support the overflow of patients that traditional hospitals were struggling to accommodate.

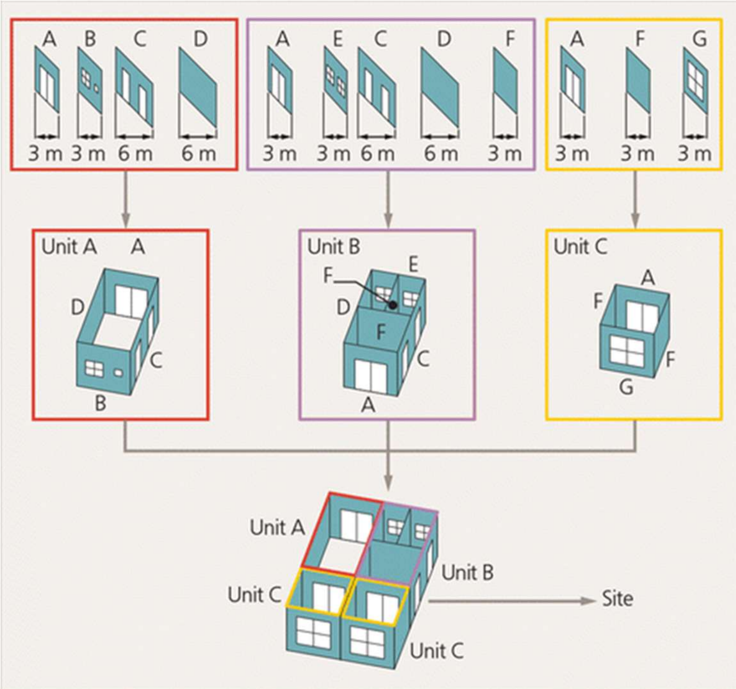



Figure 3.5 Walls type and Unit assembles (Zhou et al., 2020)

Following in the footsteps of Huoshenshan Hospital, Leishenshan Hospital was also quickly constructed to address the surge in COVID-19 cases. Iran's Qom, South Korea's Daegu, and Hong Kong took a different approach by using shipping containers to create temporary field facilities. These container hospitals, as they are sometimes called, offered a prefabricated and easily deployable solution that could be quickly set up in areas with a high density of cases. The use of containers allowed for rapid construction and provided a flexible space that could be easily expanded or moved as needed.

Kuwait, Italy and Spain also responded to the crisis by converting existing buildings into temporary healthcare facilities(Brambilla et al., 2023; Hamadah et al., 2020; Zangrillo et al., 2020). This approach allowed these regions to quickly repurpose available spaces, such as convention centres, sports arenas, and government buildings, into much-needed medical centres. By doing so, they were able to maximize the use of existing infrastructure while meeting the urgent demand for additional healthcare capacity.

Moscow, Russia, faced with a rising number of COVID-19 cases, also announced the construction of a field hospital(Reshetnikov et al., 2020). This move was part of the country's efforts to bolster its healthcare system and provide adequate care for the increasing patient load. The construction of such hospitals was not only a logistical challenge but also a demonstration of the commitment to protect the health and well-being of citizens.

The rapid deployment of these field hospitals and temporary facilities highlighted the importance of adaptability and innovation in times of crisis. It showed that in the face of a global health emergency, communities could quickly come together, leveraging available resources and technology to create effective healthcare solutions. These temporary facilities not only provided immediate relief but also bought valuable time for healthcare systems to prepare for and manage the ongoing pandemic.

The architectural layouts of Wuhan's Leishenshan Cabin Hospitals and Beijing's Xiaotangshan Hospital share fundamental similarities in their emergency response designs but also exhibit key differences tailored to their respective pandemic contexts. Both facilities prioritized rapid modular construction, with Xiaotangshan using prefabricated container wards and Leishenshan employing a hybrid of prefab units and repurposed structures. They strictly implemented the "Three Zones and Two Passages" infection control system (clean/semi-contaminated/contaminated zones with separated staff/patient pathways) to contain transmission. However, Leishenshan represented a significant evolution in scale and sophistication, while Xiaotangshan (built for SARS) housed about 1,000 beds in a single centralized compound with basic ventilation, Leishenshan (COVID-19 response) expanded to 1,600 beds with enhanced negative-pressure systems, smart hospital technologies including IoT patient monitoring, and specialized ICU pods for potential critical case escalation. The Wuhan facility also incorporated more comprehensive psychological support spaces and outdoor circulation areas absent in the earlier SARS-era design. These differences reflect how China's emergency hospital architecture progressed from Xiaotangshan's prototype of isolated containment to Leishenshan's integrated approach balancing massive patient capacity with advanced clinical capabilities and humanitarian considerations during a global pandemic.

The experiences of these regions in setting up field hospitals and temporary healthcare facilities offer valuable lessons in emergency medical response. They serve as examples of the ingenuity and resilience required to address the challenges posed by a rapidly spreading virus. As the world continues to grapple with the impacts of COVID-19, the lessons learned from these efforts will undoubtedly inform future preparedness and response strategies, helping to ensure that societies are better equipped to handle similar health crises in the future.

## 3.3 Nightingale Hospital（ExCel London）in London

BDP, in collaboration with an array of healthcare professionals, advisors, contractors, and the British Army, along with the ExCel facilities management team, undertook the herculean task of transforming the ExCel Centre into a fully functional hospital capable of accommodating 500 beds. The hospital was rapidly built in just nine days and officially opened on April 3, 2020 (Wise, 2021). This facility was not only designed to meet the immediate needs of the healthcare system but also with the foresight to be expandable up to 4,000 beds, if necessary. The hospital is divided into wards, each equipped with rows of beds and the necessary medical equipment, including ventilators and oxygen supplies. The speed at which this hospital had to be established necessitated the efficient use of the building's existing infrastructure. Although the facility was intended for a large number of patients, the number of patients ultimately admitted was less than its maximum capacity. The Figure 3.6 is inside of Excel after converting.

In order to achieve this rapid deployment, the design and construction teams focused on minimal intervention, ensuring that the building's assets were utilized to their full potential. Within the vast wards, bed spaces were demarcated using a system reminiscent of temporary exhibition stands, with simple reinforcements added to accommodate the necessary services and fittings. This approach allowed for quick and efficient installation of medical equipment and infrastructure.


Figure 3.6 Nightingale Hospital（ExCel London）(Wise, 2021)

One of the most complex aspects of the conversion process was the integration of electricity and other essential services to each bed station. The ExCel Centre's existing electrical infrastructure was significantly modified to enhance resilience and accommodate the additional load required for a hospital environment. This included the installation of uninterruptible power supply (UPS) systems and temporary generators to ensure a continuous power supply, which is critical in a healthcare setting.

The creation of negative pressure wards was a crucial measure to control the spread of the virus within the hospital. These wards are designed to ensure that air flows into the room but does not escape back into the general circulation, thus containing potential infections.

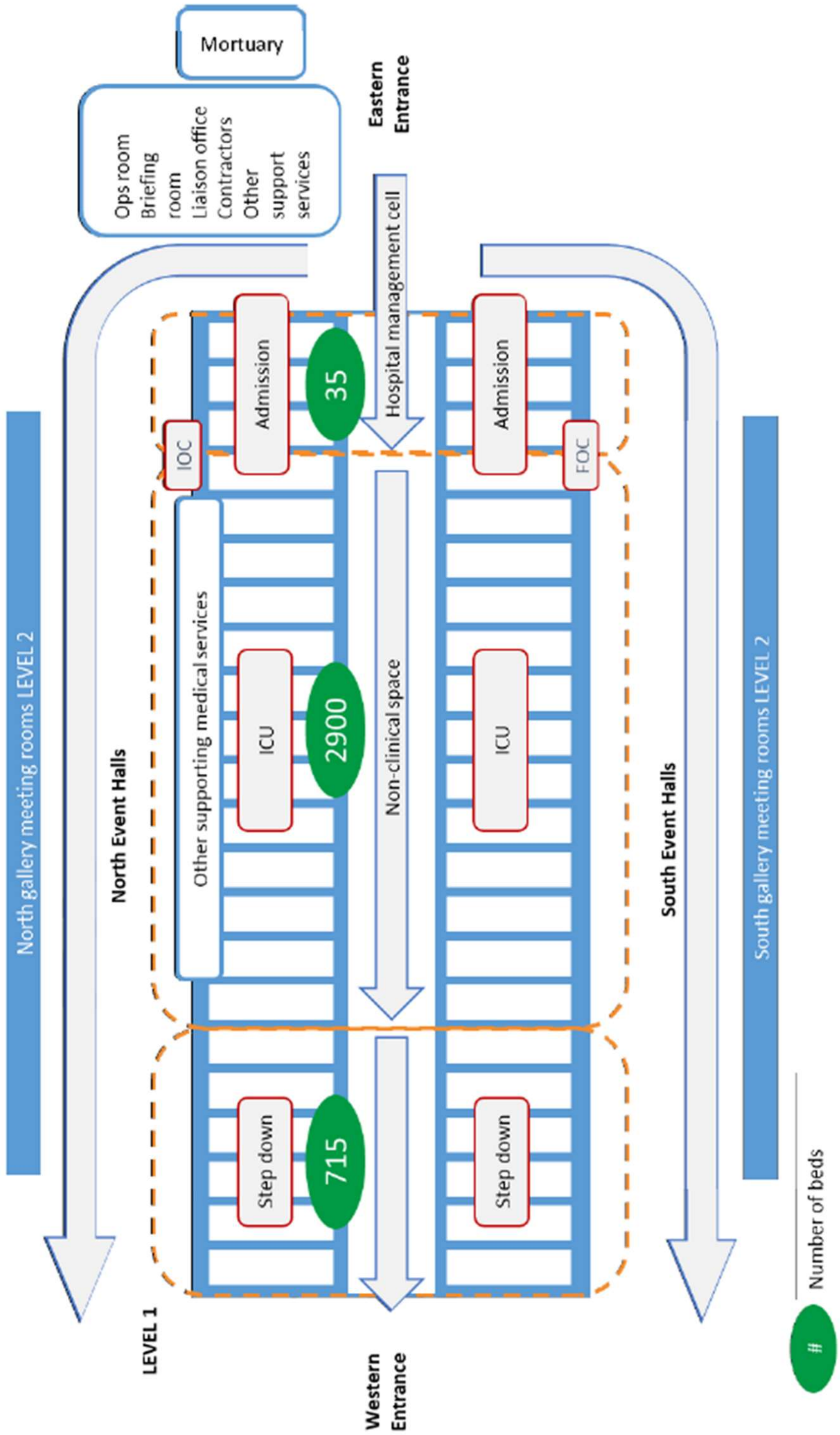



Figure 3.7 Nightingale Hospital（ExCel London）layout (Wise, 2021)

The distribution of medical gases, such as oxygen, is another essential requirement in a hospital. To meet this need, two distribution ring mains were installed around the basement car park, running at a high level to feed each bed head through service floor boxes. These mains then distributed the gases to each bed via service corridors located between the bed spaces.

Figure 3.7 shows the internal layout of the hospital, and it can be seen that the hospital only uses functional zoning in its spatial layout, and does not strictly control the direction of people and the path of air transfer as the Huoshenshan hospital does.

The wards also use only unenclosed partitions for patient privacy. This choice of an existing site for conversion means that the layout is simple and quick to build, the facilities are easy to install, and the site can be quickly dismantled and returned to its convention function when the hospital's mission is complete.

The successful conversion of the ExCel Centre into a hospital was only possible due to the swift decision-making of the project team. The design and construction processes were carried out in parallel, allowing for the efficient use of time and resources. This collaborative effort and the innovative solutions implemented are a testament to the resilience and adaptability of the construction and healthcare sectors in times of crisis. The ExCel Centre's transformation serves as an inspiring example of what can be achieved through coordinated teamwork, technical expertise, and a commitment to meeting the urgent needs of the community.

## 3.4 Summary

All the cabin hospital described above were rapidly constructed or repurposed facilities aimed at addressing the urgent need for additional healthcare capacity. They emphasized modularity and scalability, enabling quick deployment and flexibility in patient intake. For instance, Huoshenshan and Leishenshan Hospitals utilized prefabricated units and standardized designs, while Nightingale Hospital repurposed an existing convention center with minimal structural changes. Additionally, all three cases prioritized infection control, though to varying degrees. The Chinese hospitals strictly adhered to the "Three Zones and Two Passages" system to segregate clean, semi-contaminated, and contaminated areas, while Nightingale Hospital focused on open wards with temporary partitions, reflecting differences in spatial constraints and design priorities.

Despite these similarities, there were notable differences in their layouts and operational approaches. Xiaotangshan and Huoshenshan Hospitals, built during the SARS and COVID-19 outbreaks respectively, featured centralized, single-story designs with strict zoning for infection control. Huoshenshan and Leishenshan Hospitals expanded on this model, incorporating advanced features like negative-pressure systems and IoT monitoring. In contrast, Nightingale Hospital's layout was simpler, leveraging the existing infrastructure of the ExCel Centre to create large, open wards with temporary bed stations. This approach allowed for rapid conversion but lacked the granular infection control measures seen in the Chinese hospitals. The differences highlight how context—such as available time, resources, and local pandemic severity—shaped each facility's design.

These hospitals were chosen as representative examples because they demonstrated innovative solutions to the challenges posed by the pandemic. Xiaotangshan and Huoshenshan Hospitals showcased the effectiveness of modular construction and strict zoning in containing outbreaks, while Nightingale Hospital illustrated the potential of repurposing large public spaces for emergency healthcare. Their layouts were tailored to their specific needs: the Chinese hospitals prioritized containment and scalability, while Nightingale emphasized speed and adaptability. The success of these models influenced global responses, proving that rapid construction and flexible design are viable strategies during crises.

The implementation of these cabin hospitals had a profound impact on the construction industry. They underscored the importance of modular and prefabricated building techniques, which reduce construction time and costs while maintaining quality. The use of Building Information Modeling (BIM) and ontology-driven design, as seen in Huoshenshan Hospital, highlighted the potential for technology to streamline complex projects. Additionally, the repurposing of existing structures, as demonstrated by Nightingale Hospital, encouraged the industry to explore adaptive reuse as a sustainable and efficient solution. These innovations have set new benchmarks for emergency architecture, emphasizing the need for preparedness, interdisciplinary collaboration, and scalable design in future construction projects. The legacy of these hospitals extends beyond the pandemic, influencing how the industry approaches rapid-response infrastructure and resilient healthcare systems.

Through understanding of COVID-19 resilient cabin hospitals in China and the UK, evaluation will be undertaken by the following aspects: construction speed and approach; scale and capacity; purpose and patient population; healthcare staffing; infection control measures; local community involvement; adaptability and legacy and long-term integration.

China's construction model of makeshift hospitals has garnered international attention and praise for its effectiveness in responding to public health emergencies, particularly during the COVID-19 pandemic. These makeshift hospitals, often referred to as "fangcang" (cabin) hospitals, have served as a crucial component of China's rapid response to outbreaks. One of the most remarkable aspects of China's makeshift hospital construction model is the speed at which these facilities are built. In a matter of days, these hospitals can go from an empty space to a fully operational medical facility, which is essential for addressing a sudden surge in patients during a public health crisis. China often repurposes existing structures like convention centers, gymnasiums, and stadiums to set up makeshift hospitals. This approach makes efficient use of available resources and minimizes the need for entirely new construction. The design of these makeshift hospitals allows for easy scalability. They can be expanded or contracted to meet the evolving needs of a public health emergency. This adaptability ensures that resources are used effectively and efficiently. These facilities are designed to cater to specific needs, such as isolating and treating patients with infectious diseases. They are equipped with appropriate medical equipment and isolation measures to protect both patients and healthcare workers. China's construction model often involves cooperation between the government and private construction companies. This partnership helps streamline the construction process and harnesses the expertise and resources of both sectors. Maintaining strict infection control measures is a priority in these makeshift hospitals. This helps prevent cross-infection among patients and healthcare workers, which is crucial during disease outbreaks. China's makeshift hospitals are often equipped with advanced monitoring and data collection systems, which aid in patient care and data analysis. This information can be used to make informed decisions about resource allocation. Community involvement and support are key factors in the success of these makeshift hospitals. The local population is often mobilized to provide essential services, and their cooperation is essential for the overall effectiveness of the model. While this construction model has been praised, it has also faced criticism and challenges, including concerns about the speed of construction potentially sacrificing quality, as well as issues related to transparency and accountability. It's important to consider that the success of China's makeshift hospital construction model may not be directly transferable to other countries or regions due to differences in healthcare infrastructure, governance, and social dynamics. Any evaluation should consider the unique circumstances of the area in which it's implemented. In summary, China's construction model of makeshift hospitals has demonstrated its effectiveness in responding to public health emergencies. However, it is important to critically evaluate and adapt this model to the specific needs and conditions of other regions or countries if it is to be applied elsewhere.

The construction model of the Nightingale Hospital at ExCeL London was a significant effort to respond to the COVID-19 pandemic in the United Kingdom. The Nightingale Hospital at ExCeL London was constructed in an extremely short period, showing the ability to rapidly adapt and respond to the surge in COVID-19 cases. The speed of construction was a critical aspect, given the urgency of the pandemic. The use of an existing event space, the ExCeL London exhibition center, for the hospital allowed for efficient repurposing of a large area. This minimized the need for new construction and saved time and resources. The facility was designed to be scalable and modular, allowing it to adapt to changing needs. This adaptability is essential during a health crisis when patient numbers can fluctuate rapidly. The hospital was equipped with the necessary medical equipment, including ventilators and monitors, to provide care to critically ill COVID-19 patients. The availability of such equipment was crucial to its functionality. Adequate staffing and training of healthcare professionals are essential for the successful operation of a hospital. Evaluating the readiness and capability of the healthcare workforce is crucial. Effective infection control measures, including isolation units and personal protective equipment (PPE), were crucial in the design and operation of the Nightingale Hospital to prevent cross-infections among patients and healthcare workers. Evaluating how the Nightingale Hospital integrates with the broader healthcare system in London and the UK is important. It should be considered how patients were referred, transported, and transferred to and from the facility. Assessing the involvement of the local community and volunteers in supporting the hospital's operation, including the provision of essential services, can shed light on its success. It is important to evaluate the transparency and accountability of the construction and operation of the Nightingale Hospital, including how resources were allocated and decisions were made. Consideration should be given to whether the Nightingale Hospital was a short-term solution or if it has been integrated into long-term healthcare planning to ensure preparedness for future crises. Evaluating the clinical outcomes and patient experiences at the Nightingale Hospital can provide insights into the quality of care and the effectiveness of the model. Examining the lessons learned from the construction and operation of the Nightingale Hospital is essential to improve future responses to similar crises. In summary, evaluating the construction model of the Nightingale Hospital at ExCeL London involves assessing its speed, adaptability, functionality, and impact on public health during the COVID-19 pandemic. The success and effectiveness of such models may vary depending on specific circumstances and the local healthcare infrastructure.

The Table 3-1 below shows the comparison between the two:

Table 3-1 Comparison of Cabin hospital and Nightingale hospital

| **Resilient Hospital** | **“Fangcang” Hospital** | **Nightingale Hospital at ExCeL London** |
|---|---|---|
| **Origin and Naming** | These were rapidly constructed temporary hospitals in China, with the name "fangcang" | Named after Florence Nightingale, the pioneer of modern nursing, these facilities in the UK aimed to |

| | | |
|---|---|---|
| | originating from the Chinese words for " cabin." | honor her legacy and provide care during the COVID-19 crisis. |
| **Construction Speed and Approach** | China rapidly converted existing structures, such as stadiums and exhibition centers, into healthcare facilities. Construction was completed within days. | The UK converted existing spaces, such as exhibition centers, into hospitals, with construction taking a few weeks. The focus was on speed but was less rapid than the Chinese model. |
| **Scale and Capacity** | China's Fangcang hospitals were designed to accommodate thousands of patients, making them among the largest temporary facilities in the world. | The capacity of Nightingale Hospitals varied between locations, but they were generally smaller in scale than their Chinese counterparts. |
| **Purpose and Patient Population** | Initially designed for the isolation and treatment of COVID-19 patients, they also served as quarantine facilities and provided basic medical care for mild to moderate cases. | Primarily designed to provide critical care for COVID-19 patients, with a focus on patients requiring mechanical ventilation. They did not accommodate patients with mild symptoms. |
| **Healthcare Staffing** | China mobilized healthcare workers from various regions to staff these facilities. They were generally well-staffed with nurses and doctors. | The staffing model in the UK involved a mix of healthcare professionals, including volunteers, military personnel, and recently retired healthcare workers. |
| **Infection Control Measures** | Strict infection control measures were implemented to prevent cross-infection among patients and healthcare workers. | Infection control was a priority, with measures taken to minimize the risk of transmission within the facilities. |
| **Local Community Involvement** | Local communities played a role in supporting these facilities and providing essential services. | Local community involvement and support varied by location. |
| **Adaptability** | The Fangcang model demonstrated adaptability and scalability, expanding or contracting to meet changing needs. | While adaptable, they were primarily designed for critical care and less suited for non-critical cases. |
| **Legacy and Long-Term Integration** | These facilities highlighted China's ability to respond swiftly to public health crises and may serve as a model for future responses. | The legacy of Nightingale Hospitals may influence future crisis preparedness in the UK, and some facilities have been integrated into regional healthcare planning. |

In summary, while both the Fangcang Hospitals in China and the Nightingale Hospitals in the UK were constructed as temporary healthcare facilities during the COVID-19 pandemic (Zampetakis et al., 2012a), there are differences in terms of scale, purpose, staffing, and legacy. The models were adapted to their respective healthcare systems and local needs. These different types of cabin hospitals basically integrate the characteristics of cabin hospitals in most regions. These background knowledge will provide theoretical and knowledge support for the establishment of the spatial ontology of cabin hospital in the following chapters, and also provide reference for the subsequent automation layout as a condition of spatial constraints.

# Chapter 4 Methodology

This chapter describes the research methodology used in this study. Section 4.1 summarizes the methodology used and Section 4.2 explains the main methodology used for the development of the ontology knowledge base. This study reused existing knowledge extracted from two pre-existing ontologies to generate layouts.

As analysed in Chapter 2, ontologies are used in various areas of the AEC (Architecture, Engineering and Construction) industry. However, most of the existing ontologies in the AEC industry are lightweight and primarily focus on the structured representation of domain-specific knowledge or information, such as compliance checking or assessment. Furthermore, research on functional space layout is still in its infancy. In addition, this study aim to develop an ontology for generating spatial layouts for pandemic-resistant hospital wards and incorporates constraints into the generation process to meet project requirements under different design scenarios. This research aims to shorten the engineering design phase and provide decision support for subsequent on-site construction arrangements. The findings of this research enhance the reuse of ontologies and knowledge The findings of this research will enhance the reuse of ontologies and knowledge. A key feature of ontologies will be their ability to leverage existing knowledge. The future implementation will significantly reduce the cost and time required for domain-specific conceptualization compared to building from the ground up, thereby increasing efficiency. Knowledge about objects, spatial relationships, and constraints will be successfully extracted from two existing domain ontologies. This knowledge will prove to be sufficiently general to generate planning and design layouts at various scales.

## 4.1 Overview of Methodology

Initially, several advanced ontology development methods are reviewed. Since the 1990s, various methods have been established to construct ontologies (Zhipeng et al., 2016). The more commonly used methods by timeline for building ontology are Grüninger and Fox method (Uschold, Michael & King, 1995), “METHONTOLOGY” system (Fernández-López et al., 1997c), Uschold and Gruninger method (Uschold, Mike & Gruninger, 1996), and "simple knowledge engineering methodology" (SKEM) (Noy, Natalya F. & McGuinness, 2001). To some extent, these techniques overlap, and the majority of the steps encompass term specification acquisition, conceptual integration, implementation, and evaluation. SKEM furnishes a comprehensive procedure for developing an ontology, covering classes, attributes, and axioms, shows in Table 4-1.

Table 4-1 Commonly used ontology building methods

| | |
|---|---|
| **Grüninger and Fox** | Capture of incentive scenarios, formulation of informal capability questions, specification of ontology terms, formulation of formal capability questions using ontology terms, specification of axioms, and definition of terms in ontologies |
| **METHONTOLOGY** | Specification, knowledge acquisition, conceptualization, integration, implementation, evaluation and documentation |
| **Uschold and Gruninger** | Identifying purpose and scope, building ontologies, integrating existing ontologies, evaluating ontologies, and providing documentation |
| **SKEM** | Determine the domain and scope of the ontology, consider reusing existing ontologies, enumerate important terms in the ontology, define classes and class hierarchies, define attributes of classes, define aspects of slots, create instances |

In this research, a method that combines the above approaches has been used, which is shown in Figure 4.1, meanwhile using semantic knowledge of 3D layout from Luca (Caneparo, 2022b). Luca presented an approach that segregates information regarding objects, spatial relationships, and constraints from the generative process. This enhances the spatial layout in design and planning. Distinguishing between knowledge and practical application is a crucial aspect of semantic technologies, which enable access to a vast quantity of knowledge preserved in formal ontologies.

As the key link to ensure the quality and efficiency of medical service, the hospital layout design has been paid more and more attention to. Traditional hospital layout design often considers the static space demand, neglects the dynamic demand of medical activities and the elastic demand of space, and is difficult to adapt to the rapidly changing medical environment and service mode. Therefore, the introduction of ontology and the construction of ontology-based cabin hospital layout design method, it is particularly urgent and meaningful.

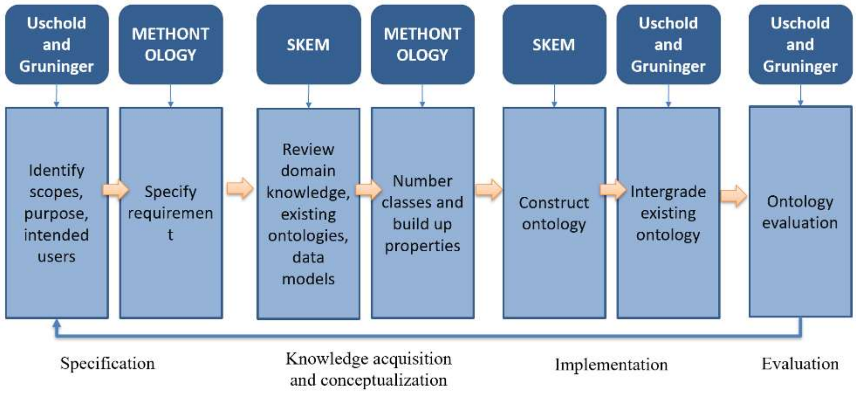



Figure 4.1 Commonly used ontology building methods

The decision to adopt a mixed approach for ontology development in this research stems from its ability to comprehensively address the multifaceted challenges of cabin hospital layout design. By integrating established methodologies such as Grüninger and Fox's scenario-driven axioms, METHONTOLOGY's systematic evaluation framework, Uschold and Gruninger's integration principles, and SKEM's structured class definitions, this approach will ensure thorough coverage of all ontology development stages while maintaining flexibility to adapt to specific project requirements.

The hybrid method will emphasize the reuse of existing knowledge, incorporating semantic principles from Luca's 3D spatial layouts and leveraging prior domain ontologies to enhance efficiency and reduce redundancy. This strategy will prove particularly valuable for addressing the dynamic nature of medical space requirements, enabling adaptive planning through clear definitions of concepts, attributes, relationships, and constraints.

By bridging theoretical ontology constructs with practical design applications, the method will facilitate intelligent decision-making for spatial configurations while maintaining scalability for different hospital sizes and emergency scenarios. This strategic combination of methodologies will ultimately create a robust framework that balances rigorous knowledge representation with real-world implementation needs, making it uniquely suited to tackle the evolving demands of epidemic-resilient healthcare facility design.

Through this approach, the research will achieve its goal of developing an ontology-driven system that optimizes cabin hospital layouts, ensuring rapid deployment, flexibility, and efficiency in response to public health emergencies. The successful implementation of this framework will set a new standard for emergency healthcare infrastructure planning and demonstrate the transformative potential of semantic technologies in architectural design.

In the design of cabin hospital layout, the introduction of ontology aims to construct an information knowledge system that meets the essential needs of medical services, through which different sources and formats of data and knowledge can be integrated and managed, provide dynamic and intelligent decision support for design. By defining the related concepts, attributes, relationships and rules in hospital layout design, ontology can help designers to precisely locate, dynamically adjust and optimize the management of medical space.

Furthermore, based on the ontology model, a set of matching algorithms will be developed to accurately calculate the spatial relations and constraints among different medical functional units. The algorithm will be designed to be applicable to all kinds of hospitals, enabling it to automatically identify contradictions and conflicts in spatial layouts while facilitating dynamic optimization and intelligent decision-making through the matching and adjustment of a preset rule base.

At the same time, the application of ontology is expected to significantly improve the efficiency and accuracy of data interaction, paving the way for the integrated management of hospital internal design. This advancement will contribute to more streamlined, adaptable, and intelligent healthcare facility planning, ultimately enhancing the responsiveness and resilience of medical infrastructure in crisis scenarios.

By achieving these goals, the research will demonstrate how ontology-driven computational methods can revolutionize hospital layout design, ensuring optimal functionality, safety, and scalability in future epidemic responses.

The review of spatial layouts in various COVID-19 cabin hospitals across China and the UK holds significant value for establishing a comprehensive spatial ontology for such facilities. By analyzing these diverse case studies, including Xiaotangshan, Huoshenshan, and Nightingale hospitals, I can identify common design principles, functional requirements, and infection control measures that are universally critical for epidemic-resilient healthcare facilities. The comparison reveals both standardized approaches (like China's strict "Three Zones and Two Passages" system) and context-specific adaptations (such as the UK's rapid conversion of existing spaces), providing a rich foundation for ontology development. These real-world examples offer concrete evidence of successful spatial configurations, workflow optimizations, and modular design strategies that can inform the ontology's classes, relationships, and constraints. Moreover, the review highlights challenges in interoperability and knowledge fragmentation across different design approaches - precisely the gaps that an ontology aims to bridge. By grounding the ontology in these practical implementations, I ensure it captures not just theoretical spatial relationships but proven, pandemic-tested design solutions that balance rapid deployment with clinical functionality. This empirical foundation will make the resulting ontology more robust, adaptable, and directly applicable to future emergency healthcare facility planning.

Based on the background knowledge of the cabin hospital, a BIM model of the smallest treatment unit will be established. The purpose of this model is to add architectural elements, facilities and other information to the space entity. By introducing the International Organization for Standardization (ISO) definition of the IFC Model as the basis for parameterization, a transformation of the BIM ontology into an OWL ontology is hoped to be achieved, thereby demonstrating information reuse and extension of the two ontologies.

To verify the validity of the ontological model, this study combines formal detection methods and query methods, mainly through structural inspection and random query to obtain the expected reasonable answers. Structural inspection will serve as the formal verification method, systematically examining the ontology's taxonomy, relationships, and constraints to ensure logical consistency and identify any potential contradictions or redundancies. This rigorous examination will validate whether the model properly adheres to domain-specific rules and requirements for hospital layouts. Complementing this, random query testing will evaluate the model's practical utility by simulating real-world design scenarios and assessing whether it generates reasonable, expected responses to diverse spatial planning challenges. This empirical validation method will test the ontology's adaptability and uncover any gaps in knowledge representation that might not be apparent through structural analysis alone. Together, these methods will provide comprehensive validation - structural inspection ensuring the model's theoretical soundness and random query testing confirming its functional performance in practical applications. The combination of these approaches follows established ontology engineering best practices, offering both rigorous formal verification and real-world applicability testing to thoroughly validate the model's effectiveness for hospital layout design.

In order to realize the automatic layout of cabin Hospital, the spatial ontology will provide constraints. The researcher hope to establish an algorithm through the method of Constraint Satisfaction Problem.

The next step is to use examples to test the feasibility of this method. First, in order to verify whether the method is applicable to complex irregular spaces, two hypothetical irregular shapes can be considered. In real environments, emergencies such as epidemics, a sharp increase in demand for hospitals, and insufficient time to build or renovate cabin hospitals are taken into consideration. In the case of lost existing building documents, some high-tech measurement methods can be used. In this research, a method of using mobile robots and depth cameras to obtain physical scenes will be adopted. The scene will then be combined with the results obtained by the method of this research and evaluated using the constraints of the spatial ontology , so that the feasibility of the selected real-life scene can be quickly determined, or the layout arrangement can be adjusted. This process will be much easier if the relevant building documents are already available.

Based on the above research, a new ontology-based cabin hospital layout design method will be proposed, which can fully meet the requirements of flexibility and sustainability of medical services, the utility model effectively improves the utilization efficiency and the service quality of the hospital space. The paper deepens the application of ontology in the field of hospital layout design by combining theory with empirical research, and shows a good research prospect and practical application value. In the future, the application of this method is expected to promote the modernization of hospital management and bring a profound reform in the field of hospital layout design.

## 4.2 Ontology development

Concerning the chosen methodology, the preparation of building information from BIM and spatial layout information from user experience is regarded as crucial for equipping the model with essential operational data. This preparation process aims to establish a data structure that results in a centralized knowledge base, grounded in ontologies, capable of enabling inference mechanisms. This model-centric approach ensures robustness and flexibility within the system.

Therefore, based on the reviewed research (Shu-Hsien Liao, 2005), this research proposes an ontological knowledge-based system. This approach involves three components, users, BIM, and reasoning mechanisms, agreeing on a common ontology that serves as a specification for a shared domain of interest. The ontology facilitates communication between these components, even if they utilize entirely different knowledge representation mechanisms and data exchange formats. Similarly, the objective of the Space Ontology's structure is to formalize knowledge in the field of architectural layout planning.

## 4.3 Space ontology

These reasons will provide sufficient justification for choosing to use an ontology to address problems that other researchers have tried to solve differently (Atkinson et al.). Comparisons with models and databases have been considered in this regard (Benevolenskiy et al., 2012).

(a) Consider ways to reuse existing ontologies. The literature review indicates that various fields require concepts to represent functional spaces, such as ward space and ICU space. Other studies have also addressed these concepts, highlighting the potential for cross-domain reuse.

(b) Ontologies support consistency checking and reasoning, which is a primary goal of the proposed approach. One of the roles of ontologies in systems engineering is to implement "intelligent databases" that offer various reasoning services over data at runtime. Unlike the "data integrity" found in traditional databases, ontologies can perform "consistency checking" and automated reasoning based on predefined rule sets.

(c) Ontologies visually represent knowledge through classes and attributes, a feature that databases lack. This visual representation is crucial for user interaction, especially for project managers who need to engage with the ontologies.

(d) It is easier to represent the complex structure of spatial design processes using ontologies rather than relational databases. The proposed ontology-based expert system aims to be flexible and easily adaptable. For instance, if the system needs to account for additional buildings or new concepts and relationships, or even scenario-based reasoning mechanisms (e.g., layout optimization), ontologies allow for the seamless addition of new entities or scenarios. In contrast, databases would require a complete revision of the table structure.

Furthermore, to ensure the knowledge base is machine-interpretable, a set of validated languages is needed to support the creation of ontologies (formalization of concepts, attributes, and relationships). The most commonly used languages include KIF, F-Logic, RDF(S), and OWL. These languages, while differing in expressive capabilities, all have well-defined syntaxes that make them computable.

All of these ontologies have different expressive capabilities, but all have well-defined syntaxes that make them computable.

In this study, we have selected the Web Ontology Language (OWL) for computing ontologies. OWL is a standard established by the World Wide Web Consortium (W3C) and is currently the most widely used ontology language (Baader). This choice was made for two primary reasons, which are explained below.:

(a) As mentioned above, BIM systems and models are equipped with standardised interfaces for data exchange, i.e. the Industrial Foundation Classes (IFC) standard (Amann & Borrmann, 2015). A number of pilot projects in academic research have experimented with IFC as an OWL ontology for the use of Semantic Web technologies. and described in Schevers and Beetz's work (Beetz et al., 2009b; Schevers & Drogemuller, 2005).

As a result of these research efforts, the ifcOWL ontology has only recently become available. This development enables practical data exchange between a given BIM and our model.

(b) The space ontology can leverage the ontology underlying the BIM, enhancing the robustness of the expert system. This approach allows our modeling domain (classes, relations, and attributes) to be potentially linked to and enriched by the logical and geometric relationships between building objects contained in the BIM ontology (ifcOWL).

Therefore, in this study, the steps and corresponding deliverables for the development of the ontology are explained below:

Step 1) Survey of Knowledge Resources

This step involves reviewing existing ontologies, taxonomies, and other sources within the construction domain to assess their reusability.

Step 2) Objective Specification

The objective of this step is to determine the classes, relations, and attributes that the ontology will comprise. This is achieved by addressing a set of competence questions, such as: Why should the ontology be built? What kind of information should it include? To clearly structure the objectives, a graphical representation is proposed for each sub-ontology.

Step 3) Definition of the General Framework of the Ontology

This step involves listing the main selected concepts (classes) and their formal interpretations.

Step 4) Definition of Topological Relations and Integration with Other Domains

In this step, the core of the ontology is presented. Classes and class hierarchies are explicitly defined, relationships between classes are established, and attributes and properties are identified according to the objectives. This step is crucial as the decision to classify a particular concept as a class or a single instance depends on the ontology's potential applications.

Step 5) Ontology Specification and Computation in the Editing Environment

The ontology is first modeled and then rendered as a 'script' in the OWL language using Protégé (Horridge, 2011). To ensure a correct and non-redundant ontology, its consistency is checked using an automatic consistency checker.

The spatial planning modeling problem is the result of a complex process involving many decision variables, defined here as the modeling domain. As a first step towards developing an ontology, it is necessary to define the different variables related to space. These domains should be extracted from existing ward layouts, and the ontology should optimize the spatial allocation problem. Consequently, the spatial domains and their connection to the building components included in a particular BIM play a crucial role.

Knowledge-intensive ontologies possess several defining characteristics that enhance their utility in practical applications(Bagchi, 2021). First, they are explicit, with clearly defined and precisely described concepts that make domain knowledge transparent and unambiguous. Second, they maintain a formal structure, utilizing machine-readable languages to represent concepts, attributes, and relationships in ways that enable computer processing and analysis. Third, these ontologies are inherently shared, designed to serve as common knowledge bases that facilitate collaboration and knowledge reuse across multiple users and systems within a specific domain. Finally, they emphasize applicability, focusing not on abstract theoretical constructs but on addressing real-world problems and meeting the concrete needs of practical application fields. This combination of explicit definitions, formal representation, shared accessibility, and practical orientation ensures that knowledge-intensive ontologies offer robust, adaptable solutions for domain-specific challenges while promoting efficient knowledge management and utilization.

The development of a knowledge-intensive ontology typically follows a structured five-stage process (Bagchi, 2021; Díaz-Agudo & González-Calero, 2007; Sun et al., 2010). First, domain analysis is conducted to thoroughly investigate and understand the specialized knowledge within a particular field, identifying its core concepts, entities, and their interrelationships. Based on these findings, the ontology design phase then establishes the structural framework, precisely defining concepts, attributes, and their various relationships. This theoretical design is subsequently implemented through formal ontology languages like OWL or DAML+OIL, transforming the conceptual model into a machine-readable knowledge base (Sarnikar & Deokar, 2010). Rigorous ontology evaluation follows to assess the system's completeness and logical consistency, verifying its accurate representation of the domain knowledge. Finally, the completed ontology is deployed in practical applications such as knowledge querying, reasoning systems, or visualization tools, demonstrating its real-world utility. This comprehensive approach ensures the resulting ontology is both theoretically sound and practically valuable for domain-specific problem-solving.

The development and application of knowledge-intensive ontology is of great significance for promoting the process of informatisation and improving the quality and efficiency of knowledge service. It is helpful to realize the deep excavation and utilization of knowledge, to promote knowledge innovation and to provide strong knowledge support for economic and social development.

The application of knowledge-intensive ontology is not only limited to theoretical research, but also has great flexibility and expansibility in practice. Here are some examples of the application of knowledge-intensive ontology: The Semantic

Web and the World Wide Web (Fensel, 2003): Knowledge-intensive ontology provides a theoretical basis for the Semantic Web, the Semantic Web is a network of nodes (resources) and relationships (semantic relationships) defined by ontologies. The Semantic Web Standards World Wide Web Consortium by the W3C (Miller & Swick, 2003), such as RDF (Resource Description Framework) and OWL (Web Ontology Language), are based on the principles of knowledge-intensive ontology. Artificial Intelligence: in the field of artificial intelligence, knowledge-intensive ontology is used to represent complex knowledge structures, supporting intelligent systems to reason and make decisions (Gulyaeva & Artemieva, 2019). For example, a knowledge base in an expert system is built based on a domain-specific ontology. Biomedical Informatics: in the field of biomedical informatics, ontology is used to unify terms and concepts used in biology and medicine, such as Gene Ontology (GO-RRB) (Gong et al., 2023) and disease ontology (DO) (Schriml et al., 2022). These ontologies help to integrate a large amount of biomedical data and promote the development of medical research. In Enterprise Information System (EIS), knowledge-intensive ontology can help to define and integrate all kinds of knowledge and information, improve the level of intelligence of EIS, and support the decision-making process of EIS (Tabatabaie et al., 2011). The application of knowledge-intensive ontology in digital libraries and museums can help to organize and retrieve a large number of cultural resources and historical materials, providing richer and more accurate information services (Chi et al., 2006).

The construction and application of knowledge-intensive ontology is a developing process, which requires close cooperation among domain experts, knowledge engineers and computer scientists. With the development of technology and the expansion of domain knowledge, knowledge-intensive ontology will continue to provide strong knowledge support and innovation impetus for every domain.

## 4.4 Site Selection

The traditional location model usually combines qualitative description with mathematical planning, and lacks comprehensive consideration of multiple complex factors such as GIS, urban planning interface and traffic convenience (Carrasco et al., 2022). In this study, based on the case of renovation of an existing building in the Nightingale Hospital (ExCel London) in the UK, which was demolished after completing its mission, the proposed renovation site was selected (Wise, 2021).

As an important consideration of hospital location, transportation convenience affects the convenience and first aid efficiency of patients and medical staff. The space on the ground floor of one of the research institution's buildings was selected for this study on the assumption that it would be converted into a temporary epidemiological resilience hospital, and was chosen according to the following conditions: openness of the space; large volumes of doorways, corridors, etc., which would facilitate the subsequent transport of equipment and materials, as well as the access of vehicles for some construction equipment; availability of access roads connecting to the automobile traffic; and the possibility of dismantling the already existing furnishings.

Considering that in reality, many completed buildings may only have traditionally designed paper 2D drawings, or it is difficult to obtain drawings and other relevant building information due to the fact that they were built too long ago, this study uses 3D maps extracted from on-site cameras. Through the use of an RGB-D camera mounted on a mobile Turtlebot robot, the 3D map of the selected space is acquired. Traditional methods relying on paper-based 2D drawings or outdated digital records often prove inadequate, especially for older buildings where original documentation may be lost or inaccessible. The RGB-D camera's ability to capture both color and depth information in real-time provides a comprehensive solution, generating accurate three-dimensional data of physical spaces without requiring pre-existing technical drawings. The choice of a mobile robotic platform offers distinct advantages for spatial data acquisition. The Turtlebot's mobility enables efficient scanning of large or complex areas that would be time-consuming and labor-intensive to measure manually. During the modeling process, the methodology intentionally simplifies certain elements by blurring non-critical details like decorative features and exact ceiling heights. This selective abstraction serves two important purposes: it focuses the analysis on essential spatial relationships needed for ontology development, and it streamlines the modeling workflow by eliminating unnecessary complexity. The resulting models prioritize functional and structural information over aesthetic or temporary features, making them particularly suitable for studying spatial configurations in potential cabin hospital conversions. Compared to professional laser scanning equipment, the RGB-D and mobile robot solution provides an affordable yet effective alternative that can be deployed in various building types and conditions. The method's flexibility makes it particularly valuable for emergency scenarios where rapid spatial assessment is required, such as evaluating buildings for potential conversion to medical facilities during public health crises. The RGB-D sensor's combination of visual and depth data allows for precise reconstruction of spatial geometries, which forms the foundation for creating accurate digital models in Revit. The site space was drawn using Revit based on this 3D map, and during the drawing process, the external conditions as well as the spatial layer heights were blurred, which helped to circumvent other lengthy information that was not important. The generated 3D spatial data serves as a neutral, updatable digital baseline that supports ongoing ontology development and refinement. By establishing this objective foundation, the research can more effectively analyze spatial relationships, test design configurations, and develop rules for optimal healthcare facility layouts. This approach not only addresses current documentation challenges but also creates a framework that can adapt to future requirements in architectural analysis and adaptive reuse projects.

## 4.5 Ward unit

The various professional drawings of Huoshenshan Hospital are open to access, and the 3D models of partial hospital wards are drawn through Revit based on the 2D drawings. Based on the IFC file of single Huoshenshan cabin hospital ward unit, the IFCOWL ontology was built by the IFC-OWL tool, restricting layout of the existing space through this ontology. Building Topology Ontology (BOT) has been referred to as the minimal OWL DL ontology for representing topology relationship between entities. As an extensible baseline, BOT's structure is recommended for use with more domain specific ontologies.

In the process of ontology-based hospital layout design, the construction of functional units is the core of the overall design. The function unit is defined by matching the medical activities with the physical space. It not only carries the basic service functions of the Huoshenshan hospital, it is also an important part to realize the demand of elastic space. In this research, the function unit is designed to concretize the function, flexibility and expansibility of the hospital space, and to ensure that the cabin hospital layout design can be realized at the same time, can also meet the improvement of the quality of medical services.

In the functional unit construction, the first consideration is the standardization of the clinic layout, which includes the determination of the optimal size range, spatial layout and necessary medical facilities. Based on the analysis of medical process, the elements of clinic layout are extracted by data acquisition, and then these elements are incorporated into the BIM model by using parameterized design software, to form a reusable, flexible adjustment of the office unit template.

Using this ontology-based approach, the research integrates information from different fields, including architectural design standards, existing hospital ontologies, in order to support the whole hospital layout in the operation of all-round optimization.

In summary, the construction of functional units in this study utilises automatic layout to shorten the design time of a general cabin hospital on the premise of combining the spatial layout of cabin hospitals in different countries, sustainability and flexibility of cabin hospital design.

## 4.6 Integration ontology

Integrating two ontologies can be a complex task, requiring the adaptation, merging and harmonisation of different conceptual frameworks. Their main approach usually consists of two main steps: Firstly, the initial step involves reconciling differences by identifying semantic correspondences (primarily similarities) between the various elements. This is achieved through similarity calculations, which help determine how different elements relate to each other. Secondly, integration step, in this step, the results from the mapping phase are utilized by merging or linking the matched elements. This process generates a new unified view, creating a cohesive and integrated ontology that harmonizes the previously disparate elements. By following these steps, it is possible to create a comprehensive and unified ontology that effectively integrates different data sources and domains, facilitating better information sharing and collaboration. This integrated approach ensures that all relevant elements are accurately represented and connected, providing a robust framework for knowledge management and application across various fields.

In order to understand the ontologies, the structure, concepts and relationships of each ontology are studied and understanding the domains they cover and the purposes they serve is primary. This allows the identification of overlapping and unique domains, determining which parts of the ontology overlap and which parts are unique to each ontology. This helps to map the corresponding concepts and relationships, which need to be done manually. Add the appropriate spatial constraints to this process to map the relationships and attributes between concepts in each ontology, ensuring that the semantics are preserved during the mapping process, thereby facilitating accurate data integration and interoperability across different systems. At the same time, due to the different origins of the two ontologies, a conflict master of semantic structure is inevitable. To resolve differences in the meaning and scope of concepts, new, broader concepts need to be created to cover these differences. This process is done through the ontology editor Protégé.

In this study, a new IFC-Space ontology is proposed by integrating ontology. As a formal logical system, ontology is used to represent the concepts and relations of domain knowledge. In this study, a complete set of knowledge ontology of hospital spatial layout is constructed, which combines the professional knowledge of hospital design field with the spatial requirements, and allows for in-depth logical reasoning, to achieve a highly personalized and dynamically adjusted layout design.

The first challenge is to build a central ontology that integrates all relevant concepts and associations in hospital design. The central ontology contains knowledge rules such as diagnosis and treatment process, patient flow line, medical equipment, safety, regulation and so on. The ontology integrates knowledge from different disciplines, including architecture, medicine, management, artificial intelligence, and geographic information systems. Through cross-domain information integration, the centre ontology can provide a global perspective and in-depth insights for hospital layout design.

In order to develop an effective rule base, a logic programming method is used in this study. These rules are based not only on design principles and building standards, but also on the best practices in medical services. The establishment of the rule base allows the design system to automatically detect the inconsistency and potential risks in the scheme, and provides theoretical support for design decision-making. In addition, the rule base adopts probability model to deal with the inherent uncertainty in the knowledge ontology, and then optimizes the layout design.

Ontologies are built on international standards, such as the Web Ontology Language (OWL) and the Resource Description Framework (RDF) , to ensure the interactivity and compatibility of knowledge. In addition, the ontology integrates multiple hospital information systems, building information model (BIM) , and other related databases. This enables ontologies to handle data in a variety of formats and sources, and to address the challenges of rapidly evolving medical technologies.

In the process of verifying ontology, a series of strict logic and empirical methods are adopted. Ontology and rule base are checked formally to ensure their logical correctness and consistency. Then, through the use case test, we test the applicability of the knowledge ontology and layout algorithm for the specific design task. At the same time, by integrating with BIM software, the application value of Ontology and rule base in practical engineering is further verified.

Finally, this study successfully defines a new category of hospital spatial layout design, and provides a design method that can be extended and adapt to future changes. This kind of ontology-based design method is expected to become the standard practice of cabin hospital layout design, and has a far-reaching impact on the whole medical facility industry. By integrating and applying interdisciplinary knowledge, this study not only improves the effectiveness and accuracy of the design, but also enhances the flexibility and adaptability of the hospital layout.

## 4.7 Validation ontology

In the research of ontology-based cabin hospital layout design, the verification of ontology is the key to achieve the design goal and ensure the close integration of theory and practice. In Chapter 2, several popular methods for building ontologies, validation has been mentioned several times, it is essential in ontology development to ensure both logical correctness and practical applicability (Noy, Natalya F. & McGuinness, 2001; Uschold, Michael & King, 1995; Uschold, Mike & Gruninger, 1996). In order to verify the correctness and applicability of ontology, it is very important to adopt multi-angle and multi-level verification method . In this study, the verification methods include formal verification and query testing. The adoption of both formal verification and query testing as validation approaches for the ontology-based cabin hospital layout design serves complementary but distinct purposes in ensuring the ontology's reliability and practical utility.

Formal verification provides a mathematical and logical foundation for the ontology's correctness. By using tools like the HermiT reasoning engine and RDF validators, this method rigorously examines the ontology's internal consistency, ensuring that its conceptual hierarchy, attributes, and relationship constraints are free from contradictions and redundancies. This step is crucial because it guarantees that the ontology's structure adheres to sound logical principles before practical application. The formal approach identifies implicit errors that might not be obvious during development, such as conflicting inheritance rules or unsatisfiable class definitions. For instance, it can detect whether a "Contaminated Zone" improperly inherits properties from a "Clean Zone," which would violate infection control principles. By resolving these issues early, formal verification enhances the ontology's robustness and ensures its spatial rules and constraints are theoretically valid.

Query testing, on the other hand, evaluates the ontology's practical applicability by simulating real-world usage scenarios. Competency Questions (CQs)—such as "Which wards are reachable from BufferZone1?" or "What medical equipment is contained in a Ward?"—test whether the ontology can deliver accurate, actionable answers to domain-relevant queries. This method bridges the gap between theoretical correctness and functional performance. For example, if the ontology fails to return the dimensions of a bathroom or the sinks in a ward, it reveals gaps in knowledge representation or missing relationships. By employing query languages like SPARQL and OWL-DL, this process mimics how end-users (e.g., architects or healthcare planners) would interact with the ontology, ensuring it meets their needs. The iterative nature of query testing allows for continuous refinement, aligning the ontology with evolving design requirements.

Together, these methods form a comprehensive validation framework: Formal verification ensures the ontology is internally consistent (no logical flaws), while Query testing confirms it is externally valid (useful for real-world tasks).

This dual approach is especially critical for cabin hospital design, where errors in spatial logic (e.g., incorrect zoning) or functional gaps (e.g., missing equipment lists) could compromise infection control or operational efficiency. The synergy between the two methods not only validates the current ontology but also extends its adaptability for future refinements and applications in intelligent healthcare design.These methods can fully verify the applicability of ontology, and to ensure that the ontology design method can effectively guide the actual hospital layout design.

### 4.7.1 Formal validation

The formal methods uses logical reasoning and mathematical proof to verify the consistency and completeness of the ontology. In this study, formal verification tools such as RDF Online Validator  and Protégé's built-in HermiT reasoning engine, were used, logical consistency and syntax checking of concept hierarchy, category attribute and relation constraint in ontology. This reasoning engine can identify contradictions between concepts, deduce implied relationships, and automatically detect possible logical errors. The RDF Online Validator checks syntax compliance with RDF/OWL standards, verifying that the ontology's framework adheres to formal language specifications and eliminating formatting errors that could disrupt interoperability with other semantic tools. Meanwhile, HermiT, as a Description Logic (DL) reasoner, performs deeper logical validation by analyzing concept hierarchies, property relationships, and constraints—detecting contradictions (e.g., conflicting class definitions), inferring implicit knowledge, and confirming. Together, these tools provide layered validation: RDF Validator ensures technical correctness in encoding, while HermiT guarantees semantic soundness, enabling the ontology to function reliably in reasoning tasks. This combination is especially vital for cabin hospital design, where flawed spatial logic or inconsistent rules could compromise infection control or operational efficiency. By leveraging both tools, the ontology achieves robustness in both form (syntax) and function (logic), forming a trustworthy foundation for layout optimization.

In formal verification, using automatic reasoning tool such as HermiT to check the consistency and completeness of ontology. This tool can find hidden knowledge, incompatibilities between concepts and potential logical conflicts in ontology, and provide information for subsequent optimization.

The formal verification results of ontology highlight the following advantages: firstly, the spatial layout knowledge expressed by ontology is supported by an effective logical framework, which ensures the rationality of planning rules and the feasibility of implementation.

Secondly, the formal verification results show that the proposed method achieves the desired goals in terms of logical consistency, accuracy of knowledge representation and feasibility of application. Furthermore, the validation process provides reliable theoretical and technical support for the further research and optimization of cabin hospital layout design, and further promotes the trend of intelligent and personalized hospital design. Through formal and strict verification, the applicability of ontology in practice has been confirmed, and laid a solid foundation for future related research and practical application.

## 4.7.2 Query validation

Use query testing is an effective way to verify the applicability of ontology. It selects typical Competency Questions (CQs) to test the structural consistency of the ontology and the logical relationships between entities (Bezerra et al., 2013). Competency Questions (CQs) are formulated as natural language queries that the ontology must accurately address, serving as practical benchmarks for its effectiveness(Noy, Natalya Fridman & Hafner, 1997). While these questions and their expected answers are often overlooked during initial ontology specification, they play a crucial role in validation by testing whether the ontology can correctly resolve real-world scenarios. Grüninger and Fox's work was the first time was the first to introduce the idea of describing competency questions (CQs) using both axioms and natural language text(Fox & Grüninger, 1994). Subsequently, other methodologies (Fernández-López et al., 1997a; Haase et al., 2008; Uschold, Mike & Gruninger, 1996) have also proposed the use of competency questions (CQs). The ability to properly answer CQs demonstrates that the ontology adequately captures and represents domain knowledge, ensuring it meets actual user needs and application requirements. This validation step will bridge the gap between theoretical ontology development and practical implementation, confirming the ontology's utility for decision-making in specific contexts like cabin hospital design. Query checking will be used repeatedly throughout the ontology development process, ensuring the ontology remains aligned with its intended goals and scope. This iterative validation approach will maintain the ontology's relevance and accuracy as it evolves, while demonstrating its practical value for real-world healthcare facility planning challenges. The continuous query testing will serve as an ongoing quality control mechanism, guaranteeing that the ontology consistently meets both technical requirements and user needs throughout its lifecycle.

Query testing is a key step to verify the validity of IFC-Space ontology. It aims to test the structural consistency of the ontology and the logical relationships between entities through CQs, and provide the basis for the follow-up layout design optimization. In the present study, validation employed a detailed questions set that integrates information from ontology. Query testing involves evaluating whether the ontology can effectively answer a predefined set of competency questions. These questions represent the key queries that users need to make within the domain covered by the ontology. The process begins with the formulation of these competency questions during the initial stages of ontology development. In the middle of ontology, query detection can be used repeatedly. Once the ontology is constructed, query testing is performed by running these competency questions against the ontology to check if it can provide accurate and complete answers. Table- is a list of some CQs example.

Table 4-2 Competency Questions list

| |
|---|
| **CQ 1 : Which individuals are instances of Facility?** |
| **CQ 2 : What are the dimensions of Bathroom1?** |
| **CQ 3 : Which wards are reachable from BufferZone1?** |
| **CQ4: where all the sink belong to?** |
| **CQ5 : What medical equipment contained in Ward?** |

During query testing, the first step is to define a formal specification language, among which OWL-DL and SPARQL are the most widely used choice, to execute these questions on the ontology.

OWL-DL (Web Ontology Language Description Logic) is a sublanguage of the OWL specifically designed to provide maximum expressiveness while retaining computational completeness and decidability (Motik et al., 2005). It is based on Description Logics (DL), which are formal knowledge representation languages. OWL-DL allows for complex expressions to describe relationships between concepts, including features like cardinality constraints, enumerations, and property restrictions. It ensures that all logical conclusions derivable from the defined ontology are computable, meaning reasoning engines can determine all entailments from the ontology. Additionally, OWL-DL guarantees that all reasoning tasks will complete in finite time, making it possible to ensure the performance of reasoning engines.

SPARQL (SPARQL Protocol and RDF Query Language) is a powerful query language and protocol for accessing and manipulating data stored in Resource Description Framework (RDF) format(Hogan & Hogan, 2020). RDF is a standard model for data interchange on the web, representing information about resources in a graph form. SPARQL is used to query, retrieve, and manipulate this RDF data.

If the ontology returns the expected results, it indicates that the ontology's structure, relationships, and data representation are correctly modelled. However, if the results are incorrect or incomplete, it highlights areas where the ontology needs refinement. This iterative process of query testing and subsequent refinement ensures that the ontology not only captures the domain knowledge accurately but also meets the practical requirements of its intended users.

Query testing also helps in identifying logical inconsistencies, missing relationships, and gaps in the ontology's coverage. By continuously testing and validating the ontology against real-world queries, developers can ensure that the ontology remains robust, reliable, and fit for purpose. This process is crucial for maintaining the ontology's quality and effectiveness, making it a valuable tool for knowledge representation and decision-making within its specific domain.

In summary, through the comprehensive application of the above methods, the ontology-based cabin hospital layout design has been strictly verified. The formal methods ensures the logical correctness and rationale of the ontology, and the use query test verifies the. The results show that the IFC-Space ontology can effectively direct the layout design of epidemic resilient cabin hospitals, ensure the rational use of hospital space and rapidly adapt to changing needs, it lays a solid foundation for the research and practice of hospital layout design in the future.

IFC-Space ontology shows strong adaptability and forward-looking in validation. The test not only verifies the rationality of the methodology, but also provides an extensible technical scheme and theoretical support for the intelligent and fine management of the hospital layout ontology in the future.

## 4.8 Ward layout design

The COVID-19 pandemic has underscored the urgent need for adaptable and efficient healthcare infrastructure, particularly in the design of temporary cabin wards to accommodate the surge in patient numbers. Constraint Satisfaction Problem (CSP) techniques are widely used in spatial layout design, particularly for optimizing complex arrangements where multiple interdependent requirements must be satisfied(Baykan & Fox, 1991; Zawidzki et al., 2011). Automated design techniques utilizing Constraint Satisfaction Problem (CSP) methods provide a sophisticated solution to rapidly generate optimal layouts for these temporary wards. CSP methods enable the systematic exploration of design configurations by defining and solving a set of constraints tailored to the specific requirements of COVID-19 cabin wards.

The application of CSP methods in space layout design offers several distinct advantages that substantially enhance the design process. CSP methods provide a systematic framework for exploring various design alternatives. By defining variables, domains, and constraints, designers can automatically generate and evaluate multiple layout configurations that adhere to the specified constraints. Space layout design often encompasses a multitude of complex constraints, including spatial relationships, accessibility, safety regulations, and aesthetic considerations. CSP methods effectively manage these intricate constraints through techniques such as constraint propagation and backtracking, ensuring that all constraints are satisfied in the final design.

The declarative nature of CSP enables designers to easily specify and modify constraints without altering the underlying solution algorithm. This flexibility is particularly valuable in space layout design, where requirements and constraints may evolve throughout the design process. CSP methods automate repetitive and time-consuming design tasks, such as positioning elements within a space while ensuring compliance with constraints. This automation not only saves time but also reduces the potential for human error, resulting in more accurate and consistent designs.

Furthermore, CSP can be combined with optimization techniques to identify the optimal layout according to specific criteria, such as minimizing unused space, maximizing accessibility, or optimizing the flow of movement within the space. This capability ensures that the final design is not only feasible but also optimal in terms of the desired objectives. CSP-based tools facilitate an interactive design process, allowing designers to iteratively refine constraints and explore different layout options in real-time. This interaction fosters creativity and allows for rapid adjustments based on feedback and new requirements.

CSP methods are scalable, capable of handling large and complex spaces, making them suitable for a wide range of design projects, from small rooms to large commercial buildings or urban planning. Advanced CSP algorithms can efficiently manage the increased complexity associated with larger problems. Additionally, CSP can be integrated with other design tools and technologies, such as CAD software and Building Information Modeling (BIM). This integration allows for seamless data exchange and enhances the overall design workflow.

Finally, CSP methods can detect conflicts between constraints early in the design process, enabling designers to address and resolve these issues before they become problematic. Early conflict detection leads to more robust and viable designs, ultimately improving the quality and feasibility of the final layout.

In designing COVID-19 cabin wards, CSP methods involve defining variables representing various elements of the layout, such as the size and location of patient cabins, and the placement of medical equipment. Constraints are then applied to ensure the design meets critical criteria, such as maintaining adequate spacing between cabins to minimize infection risk, ensuring clear pathways for staff movement, and providing sufficient ventilation and isolation measures to control the spread of the virus.

One significant advantage of CSP-based automated design for COVID-19 cabin wards is its ability to handle and reconcile multiple constraints simultaneously. For example, CSP algorithms can ensure that each cabin has direct access to medical facilities while also maintaining isolation protocols. Additionally, these methods can incorporate regulatory requirements and best practices for infection control, ensuring that the resulting designs are both functional and compliant with health guidelines.

The use of CSP methods allows for the rapid generation and evaluation of numerous layout configurations, significantly accelerating the design process compared to traditional methods. This speed is crucial in pandemic situations where time is of the essence. Automated CSP tools can quickly identify the most effective layout that maximizes patient capacity while ensuring safety and operational efficiency. Furthermore, these tools can be integrated with Building Information Modeling (BIM) systems, enabling real-time updates and adjustments as new information or requirements emerge.

The integration of CSP methods in the design of COVID-19 cabin wards also allows for flexibility and scalability. As the pandemic situation evolves, the ability to quickly reconfigure and adapt the ward layout to changing needs becomes invaluable. This adaptability ensures that healthcare facilities can respond promptly to fluctuations in patient numbers and changes in treatment protocols, thereby maintaining high standards of care.

In summary, the application of CSP methods to the automated design of COVID-19 cabin wards provides a robust framework for addressing the unique challenges posed by the pandemic. By automating the design process and ensuring adherence to critical health and safety constraints, CSP methods contribute to the creation of efficient, adaptable, and compliant healthcare environments that can effectively support patient care during public health emergencies.

In this study, we will extract the key data from the historical hospital layout planning cases and input them into the ontology system as test cases to see if the system can produce a reasonable layout plan, and compare with the results obtained by traditional methods, to verify the adaptability and flexibility of ontology under unconventional requirements. Use case testing can not only verify the functionality of ontology, but also verify its usability and effectiveness from the user's point of view.

# Chapter 5 Ontology Development

The development of a spatial ontology for cabin hospital layouts is a critical task that aims to enhance the efficiency, safety, and adaptability of hospital spaces. Spatial ontology provides a formal framework to define and manage the spatial characteristics and relationships of various entities within a hospital environment. This chapter delves into the intricacies of creating a comprehensive spatial ontology using the Web Ontology Language (OWL), focusing on the unique requirements and constraints of cabin hospital.

The primary challenge in developing a spatial ontology lies in accurately defining the default attributes and spatial relationships that are essential for effective workspace management. This includes the physical dimensions, orientation, and location of entities, as well as the structural organization of space. By capturing these elements, the ontology facilitates a better understanding of the spatial dynamics within a hospital, which is crucial for optimizing workflow, ensuring patient safety, and enhancing the overall efficiency of medical operations.

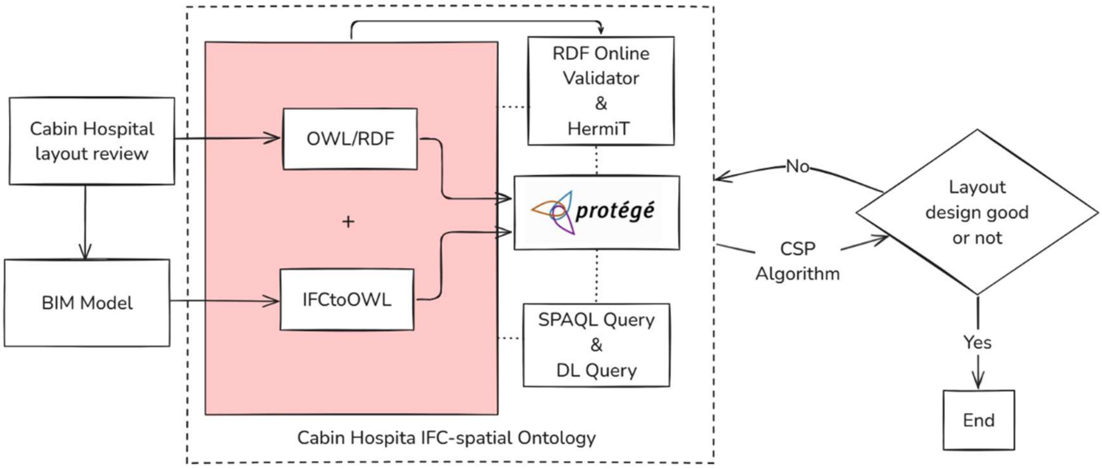



Figure 5.1 Cabin hospital space ontology development process

The spatial ontology is developed using the OWL language, allowing for precise formal descriptions of concepts and their relationships. OWL properties, which include object properties and datatype properties, play a pivotal role in defining the binary relations between classes and individuals. This chapter explores the various attributes of OWL properties, such as functionality, inversion, transitivity, and symmetry, and their implications for modelling spatial relationships.

Furthermore, this chapter discusses the integration of the proposed spatial ontology with BIM data, leveraging the Industry Foundation Classes (IFC) standard to enhance the interoperability and flexibility of hospital layout designs. By utilizing tools like Protégé for ontology editing and visualizations, the study demonstrates how a well-structured spatial ontology can support dynamic and intelligent hospital layout planning. Figure 5.1 outlines an integrated workflow for evaluating and optimizing cabin hospital layouts by combining Building Information Modeling (BIM), semantic web technologies, and computational optimization methods. The process begins with a BIM model of the facility, which is then converted into a semantic format using IFC-to-OWL/RDF transformation tools to create a machine-readable ontology representation. This ontology undergoes rigorous formal validation through the RDF Online Validator for syntactic correctness and Protégé's HermiT reasoner for logical consistency checking, ensuring the spatial relationships and constraints are properly defined. The refined ontology is then tested using SPARQL and DL queries to verify its ability to answer competency questions about the layout's functionality and compliance with healthcare requirements. The system evaluates whether the current layout meets all criteria, and if deficiencies are found, a Constraint Satisfaction Problem (CSP) algorithm is applied to generate optimized layouts that satisfy all spatial, functional, and regulatory constraints. This iterative process continues until the layout passes all validation checks, resulting in a design that balances infection control protocols, workflow efficiency, and spatial utilization. The workflow demonstrates how the integration of BIM data, formal ontologies, and computational optimization can systematically improve healthcare facility design, particularly for rapid deployment units like cabin hospitals where strict adherence to spatial constraints is critical for operational effectiveness and patient safety.

## 5.1 Space Ontology

The primary challenge in creating a spatial ontology for a cabin hospital is to define suitable default attributes to support and manage several key aspects:

(1) Spatial Physical Entities: Establish default spatial data such as dimensions, orientation, and location.

(2) Spatial Structure: Describe the organization of space within the site environment to define the spatial data structure, which is crucial for detailing the spatial relationships between entities based on their geometric locations.

This spatial classification is computerized within the OWL (Web Ontology Language) ontology editing environment. OWL classes are interpreted as collections of user-defined 'individuals', described using formal specifications that precisely define the membership criteria for each class.

The goal is to build an OWL spatial ontology, serving as a formal description of the concepts (OWL classes) that model the spatial layout and relationships between different spaces. Each concept in the ontology is detailed through various relationships with other concepts or attributes (OWL attributes) and restrictions on these attributes (OWL restrictions). These attributes accurately define the membership requirements for the class.

Specifically, "OWL properties" are binary relations on classes, categorized into two main types: object properties (relations between two classes or individuals) and data type properties (relations between individuals and data type values such as real numbers, decimal numbers, strings, Booleans, time instances, etc.). In essence, they link an individual to a specific data value.

Moreover, OWL enhances attribute meanings through attribute properties, such as functionality (FU), inversion (IN), transitivity (TR), symmetry (SY), asymmetric inversion (AS), and invertibility (IR). The key types of object attributes (relations) and their specifications include:

(1) Functional attribute ensures that for a given individual, at most one other individual can be related to it through that attribute.

(2) Inverse functional attribute means its inverse attribute is functional; thus, for a given individual, at most one other individual can be related through that property.

(3) Transitive property implies that if it relates individual a to b and b to c, then a is also related to c through this property.

(4) Symmetric property means if it relates individual a to b, then b is also related to a through the same property.

Classes are organized hierarchically into super-classes and subclasses, where subclasses specialize in ("are subsumed by") their superclass. For instance, the class 'room' could be divided into therapeutic and non-therapeutic rooms or corridors. The ontology focuses strictly on the functional division of space.

Visualizing ontologies can aid in their development, exploration, and validation. Various ontology visualizations have been developed, often used within ontology editor environments like Protégé.

### 5.1.1 Definition and classification of spatial concept

In the ontology-based cabin hospital layout design method, the definition and classification of spatial concept is the foundation of constructing spatial ontology, and the key is to define the attributes and functions of different hospital spatial types, the framework of spatial attribute relationship and parameter configuration is established. In the design of cabin hospital layout, the space can be roughly divided into clinical space and service logistics area. Each type of space has its unique functional requirements and design constraints, must play a role in the overall operation of the hospital and treatment effect. In this study only selected clinical space, which are wards, supporting ancillary facilities such as bathrooms toilets and corridors.

Clinical space is the core of the hospital, including the ward, corridors and other places. The design of clinical space should focus on the safety and comfort of patients, improve medical efficiency and reduce the risk of cross-infection as much as possible. For example, by effectively separating the clinical space from the public area, not only patients' privacy can be guaranteed, but also patients' psychological stress can be relieved, while avoiding the random entry of outsiders. In addition, the layout of the clinical space needs to consider future developments to ensure that there is sufficient flexibility to accommodate additional devices or services.

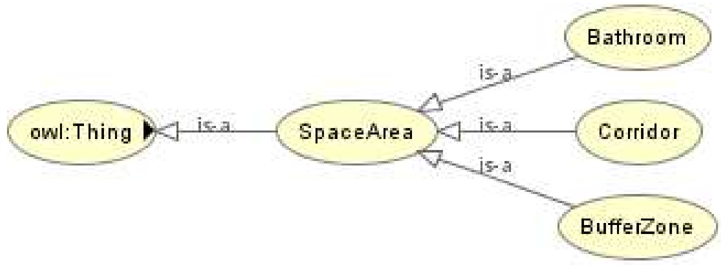



Figure 5.2 Space area class Hierarchy illustration

For spatial area, there are three classes, which is Ward, Buffer zone and Bathroom, Figure 5. 2 depicts a simple hierarchical ontology structure represented in a flowchart format, illustrating basic class relationships within a spatial classification system. The structure begins with the broadest category "out:Thing" (referring to the top-level class "Thing" from a standard ontology), which serves as the parent class for all other entities. Below this, the "SpaceArea" class appears as a direct subclass, representing a general spatial division or zone within a facility. The most specific level shows "BufeZone", which inherits from "SpaceArea" and would represent a particular type of spatial area with specialized functions, such as the transitional zones critical in healthcare facilities for infection control between clean and contaminated areas. This three-tiered hierarchy demonstrates how ontology engineering organizes domain knowledge from general to specific concepts, enabling precise categorization of spatial elements that could be applied to architectural planning, particularly in contexts like cabin hospital design where clear zoning definitions are essential for operational safety and efficiency. The simplicity of this structure suggests it may be part of a larger ontology framework where these classes would be further elaborated with properties, constraints, and relationships to other spatial elements., the definition of them shows in Table 5-1.

Table 5-1 Class Definition

| Class name | Definition |
|---|---|
| **Ward** | the space for operation treatment for patient, according the case of Huoshenshan hospital, there are two type of wards, |

| | |
|---|---|
| | they are mirrored. |
| **Buffer zone** | In case of covid-19, in order to control the virus spreading, medical staff need to disinfect before entering and after leaving the ward, this is function of buffer zone. |
| **Bathroom** | Additional facilities in the ward |
| **Corridor** | A space for people to pass through, connecting each treatment unit |

In addition to the space type, a minimum treatment unit also contains some general facilities, the class detail is shown int Table 5-2.

Table 5-2 Class of General Facility

| **Class name** | |
|---|---|
| **General Facility** | |
| **General Facility has sub class** | Light, Bed, Shower, Toilet, Socket |

According to the instructions for China Cabin design manual(MA Longxin et al., 2022) and a report of pandemic resilient hospital(HKS Architects & Arup, 2021), a well functional cabin hospital also contains several medical equipment.

Table 5-3 Medical Equipment

| **Class name** | |
|---|---|
| **Medical equipment** | Equipment for medical Operation and safety |
| **Has subclass** | |
| **Negative-pressure ventilation** | The ward is under negative pressure to prevent the spread of the virus |
| **Medical contaminant collecting bin** | to collect masks, gloves, etc. that are replaced in the buffer zone after medical staff complete the operation. |
| **Ventilator** | Important equipment to maintain vital signs of critically ill COVID-19 patients |
| **Calling Device** | To call other medical stuff, in ward also in bathroom for patient need help |
| **Medical monitor** | Used by healthcare professionals to measure the health |

| | |
|---|---|
| | status of patients |
| **UV germicidal lamp** | In Buffer zone, for virus spread control |

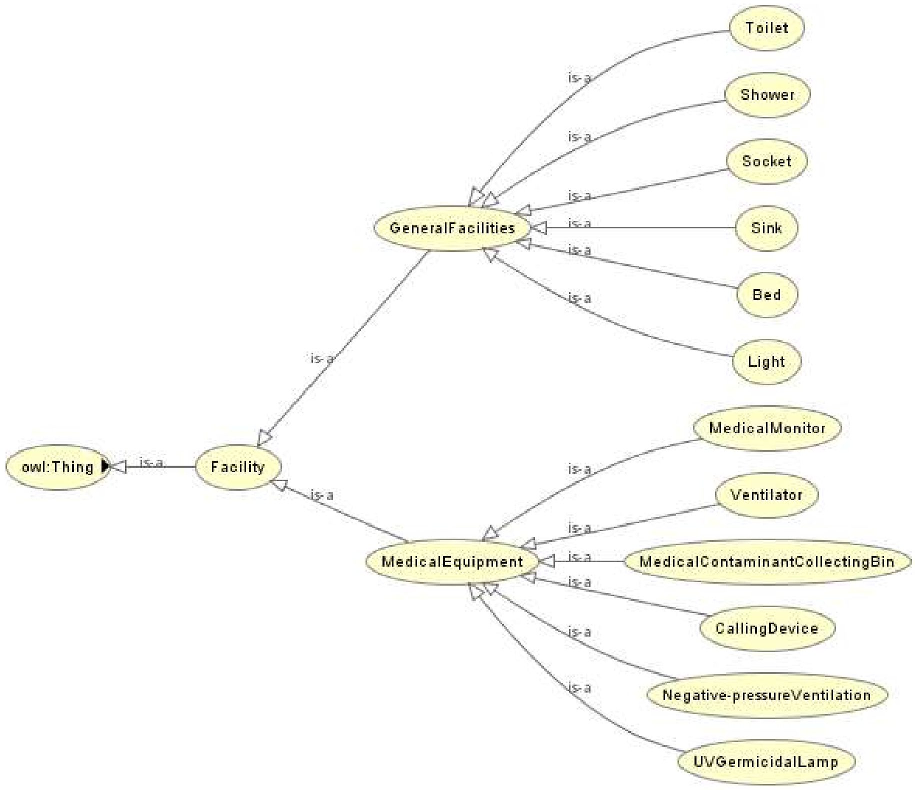



Figure 5.3 Facility Class Hierarchy

Figure 5.3 presents a hierarchical classification system for healthcare facility components, organized into functional categories. At the highest level, it appears to group elements, Facility into 2 primary classes: General Facilities and Medical Equipment. Under general facilities, basic amenities like Toilet, Shower, socket, Bed, and Light are listed, representing essential patient support infrastructure. The Medical Equipment branch includes critical medical equipment such as collecting bin, negative-pressure ventilation and UV lamp, emphasizing environmental safety measures, like virus spread control. The ventilator and Monitor are highlighting life-support systems. The structure effectively demonstrates how ontology engineering classifies healthcare components from general facilities to specialized medical and contamination-control systems. This taxonomy could serve as part of a larger ontology for cabin hospital design, where clear categorization of equipment and zones is crucial for infection prevention and operational efficiency. The inclusion of both clinical

(Ventilator) and environmental safety (UV Lamp) elements reflects the dual priorities of patient care and pathogen containment in epidemic-resilient facilities.

To sum up, the definition and classification of the concept of space plays an important role in the process of establishing ontology. Each type of space needs not only to meet current functional needs, but also to take into account possible changes in the future, such as increased human traffic and the introduction of new equipment as a result of advances in science and technology. Therefore, the ontology model of hospital space should be flexible to adapt to the changing medical environment and social needs.

### 5.1.2 Spatial attributes and their representation in ontology

In the field of cabin hospital layout design, spatial attributes and their representation in ontology are indispensable components in the process of optimal design. This study reviews the layout of previous cabin hospitals, adopts a mixed layout model, which is use the layout model of Wuhan cabin hospital, and accepts the transformation of existing buildings by Nightingale Hospital to analyze the properties and expressions of spatial relationships.

The analysis of spatial attributes covers partial aspects of the hospital environment, including but not limited to spatial size, layout efficiency, etc., form a multi-dimensional parametric evaluation. For example, in terms of space size, a series of space size parameters are formed by taking into account clinical requirements, walking lines and equipment. The facilities contained in the space also determine the function of the space to a certain extent.

During the construction of space ontology, Web Ontology Language is used to define the spatial attributes, which ensures the clear and consistent expression of spatial attributes in ontology (Fensel, 2003; Hitzler et al., 2020). As Fensel and Hitzler’s work emphasized, the adoption of Web Ontology Language (OWL) for constructing space ontologies is driven by its unique capabilities to formally and unambiguously define spatial concepts and relationships. First, OWL provides a standardized, machine-interpretable framework to precisely define spatial concepts—such as "Ward" or "BufferZone"—along with their attributes (e.g., dimensions, adjacency rules) and relationships (e.g., "isAdjacentTo" ). This eliminates ambiguity in spatial knowledge representation, ensuring consistency across applications. Additionally, OWL's foundation in description logic enables automated reasoning about spatial configurations, allowing it to detect contradictions or deduce implicit relationships (e.g., transit paths between modules), which is crucial for validating healthcare layouts. Furthermore, OWL's support for modular ontology development allows spatial ontologies to scale and adapt, accommodating new constraints—such as pandemic-specific zoning rules—without requiring a complete model restructuring. OWL's expressiveness, computational tractability, and standardization make it indispensable for creating robust spatial ontologies that are both theoretically sound and practically actionable in architecture and healthcare design.

The implementation of multi-directional attribute analysis is based on the ontology framework of concept and instance. In the design of space, first of all, according to the functional needs of the hospital space is divided into ward areas, bathroom areas, corridor areas. Each area has not only the basic physical attributes, but also the related use function. In this way, the multi-information of spatial attributes is integrated into a unified ontology model, which provides powerful data support and logical reasoning ability for further spatial layout optimization.

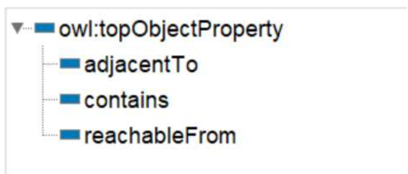



Figure 5.4 Object Property

There are three Object properties that express spatial location relationships in the ontology, as shown in Figure 5.4.

(1) "adiacentTo" refers to the neighbourhood between spaces.

(2) "contains" means attribution relationships, e.g. what equipment is contained in the space.

(3) “reachableFrom” refers to reachability, it shows whether spatial entities are connected to each other. For example, in order to realise the hospital space about limiting the pathway of virus transmission, medical staff can only enter the ward through the buffer zone, then the relationship between the ward buffer zone and the corridor is that the corridor is accessible to the buffer zone, and the buffer zone is accessible to the ward, but the ward is not directly accessible from the corridor.

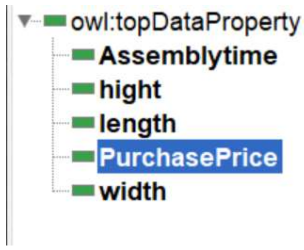



Figure 5.5 Data properties in space ontology

The space ontology also defined five Data properties, shows in Figure 5.5, these enrich the information of the instances in the ontology. Assembly time means the time required for one person to install this entity. Hight, Length and  are the data to show the size of the entity. Purchase price is money cost, which is a factor to consider budget control. These data properties play an important role in space size, later construction time and cost control.

Through the induction and standardization of ontology, the space attribute in the design process can be systematized and handled automatically. This study verifies that the representation of spatial attributes in ontology can greatly improve the scientific and accuracy of hospital layout design. Through this research, hospital managers and designers can not only make more reasonable layout decisions in the initial design stage, but also improve the efficiency of resource utilization, optimize the design plan.

In ontology modelling, it is important to pay attention to the inter-relationship and compatibility between spatial types to ensure that hospitals can provide high-quality medical services at the same time, it can also provide a safe, comfortable and efficient environment for patients and staff.

The Figure 5.6 represents a structured layout of this hospital space ontology with multiple rooms and facilities, indicating not only the physical layout but also the hierarchical and functional relationships between different entities. The different colours and types of arrows help distinguish the nature of these relationships, providing a clear understanding of the hospital's structure and organization.

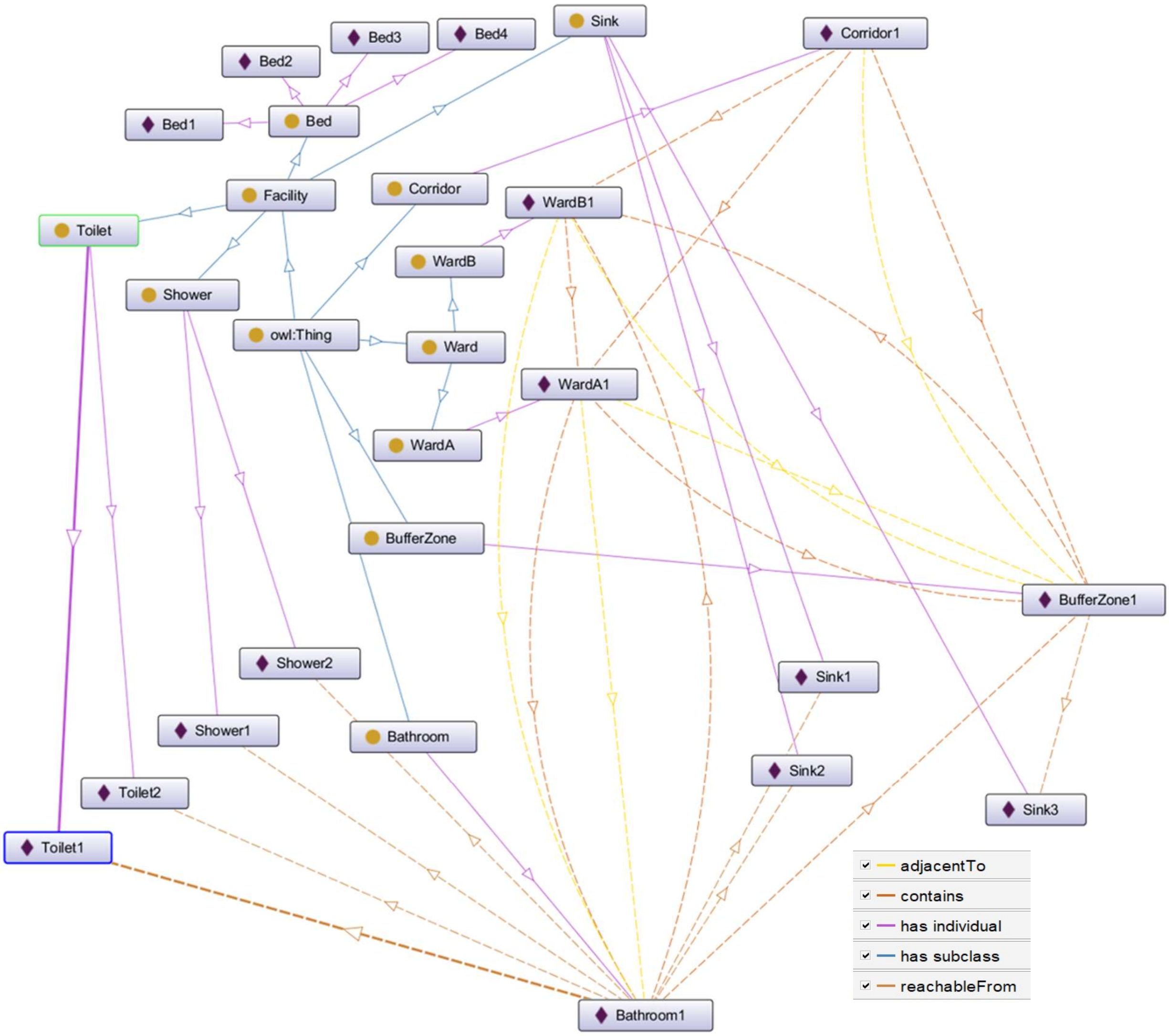



Figure 5.6 Structured layout of space ontology for cabin hospital

## 5.2 Develop an ontology for BIM data

To semantically represent BIM, an ontology is required to define the structure and organization of building information (Karshenas & Niknam, 2013). The AEC-

FM industry currently lacks a standardized ontology for converting BIM data into a semantic format. Bates et al. (Beetz et al., 2009a) use EXPRESS-to-OWL conversion program to develop ifcOWL ontology. The goal of the Linked Data Working Group (LDWG) (*BuildingSMART: Linked Data Working Group* , 2016) established under the umbrella of BuildingSMART International is to formalize the standard ifcOWL ontology; the recommendations made by the group can be found in (*BuildingSMART Proposed Recommendation: EXPRESS-to-OWL conversion routine,* 2015). There are ongoing efforts to standardize the ifcOWL ontology (3rd International Workshop on Linked Data in Architecture and Construction (LDAC), 2016; Pau(Curry et al., 2013)wels and Terkaj, 2016).

The AEC-FM industry comprises diverse individuals and organizations working collaboratively across various fields of expertise. Approaches for developing ontologies in such a multi-disciplinary environment include:

1) Single Comprehensive Ontology (Behzadan et al., 2015): Develop one overarching ontology that encompasses all knowledge domains involved throughout the building lifecycle. This approach provides a unified vocabulary for semantic specification across the AEC-FM sector. However, integrating all AEC-FM information sources with this global ontology can be challenging. Successful implementation would result in a complex ontology with thousands of concepts and relationships, which could be difficult to understand and maintain.
2) Each domain develops its own ontology separately. For instance, architecture might create ontologies for design, scheduling, cost estimation, procurement, standards, and facility management(Zeb & Froese, 2011). While this approach allows flexibility in updating or expanding domain-specific ontologies without impacting others, it introduces challenges in information sharing. The lack of a common vocabulary among different domain ontologies makes it difficult to align or compare them. Sharing information across domains necessitates ontology mapping, which is a manual and time-consuming process similar to IDM and MVD development. Consequently,

this method may not offer significant improvements over current information exchange practices; it merely replaces domain data models with ontologies and IDM/MVD with ontology mapping.

3) Extended Shared Ontology (Özturk, 2021): Each domain develops its own ontology by extending a shared foundational ontology. This method balances between having a single comprehensive ontology and independent domain-specific ontologies. By building upon a common base, domains can maintain consistency while tailoring the ontology to their specific needs. This approach facilitates easier integration and sharing of information compared to completely independent ontologies.

Each of these methods has its advantages and challenges, and the choice of approach depends on the specific requirements and constraints of the AEC-FM project.

To address the limitations of Method 2, Method 3 was introduced. This approach involves creating a shared ontology that encompasses common concepts applicable across all domains of the building lifecycle. Subsequently, each domain develops its own ontology by extending this shared foundational ontology. The primary role of the shared ontology is to establish a unified vocabulary that facilitates consistency and interoperability among the concepts used in the various domain-specific ontologies (Karshenas & Niknam, 2013) .

In the context of BIM, the preparatory work for integrating architectural information and construction process information derived from user experience is essential. This preparatory phase aims to define a data structure that establishes a knowledge-based centralized repository (ontology-based), capable of enabling inference mechanisms. Such a model-centric approach endows the system with robustness and flexibility.

Drawing upon the reviewed research by (Zhao et al., 2016) , they propose a knowledge-based system supported by ontology. By employing this method, the three components, users, BIM, and inference mechanisms, achieve alignment through a shared ontology, serving as the standard for their domain of shared interest. The ontology facilitates communication between these components, even if they employ entirely different knowledge representation mechanisms and data exchange formats. The objective of the ontology is to formalize the domain knowledge and operational knowledge of architectural space entities.

Specifically, the key points of this model are as follows:

1) Establishing a unified data structure for architectural information and construction process information. This standardizes the storage and retrieval of information, enhancing its usability and consistency.
2) Creating a knowledge-based centralized repository that enables comprehensive management of all relevant information. This repository is founded on ontology to support complex inference mechanisms.
3) Inference Mechanisms: Utilizing inference mechanisms, the system can analyse and deduce information from the repository, providing more intelligent and automated functionalities.

The ontology acts as a bridge within the system, ensuring seamless communication and data exchange between users, BIM, and inference mechanisms. It unifies disparate knowledge representation mechanisms, facilitating coordinated operation among the components.

Formalizing the domain knowledge and operational knowledge of architectural space entities to make it more systematic and standardized, thereby enhancing information sharing and interoperability.

Through these methods, an ontology-supported knowledge-based system can enhance the efficiency and intelligence of Building Information Modelling, promoting collaboration among users, BIM, and inference mechanisms.

### 5.2.1 Cabin Hospital BIM Model

In the process of exploring ontology-based cabin hospital layout design, the development of ontology which is closely related to BIM data is a key step to realize intelligent and automatic design. The aim of Cabin Hospital BIM model is to create a hospital architecture model that can adapt to the changing needs and temporary space adjustment in the future, to adapt to innovative treatment procedures, changes in population flow or public health emergencies.

In order to achieve this goal, the BIM model needs elastic parameterization. Specifically, this involves extending the IFC standard to better support custom parameter settings for a variety of hospital specialty equipment and space requirements.

On this basis, the ontology is used to further optimize BIM data. The BIM model can return to verify whether the design meets the requirements of the spatial constraint relationship according to the rules defined in the ontology. For example, whether the ward has a bathroom, the ontology rule library can query to verify whether the design follows the rule.

Combining the methods and mechanisms mentioned above, the ontology-based cabin hospital BIM model can not only dynamically respond to the changes of internal and external environment, but also provide a new model for hospital BIM, it will also greatly promote the automation and intelligence level of hospital layout design, and provide solid technical support for highly personalized and flexible requirements in hospital design in the future.

This study integrates two different epidemiological hospitals based on the construction model of the NHS hospital and the spatial zoning of the Huoshenshan hospital.

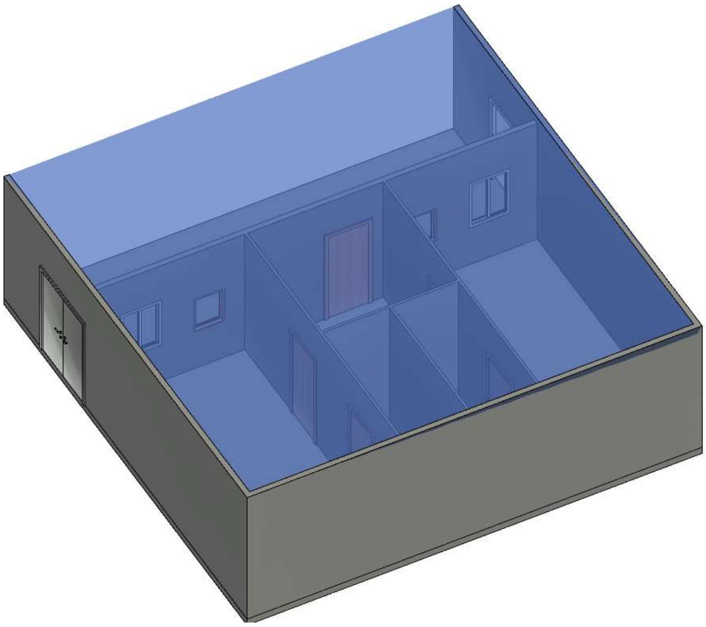


Figure 5.7 Partial hospital BIM model

The BIM model was created using Autodesk Revit software, which is shown in Figure 5.7. As this research focuses more on the functional space, details such as the furniture within the space were not taken into consideration during the BIM model building process. At the same time, in order to reduce the complexity of the model, the details of the building entities were blurred and only the basic structural equipment provided by Revit was selected. By analysing the above two hospitals, the ward layout was reduced to a minimum repeatable unit as shown in Figure 5.7. To ensure connectivity of the space, the ward unit corridors are connected by doors.

## 5.2.2 Ontology from IFC

Developing the ontology of building information model BIM-RRB- data is an important step to realize the cabin hospital layout design. IFC is an open and internationally recognized standard BIM data model, it plays a basic supporting role in the construction of the ontology of hospital layout. IFC ontology provides a complete set of data models, which can represent the various elements of a building project and their relationships in detail. It is one of the core technologies to realize the overall digitization and intelligent of hospital layout.

In the design of hospital layouts, IFC ontology can be used to accurately describe all kinds of spatial function units and their attributes. For example, through the pre-defined entities and relationships in the IFC, it is possible to precisely define the space of the consultation room, the operating room, the ward and the supporting facilities such as the waiting area, the clean area, etc., and their specific location, volume, material attributes of digital coding for the follow-up elastic analysis and design optimization to lay a data foundation.

Elastic parameterization in BIM model is a key extension of IFC ontology, which allows designers to introduce uncertainty and variability into the model in order to adjust and optimize the hospital space design scheme in real-time. By setting up variables such as adjustable space modules, wall structures and accessories, the design scheme can keep the overall layout in harmony and adapt to changes brought about by future use needs or technological developments. These flexible parameterized functions have realized the transformation from a single function of original BIM software to multi-level, dynamic and interactive design process.

The introduction of IFC also deepens the automatic checking of design specifications, which is based on the built-in logical rules of the ontology model. By matching the ontology rule base, the design scheme can realize the real-time compliance check, such as the convenience of patient flow line, the barrier-free design standard of facilities and so on. In hospital layout design, especially in complex engineering, this kind of automatic inspection greatly improves the design efficiency and reduces the possibility of human error.

In addition, an improved BIM data model utilizing IFC ontology can facilitate cross-professional data interaction, which is critical to integrating information about the internal and external environment of a hospital. The layout design of the hospital should not only consider the rational layout of the internal functional areas, but also pay attention to the relationship between the building and the external traffic, urban facilities, etc.. IFC's multi-dimensional analysis platform makes it possible to optimize and adjust hospital layout at a more macro level by pooling and analyzing external data such as urban planning and traffic models.

IfcOWL provides a Web Ontology Language (OWL) representation of the

Industrial Foundation Class (IFC) schema. Using the ifcOWL ontology, building data can be represented using state-of-the-art web technologies (Semantic Web and Linked Data technologies). As a result, IFC data can be used in a directed labelled graph (RDF).

The data transformation process relies on IfcOpenShell as the IFC parser. IfcOpenShell is an Open Source (LGPL) software library for processing IFC. Full parsing support is provided for IFC2x3 TC1, IFC4 Add2 TC1, IFC4x1, IFC4x2 and IFC4x3 Add2. Extensive geometry support has been implemented for IFC versions IFC2x3 TC1 and IFC4 Add2 TC1. Support for arbitrary IFC modes can be extended at compile time with C++ and at run time with Python. The following figure 5.8 shows the code used for data conversion.

```python
from ifcopenshell import open
import rdflib
from rdflib import RDF, Graph, Namespace, URIRef

# Open IFC file
ifc_file = open("model.ifc")

# Create RDF graph
g = Graph()

# Define namespaces
IFC = Namespace("http://www.buildingsmart-tech.org/ifcOWL/IFC4#")

# Example: Convert IFC entities to RDF triples
for entity in ifc_file:
    uri = URIRef(f"http://example.org/{entity.id()}")
    g.add((uri, RDF.type, IFC[entity.is_a()]))

# Serialize to TTL
g.serialize("model.ttl", format="turtle")
```

Figure 5.8 Data conversion code

This Python script converts an IFC (Industry Foundation Classes) file into RDF (Resource Description Framework) data in Turtle (TTL) format using the ifcopenshell and rdflib libraries. The process begins by loading the IFC model with ifcopenshell.open() and initializing an empty RDF graph. The script defines the IFC OWL namespace (http://www.buildingsmart-tech.org/ifcOWL/IFC4#) to properly classify IFC entities in the semantic web context. During conversion, the code iterates through each entity in the IFC file, generating a unique URI for each based on its ID and adding an RDF triple that states the entity's type according to its IFC class. Finally, the resulting RDF graph is serialized into Turtle format and saved as "model.ttl". While this provides a basic conversion framework, the implementation could be enhanced by including property sets, attributes, and relationships between entities, as well as more sophisticated URI patterns and additional ontologies to enable richer semantic querying and reasoning over the IFC data. In the code shown in Figure5.8, the entity id is the id name automatically generated when the BIM model is dumped into an IFC file, which is relatively complex and not easy to read. For example, the id of the wall is IfcWallStandardCase, which is not suitable for extracting effective information among many long names. In this study, it is considered that only spatial division, and the functional division of space is achieved by being surrounded by different types of walls, so only spatially relevant entity ids are extracted from them to achieve ontological simplification. At the same time, the distinction between types of walls relies on the entities contained by the wall, such as windows and doors, e.g. in section 3.2 wall A has a double opening wide door. Therefore, on the basis of this code, the entity id part is named automatically, in this study, all the entities are walls, doors and windows, so it will be simplified from IfcWall, IfcDoor, IfcWindow, the details are as Figure 5.9.

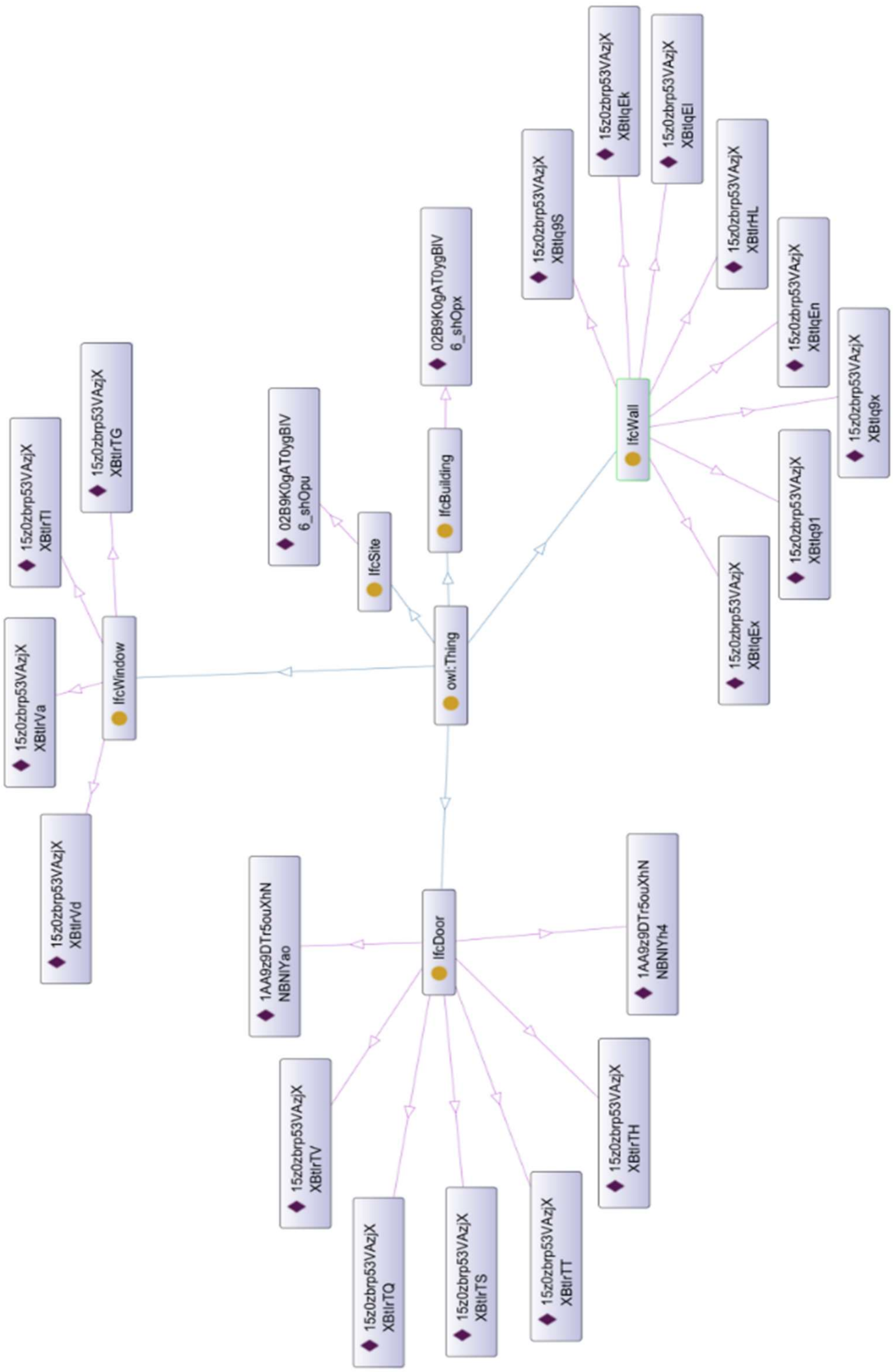



Figure 5.9 Structure of IFC element

In ontology editor Protégé, with overall of this ward unit ontology shows in Figure 5.9. For example of IfcWall class, this class has 10 individuals, which represents all different walls in section 3.2. The individual named “15z0zbrp53VAzjXXBtIq9x” is a wall, it has name "Basic Wall: Aluminium Wall 110:

337106 ", that means wall type D is an aluminium wall with thicknesses 100 mm and tag number 337106. This individual name is IFC-GUID, which is an unique identifier for object instances the IFC specification uses that follows the universal unique identifier standard UUID with its implementation as a globally unique identifier GUID. The generated GUID is compressed for exchange purpose following a published compression function. Table 5-4 lists the correspondence between IFC-GUID and wall types.

Table 5-4 Wall type correspondence

| **IFC-GUID** | **Wall Type** |
|---|---|
| 15z0zbrp53VAzjXXBtIq91 | A |
| 15z0zbrp53VAzjXXBtIqEx | B |
| 15z0zbrp53VAzjXXBtIqEk | C |
| 15z0zbrp53VAzjXXBtIqEl | D |
| 15z0zbrp53VAzjXXBtIqEn | E |
| 15z0zbrp53VAzjXXBtIq9S | F |
| 15z0zbrp53VAzjXXBtIq9x | G |

To sum up, by developing the ontology of BIM data, especially from IFC, the method of hospital layout design can be innovated, realized from the static, the partition layout design to the dynamic, the integration intelligent design transformation. This method not only ensures the rigor and scientific of hospital design process, but also provides strong data support and processing ability to meet the space demand of cabin hospital. In the future, with the further development and application of artificial intelligence technology, the role of IFC in the construction of hospital layout ontology will become more obvious, thus promoting the innovation and development of the whole medical and health construction industry.

## 5.3 Integration of IFC ontology and Space ontology

In order to integrate these two ontologies, the first step is to identify the corresponding concepts, and after extracting the concepts (classes, properties, etc.) from the two ontologies identify the equivalent or related concepts between the two ontologies. This involves the use of string matching and manual checking. In the IFC ontology it is more oriented towards the component entities in this space, whereas the space ontology reflects the spatial relationships, so most of the classes in the IFC ontology are similar to the facilities in the space ontology, and the attributes are contained. the individuals in the IFC ontology are the different kinds of doors, windows, and walls, and are similar to the toilet in the facilities, etc. In order to distinguish IFC instance from space instance, add the class "BuildingElement" as a building component in the space ontology.

The second step is to create mappings between the corresponding concepts, using the standard ontology mapping language OWL (Web Ontology Language)) to define these mappings. Some specific mappings are listed in the Table 5-5.

Table 5-5 Ontology Concept Mapping

| | **IFC Otology (ttl)** | **Space Ontology (xml)** |
|---|---|---|
| **BuildingElement Classes** | IfcWall | http://example.org/covid19-cabin#Wall |
| | IfcDoor | http://example.org/covid19-cabin#Door |
| | IfcWindow | http://example.org/covid19-cabin#Window |
| **Object Properties** | contains | http://example.org/covid19-cabin#contains |
| | adjacentTo | http://example.org/covid19-cabin#adjacentTo |
| **Datatype Properties** | hasName | Similar name properties |

```python
# Extract and map IfcWall entities
walls = ifc_file.by_type('IfcWall')
for wall in walls:
    map_ifc_entity_to_owl(wall, 'IfcWall')

# Extract and map IfcDoor entities
doors = ifc_file.by_type('IfcDoor')
for door in doors:
    map_ifc_entity_to_owl(door, 'IfcDoor')

# Extract and map IfcWindow entities
windows = ifc_file.by_type('IfcWindow')
for window in windows:
    map_ifc_entity_to_owl(window, 'IfcWindow')

# Extract and map IfcBuilding entities
buildings = ifc_file.by_type('IfcBuilding')
for building in buildings:
    map_ifc_entity_to_owl(building, 'IfcBuilding')

# Extract and map IfcSite entities
sites = ifc_file.by_type('IfcSite')
for site in sites:
    map_ifc_entity_to_owl(site, 'IfcSite')
```

Figure 5.10 Data mapping code

Based on the mappings in Table 5-5, the mapping code shown in Figure 5.10 is used. The script begins by loading the IFC file (D12137.ifc) and defining two key namespaces: IFC for the IFC2X3 ontology and EX for custom example properties. An RDF graph is initialized to store the triples, and the namespaces are bound to prefixes for cleaner serialization. The core functionality lies in the map_ifc_entity_to_owl function, which maps IFC entities to OWL by creating URIs from their GlobalId, assigning their IFC type (e.g., IfcWall), and optionally adding their Name and Description as literals with custom predicates (ex:hasName, ex:hasDescription). The script then processes specific IFC entity types—walls, doors, windows, buildings, and sites—by extracting them from the IFC file and passing them to the mapping function. This selective approach allows for focused conversion of relevant entities while omitting others. Finally, the RDF graph is serialized into Turtle format and saved as ward.owl. The script provides a structured foundation for IFC-to-OWL conversion, though it could be extended to handle more entity types, properties, and relationships for richer semantic representation. The output enables semantic querying and integration with other linked data in ontology-driven applications.

The final step is to merge the ontology, using RDF library and the mappings defined in the previous step. Fig 5.11 is the code of this process, running the code results in printing out the first 1000 characters as a display, The error suggests that the attribute "{http://www.w3.org/1999/02/22-rdf-syntax-ns#}" about is not found in the element tagged as "{http://example.org/covid19-cabin#}width". This might mean that the structure of the XML elements is different than assumed. After inspecting the actual structure of the XML elements in detail, the cause of the error was learned to be a conflict on the data Property of hasName in the IFC ontology and "AnnotationProperty rdf:about=http://www.co-ode.org/ontologies/ont.owl#hasName in the space ontology. "hasName" data include size of entities, based on the above error, the mapping of the data property "hasName" is updated to "length" and "hight".

```
# Define namespaces
EX = Namespace("http://example.com/ifc2owl#")
IFC = Namespace("http://ifcowl.openbimstandards.org/IFC2X3_TC1#")
COVID = Namespace("http://example.org/covid19-cabin#")

# Create a new graph for the merged ontology
merged_graph = rdflib.Graph()

# Add data from the Turtle ontology
for stmt in ttl_graph:
    merged_graph.add(stmt)

# Map XML data to RDF
for elem in xml_root.iter():
    if elem.tag == "{http://example.org/covid19-cabin#}IfcDoor":
        door_uri = URIRef(elem.attrib["{http://www.w3.org/1999/02/22-rdf-syntax-ns#}about"
        merged_graph.add((door_uri, rdflib.RDF.type, COVID.IfcDoor))

# Serialize the merged graph to Turtle format
merged_turtle = merged_graph.serialize(format="turtle")

print(merged_turtle)
```

Figure 5.11 Code for Concept Merging

The goal of extending mappings to include more detailed and comprehensive mappings is to create an integration that coordinates classes, properties, and relationships in two ontologies.


Figure 5.12 Usage of “attachTo”

Since the new classes window and door are added to the ontology, to better describe their relationship to other instances, add the Object property "attachTo", which means flush with other instance, this property’s usage is shown in Figure 5.12. The usage of property means the individuals that use this object property, as shown in the figure, for example, Double-side Door 1 is attached to wall type. The repetition of certain entries, such as "Single Leaf Door" across multiple wall types, implies standardized rules or constraints applied to various cases. In Section 3.2, different types of doors are placed on different walls, and the functions of the spaces formed by different combinations of building components are also different, which enables classification or labeling for different scenes or conditions.

And for IFC-space ontology’s structure of Building elements with facility class is shown in Figure 5.13. It shows the relationship between Space area and building entities. The difference in the combination of building entities forms spaces with different functions.

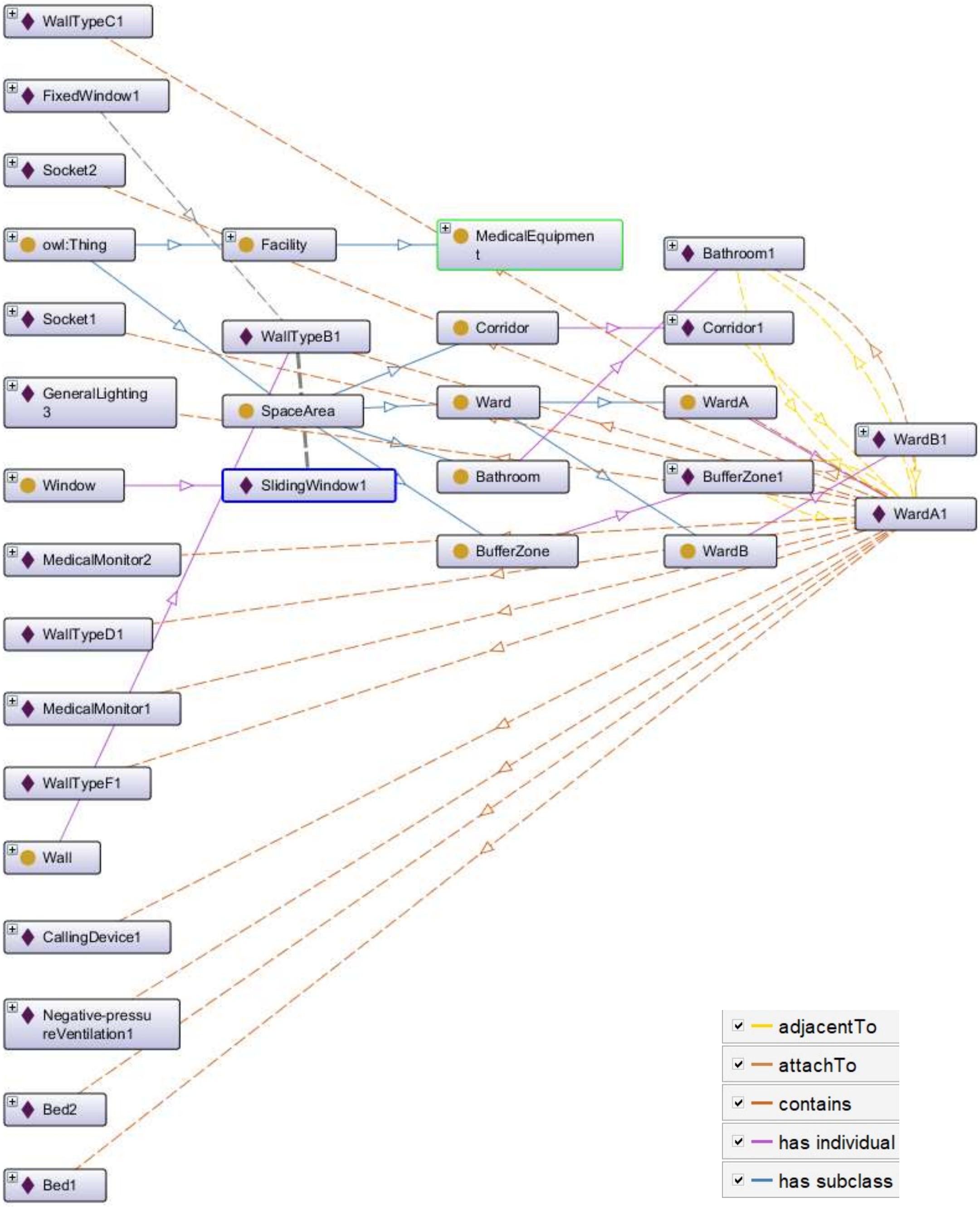



Figure 5.13 Structure relationship between building element and space

## 5.4 Validation Integrated Ontology

### 5.4.1 Formal validation

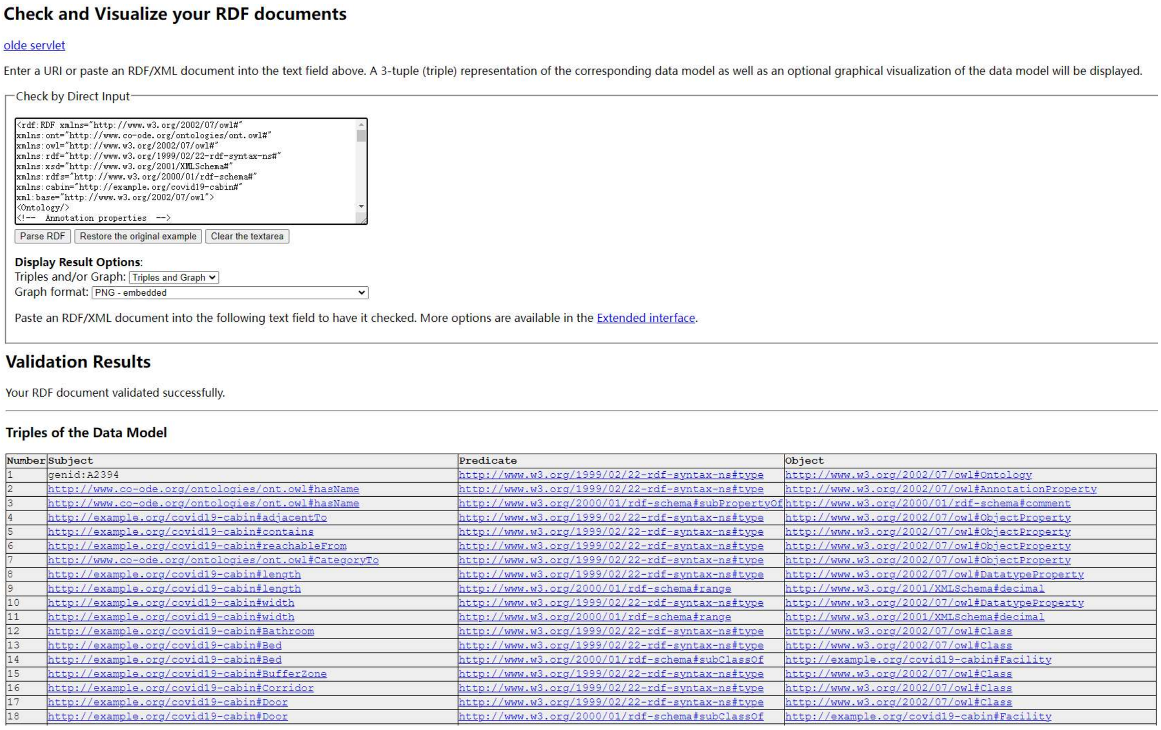



Figure 5.14 Interface of Online validation and result

The W3C (World Wide Web Consortium) online Validation Service is a tool provided by the W3C to help web developers ensure that their code adheres to web standards. This service checks the markup of web documents against the established guidelines set by the W3C, identifying errors and potential issues that could affect the display and functionality of web pages across different browsers and devices. By using this service, developers can improve the quality and accessibility of their websites, leading to better user experiences. The W3C Validation Service is an essential resource for maintaining the integrity and reliability of web content, promoting consistency and best practices in web development.

Fig 5.14 shows the W3C online verification page and the verification result which is successful. This proves that there is no conflict and no error in the syntax of the ontology.

Reasoning over an ontology involves inferring additional knowledge based on the defined relationships, properties, and class hierarchies, also reasoning can help us understand implicit relationships and classifications. Figure 5.19 is the log report of IFC-space ontology, it shows this ontology was processed in 36 ms by Hermit.

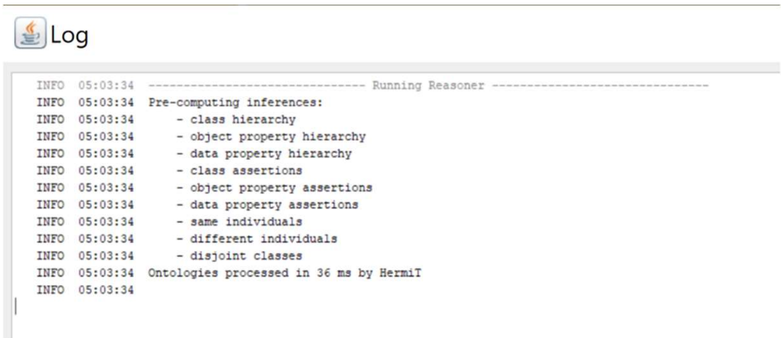



Figure 5.15 Log page for reasoning IFC-space ontology

## 5.4.2 Query validation

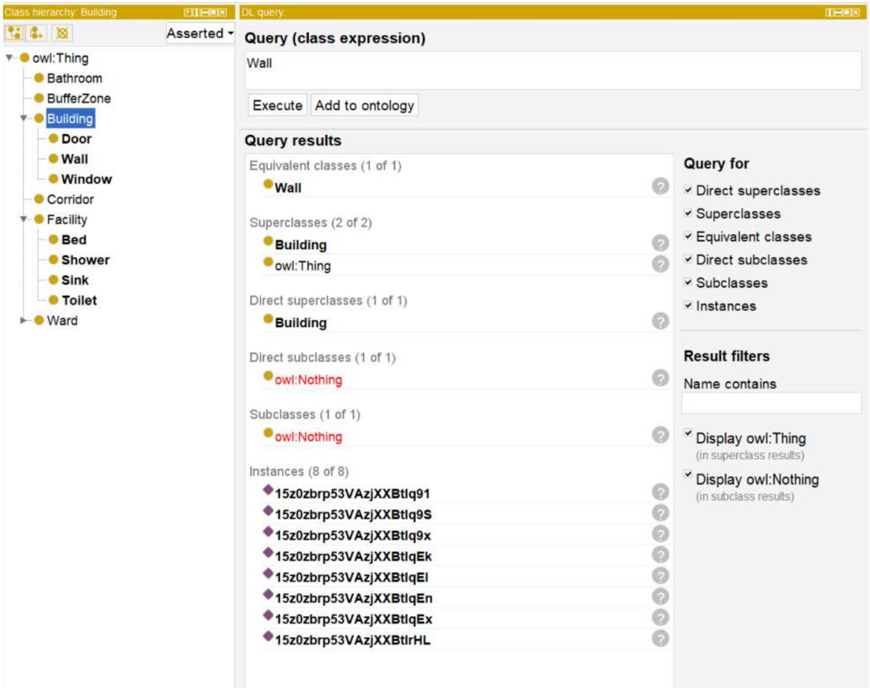



Figure 5.16 DL query interface

In Protégé, DL query is an interface for quickly querying class expressions, which can immediately query the superior and subordinate relationships of classes. For example, the class "wall", it has 2 super class, 1direct superclass, also 8 instances, shows in Figure 5.16.

In section 4.7.2, 8 CQs has been asked. OWL-DL query cannot answer all the CQ. SPARQL queries consist of triple patterns, which are similar to RDF triples but can contain variables. These triple patterns form the basis of a query, where the variables can be matched against the RDF data. SPARQL also supports a wide range of filters, allowing for complex querying based on string matching, numerical comparisons, and more. Table-is the list of CQs, SPARQL query pattern and answers.

Table 5-6 Answers for CQs

**CQ 1 : Which individuals are instances of Facility?**

```
PREFIX cabin: <http://example.org/covid19-cabin#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?facility
WHERE {
  ?facility rdf:type ?type .
  ?type rdfs:subClassOf* cabin:Facility .
}
```

http://www.co-ode.org/ontologies/ont.owl#**MedicalContaminantCollectingBin1**
http://www.co-ode.org/ontologies/ont.owl#**Ventilator2**
http://www.co-ode.org/ontologies/ont.owl#**Ventilator1**
http://www.co-ode.org/ontologies/ont.owl#**MedicalMonitor2**
http://www.co-ode.org/ontologies/ont.owl#**MedicalMonitor3**
http://www.co-ode.org/ontologies/ont.owl#**MedicalMonitor1**
http://www.co-ode.org/ontologies/ont.owl#**MedicalMonitor4**
http://www.co-ode.org/ontologies/ont.owl#**Negative-pressureVentilation1**
http://www.co-ode.org/ontologies/ont.owl#**Negative-pressureVentilation2**
http://www.co-ode.org/ontologies/ont.owl#**CallingDevice1**
http://www.co-ode.org/ontologies/ont.owl#**CallingDevice3**
http://www.co-ode.org/ontologies/ont.owl#**CallingDevice2**
http://www.co-ode.org/ontologies/ont.owl#**CallingDevice4**
http://www.co-ode.org/ontologies/ont.owl#**UVGermicidalLamp1**
http://www.co-ode.org/ontologies/ont.owl#**Socket4**
http://www.co-ode.org/ontologies/ont.owl#**Socket5**

http://www.co-ode.org/ontologies/ont.owl#**Socket2**

http://www.co-ode.org/ontologies/ont.owl#**Socket3**

http://www.co-ode.org/ontologies/ont.owl#**Socket1**

http://example.org/covid19-cabin#**Toilet2**

http://example.org/covid19-cabin#**Toilet1**

http://example.org/covid19-cabin#**Sink2**

http://example.org/covid19-cabin#**Sink3**

http://example.org/covid19-cabin#**Sink1**

http://example.org/covid19-cabin#**Shower2**

http://example.org/covid19-cabin#**Shower1**

http://example.org/covid19-cabin#**Bed4**

http://example.org/covid19-cabin#**Bed3**

http://example.org/covid19-cabin#**Bed2**

http://example.org/covid19-cabin#**Bed1**

http://www.co-ode.org/ontologies/ont.owl#**GeneralLighting4**

http://www.co-ode.org/ontologies/ont.owl#**GeneralLighting5**

http://www.co-ode.org/ontologies/ont.owl#**GeneralLighting1**

http://www.co-ode.org/ontologies/ont.owl#**GeneralLighting2**

http://www.co-ode.org/ontologies/ont.owl#**GeneralLighting3**

**CQ 2 : What are the dimensions of Bathroom1?**

```
PREFIX cabin: <http://example.org/covid19-cabin#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?length ?width
WHERE {
  cabin:Bathroom1 cabin:length ?length .
  cabin:Bathroom1 cabin:width ?width .
}
```

A: length: "**3**"^^http://www.w3.org/2001/XMLSchema#decimal

width: "**3**"^^<http://www.w3.org/2001/XMLSchema#decimal>

**CQ 3 : Which wards are reachable from BufferZone1?**

```
PREFIX cabin: <http://example.org/covid19-cabin#>
```

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?ward
WHERE {
  ?ward rdf:type ?type .
  ?type rdfs:subClassOf* cabin:Ward .
  ?ward cabin:reachableFrom cabin:BufferZone1 .
}
```

A: http://example.org/covid19-cabin#**WardA1**

http://example.org/covid19-cabin#**WardB1**

**CQ4: where all the sink belong to?**

```
PREFIX cabin: <http://example.org/covid19-cabin#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?sink ?container
WHERE {
  ?sink rdf:type cabin:Sink .
  ?container cabin:contains ?sink .
}
```

| sink | container |
|---|---|
| A: | |
| http://example.org/covid19-cabin#Sink2 | http://example.org/covid19-cabin#Bathroom1 |
| http://example.org/covid19-cabin#Sink3 | http://example.org/covid19-cabin#BufferZone1 |
| http://example.org/covid19-cabin#Sink1 | http://example.org/covid19-cabin#Bathroom1 |

**CQ 5 : What medical equipment contained in Ward?**

```
PREFIX cabin: <http://example.org/covid19-cabin#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
SELECT ?MedicalEquipment
WHERE {
  cabin:WardA1 cabin:contains ?MedicalEquipment .
}
```

A: http://www.co-ode.org/ontologies/ont.owl#**WallTypeD1**
http://www.co-ode.org/ontologies/ont.owl#**GeneralLighting3**
http://example.org/covid19-cabin#**Bed2**
http://www.co-ode.org/ontologies/ont.owl#**WallTypeC1**
http://www.co-ode.org/ontologies/ont.owl#**WallTypeB1**
http://www.co-ode.org/ontologies/ont.owl#**Negative-pressureVentilation1**
http://www.co-ode.org/ontologies/ont.owl#**MedicalMonitor2**
http://www.co-ode.org/ontologies/ont.owl#**MedicalMonitor1**
http://www.co-ode.org/ontologies/ont.owl#**Socket2**
http://www.co-ode.org/ontologies/ont.owl#**CallingDevice1**
http://www.co-ode.org/ontologies/ont.owl#**WallTypeF1**
http://example.org/covid19-cabin#**Bed1**
http://www.co-ode.org/ontologies/ont.owl#**Socket1**

## 5.5 Modelling of spatial relationships and constraints

Building on the spatial ontology framework and BIM integration discussed in the preceding sections, this part of the study focuses on transforming semantic spatial definitions into actionable layout constraints. The relationships and attributes previously encoded in OWL—such as spatial adjacency (adjacentTo), containment (contains), reachability (reachableFrom), and physical dimensions (length, width, height)—now serve as critical inputs for the computational optimization of cabin hospital layouts. By formalizing these relationships into a constraint-based mathematical model, the system ensures that the final design meets both spatial feasibility and healthcare-specific requirements.

In the ontology structure shown earlier in Figure 5.6, the spatial entities (e.g., wards, buffer zones, bathrooms) are semantically connected through hierarchical classes and object properties. These formal definitions lay the foundation for the spatial configuration model. For example, adjacency constraints between wards and buffer zones are derived from the reachableFrom and adjacentTo object properties, while the containment of equipment in rooms maps to the contains property. These ontological relationships are systematically translated into numerical constraints to guide spatial layout.

The spatial configuration problem is formulated as a Constraint Satisfaction Problem (CSP), which integrates both the physical geometry (e.g., size, position) and logical relations (e.g., access paths, separation rules). The model is further enhanced by employing dynamic semantic web technologies, which allow real-time reasoning and adaptation when the underlying ontology is updated. This ensures that spatial rules are not only statically defined but also actively govern the optimization process.

As illustrated in Figure 5.17, the ward layout problem is treated as a continuous spatial division within a fixed rectangular site, with clearly defined length (L), width (W), and the dimensions of individual spaces (l, w). Each space, corresponding to an OWL individual of class Ward, Bathroom, or BufferZone, is positioned based on center coordinates (xi, yi) and must adhere to non-overlapping, boundary, and alignment constraints. These are directly tied to previously established data properties (see Figure 5.5) and are necessary to ensure compliance with both functional and safety requirements.

In the following content, three core constraint categories will be explained, which are Non-overlapping Constraint, Boundary Constraint and Alignment Constraint.

Non-overlapping Constraint ensures that no two spatial units intersect. This reflects the ontology's assumption that each spatial entity instance occupies a distinct and exclusive area.

Boundary Constraint guarantees that all entities lie within the site limits—critical for site feasibility and derived from the spatial location and size data properties.

Alignment Constraint organizes the spatial units in a consistent left-to-right, top-down pattern, supporting clear workflow paths and minimizing travel distances, which relates to the logical sequencing implied by the reachableFrom relationships.

Python code implementing these constraints using the constraint library and visualized through matplotlib is shown in Figures 5.18 and 5.19. The layout algorithm aims to maximize spatial efficiency while conforming to both hard constraints (e.g., spacing, containment) and soft objectives (e.g., layout uniformity, optimal capacity).

In this ontology-driven approach, constraints are not arbitrary but are systematically derived from a validated semantic model. This integration of BIM-derived geometry with ontology-based semantics (as previously demonstrated in Figures 5.1, 5.13, and 5.15) ensures both practical implementability and logical consistency. Moreover, the modularity of the ontology allows for future enhancements, such as incorporating new room types or medical technologies, without requiring a complete redesign of the spatial layout engine.

In summary, this section demonstrates how formal semantic representations are operationalized into a robust spatial optimization model. This integration significantly enhances the flexibility, intelligence, and resilience of cabin hospital layout design—key considerations for rapidly deployable healthcare infrastructure.

### 5.5.1 CSP Algorithm

In the process of ontology-based cabin hospital layout design, the modelling of spatial relations and constraints is crucial to ensure design quality and adaptability. This section focuses on the establishment of a comprehensive reflection of the clinical space between the functional units and its constraints of the mathematical model, to capture and map the interactions and constraints of spatial elements in complex medical activities.

To achieve this goal, the first step is to build a detailed ontology library of spatial relations, which describes the physical attributes and logical functions of hospital space in a formal way, which has been described in detail in previous sections. In the ontology library, spatial units are given static attributes such as area, location, shape. By introducing ontology rule base, the decision-making in the design process can automatically consider the complex constraints from macro to micro.

Furthermore, the model uses dynamic semantic web technology and constraint satisfaction problem (CSP) algorithm to optimize the spatial configuration. The Dynamic Semantic Web can not only process and infer complex queries about spatial attributes and relationships, but also automatically adapt to changes when ontology knowledge is updated, the CSP algorithm can fully explore the potential combination of different spatial units to determine the best match. This approach also supports robust analysis, that is, the introduction of random factors or changes in assumed conditions into the spatial configuration to assess the resilience of the design to future uncertainties. As the model matures, this integrated spatial relation and constraint modeling can not only guide the formulation of the initial design scheme, but also assist the rapid assessment and decision-making in the subsequent design elaboration and revision stages.

The ward layout problem is represented as a continuous multi-ward layout problem, assuming that the space to be laid out is set up as a rectangular structure, and the layout division abides by the following principles: the length and width of the initial site and the spaces to be laid out are known, and the spaces are laid out in a top-to-bottom order from left to right. The model parameters are described as follows, and the relationship between the parameters is shown in Figure 5.21.

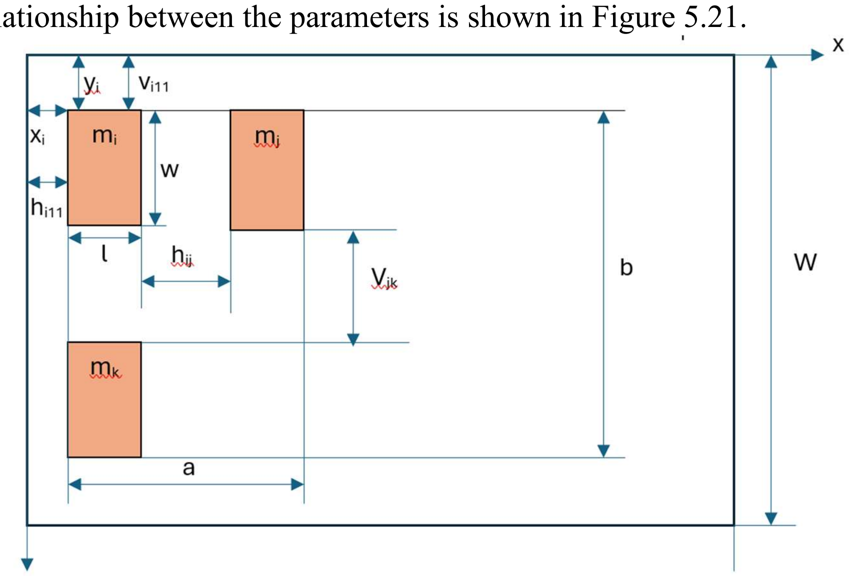



Figure 5.17 Space plan graph

In the Figure 5.17, the x-y plane coordinate system is established in which mi,mj,mk represent the i-th, j-th and k-th space respectively. L represents the total length of whole space. W represents the total width of whole space, xi and yi represent the centre coordinates in the x, y direction of the space mi. And l, w represent the length and width of the space which are 3 meter and 6 meters, hij represent the minimum horizontal distance between the space mi and mj; vjk represent the minimum vertical distance between the space mj and mk which is 3 meters. The hi11 and vi11 denote the minimum distance between space mi and the boundary in the horizontal and vertical directions, and a and b denote the maximum horizontal and minimum vertical distance in all spaces.

The parameters are as follows:

L: Total length of the site

W: Total width of the site

n: Number of space

l: Length of each space (given as 3 meters)

w: Width of each space (given as 6 meters)

xi : Centre coordinate of space mi in the x-direction (range from l/2 to L-l/2)

yi: Centre coordinate of space mi in the y-direction (range from w/2 to W-l/2)

hij: Minimum horizontal distance between space mi and mj (given as 0 meters)

vjk: Minimum vertical distance between spaces mj_and mk (given as 3 meters)

hi11: Minimum horizontal distance between space mi and the site boundary (given as 0 meters)

vi11: Minimum vertical distance between space mi and the site boundary (given as 0 meters)

a: Maximum horizontal distance in the entire layout

b: Maximum vertical distance in the entire layout

A reasonable spatial layout should ensure that each space division does not overlap and is arranged in the order from top to bottom within the given initial site range. These are the three constraints defined in this research: Non-overlapping; Boundary and Alignment.

1) Non-overlapping constraint:

Each space must not overlap with any other space. This can be formulated using the minimum horizontal and vertical distances between spaces:

$$|x_i - x_j| \geq \frac{l}{2} + \frac{l}{2} + h_{ij} = l + h_{ij}$$

$$|y_i - y_j| \geq \frac{w}{2} + \frac{w}{2} + v_{ij} = w + v_{ij}$$

2) Boundary constraint:

Each space must be within the boundaries of the overall space:

$$x_i - \frac{l}{2} \geq h_{i11}$$

$$x_i + \frac{l}{2} \leq L - h_{i11}$$

$$y_i - \frac{w}{2} \geq v_{i11}$$

$$y_i + \frac{l}{2} \geq W - v_{i11}$$

3) Alignment constraint:

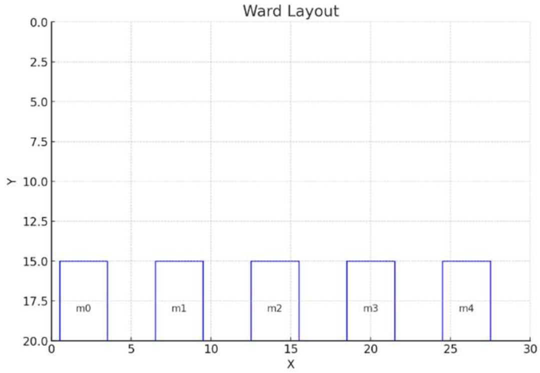
Since the spaces are laid out in a top-to-bottom order from left to right, we need to enforce alignment constraints. Assuming spaces are aligned in rows:

For spaces mi and mj in the same row: $x_j = x_i + l + h_{ij}$

For spaces mj and mk in the same column: $y_k = y_i - (w + v_{ij})$

## 5.5.2 Auto cabin layout Experiment

Code editing in Python based on the above constraints using the constraints library, and visualize a layout plan via 'matplotlib'. As a validation use these data as examples: L = 30, W = 20, l = 3, w = 6, h_ij = 3, v_jk = 3, h_i11 = 2, v_i11 = 2, num_spaces = 5. The result and code are shown in Figure 5.18.




```
from constraint import Problem, AllDifferentConstraint

def ward_layout_csp(L, W, l, w, h_ij, v_jk, h_i11, v_i11, num_spaces):
    problem = Problem()
    coords = [(f"x{i}", f"y{i}") for i in range(num_spaces)]

    # Add variables with their domains
    for (x, y) in coords:
        problem.addVariable(x, range(l // 2, L - l // 2 + 1))
        problem.addVariable(y, range(w // 2, W - w // 2 + 1))

    # Add non-overlapping constraints
    for i in range(num_spaces):
        for j in range(i + 1, num_spaces):
            x_i, y_i = coords[i]
            x_j, y_j = coords[j]
            problem.addConstraint(lambda xi, yi, xj, yj, h=h_ij: abs(xi - xj) >= l + h or abs(yi - yj) >= w + v_jk, (x_i, y_i, x_j, y_

    # Add boundary constraints
    for (x, y) in coords:
        problem.addConstraint(lambda xi, h=h_i11: xi - l // 2 >= h, (x,))
        problem.addConstraint(lambda xi, h=h_i11, L=L: xi + l // 2 <= L - h, (x,))
        problem.addConstraint(lambda yi, v=v_i11: yi - w // 2 >= v, (y,))
        problem.addConstraint(lambda yi, v=v_i11, W=W: yi + w // 2 <= W - v, (y,))

    # Add alignment constraints (for simplicity, aligning all in one row and one column)
    for i in range(1, num_spaces):
        problem.addConstraint(lambda xi, x_prev, h=h_ij: xi == x_prev + l + h, (coords[i][0], coords[i-1][0]))
        problem.addConstraint(lambda yi, y_prev, v=v_jk: yi == y_prev - (w + v), (coords[i][1], coords[i-1][1]))

    return problem.getSolutions()
```

Figure 5.18 Code for space layout and result

The aim to maximize the overall horizontal and vertical distances to ensure uniformity and optimal utilization of the layout space. The objective function is:

$$\max\ (a + b)$$

$$a = \left\{ \max_{i=1}^{n} \left( x_i + \frac{l}{2} \right) - \min_{i=1}^{n} \left( x_i - \frac{l}{2} \right) \right\}$$

$$b = \left\{ \max_{i=1}^{n} \left( y_i + \frac{w}{2} \right) - \min_{i=1}^{n} \left( y_i - \frac{w}{2} \right) \right\}$$

By using these data as examples: L = 30, W = 20, l = 3, w = 6, h_ij = 0, v_jk = 3, the code and layout result show in Figure 5.19, result says: Maximum number of rooms that fit: 20 rooms.

```
import matplotlib.pyplot as plt
import math

# Define the dimensions and space parameters
L = 30
W = 20
l = 3
w = 6
h_ij = 0
v_jk = 3

# Calculate the maximum number of rooms that fit horizontally and vertically
max_rooms_horizontal = math.floor(L / (l + h_ij))
max_rooms_vertical = math.floor(W / (w + v_jk))

# Calculate the total number of rooms
total_rooms = max_rooms_horizontal * max_rooms_vertical
print(f"Maximum number of rooms that fit: {total_rooms}")

# Function to draw the layout
def draw_max_layout(L, W, l, w, h_ij, v_jk):
    fig, ax = plt.subplots()
    ax.set_xlim(0, L)
    ax.set_ylim(0, W)
    ax.set_aspect('equal')
    plt.gca().invert_yaxis()

    room_index = 0
    for i in range(max_rooms_horizontal):
        for j in range(max_rooms_vertical):
            x = (l / 2) + i * (l + h_ij)
            y = (w / 2) + j * (w + v_jk)
            rect = plt.Rectangle((x - l / 2, y - w / 2), l, w, fill=None, edgecolor='blue')
            ax.add_patch(rect)
            plt.text(x, y, f"m{room_index}", ha='center', va='center')
            room_index += 1

    plt.xlabel("X")
    plt.ylabel("Y")
    plt.title("Maximized Room Layout")
    plt.grid(True)
    plt.show()

# Draw the layout
draw_max_layout(L, W, l, w, h_ij, v_jk)
```

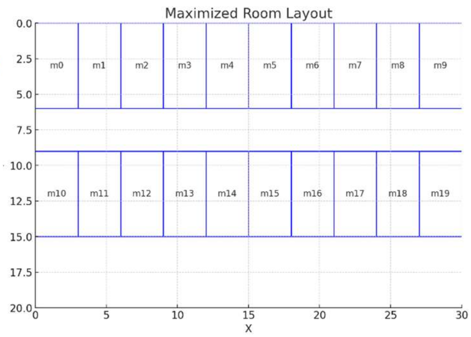



Figure 5.19 Code for optimised space layout

The above results confirm that the algorithm is feasible for automatic layout of square cabin hospitals. The spatial relationships (e.g., adjacentTo, reachableFrom) and attributes (e.g., length, width) formalized in the OWL ontology are operationalized as constraints in the CSP section. For instance, the ontology's reachableFrom property ensures that wards are only accessible via buffer zones, which is enforced in the layout optimization through vertical/horizontal distance parameters (e.g., v_jk=3m). The minimum vertical distance between wards and corridors reflects the infection control logic encoded in the ontology's "reachableFrom" property (Figure 5.4). And, the same is true for Object Property "adjacentTo" (CSP constraint: $|x_i - x_j| \geq \frac{l}{2} + \frac{l}{2} + h_{ij} =$

$l + h_{ij}$ ) and Data property length and width attributes ($l$=3$m$,$w$=6$m$).

However, this hypothetical experiment does not take into account the limitations of room functions and combinations, which will be improved in Chapter 6.

The CSP algorithm operationalizes ontology rules into solvable constraints, ensuring the layout adheres to both functional requirements and physical limits. The integration of ontology (Sections 5.1–5.3) and CSP (Section 5.5) creates a closed-loop system: the ontology defines spatial semantics, while the CSP validates and optimizes their physical implementation.

# Chapter 6 Experimental results

In previous discussions, it is explored the use of automatic layout algorithms in regular, rectangular spaces for hospital arrangements. However, real-world applications often encounter irregularly shaped spaces, necessitating advanced layout techniques. This chapter delves into the complexities of designing a cabin hospital within such irregular environments, providing insights and methodologies for efficient spatial arrangement.

In real scenes, buildings or sites are often irregular. Two distinct irregular spaces are began with an examination of : concave and convex polygons, each presenting unique challenges for layout. Using specified coordinates, we adapt our approach to ensure that rooms fit within these irregular boundaries without overlap, employing visual aids to demonstrate our strategies. Such experiments can prove that the method of this study is feasible and can be quickly adjusted, which is in line with the flexible and variable layout that this study wants to achieve.

The chapter then transitions to a practical application at Brunel University's Michael Sterling and Wilfred Brown buildings. These buildings, chosen for their modern design and connectivity, offer a real-world scenario for implementing our layout strategies. Utilizing advanced mapping technologies like TurtleBot and RTAB-Map, we generate indoor layouts that reflect both the structural and functional requirements of a cabin hospital. This experiment validates the proposed method's ability to rapidly extract critical building information (e.g., spatial coordinates) in scenarios where traditional architectural or site data files are unavailable. The results demonstrate that this approach enables efficient layout design even under constrained or time-sensitive conditions.

Furthermore, we explore the conversion of a significant sports venue, the Ningbo Olympic Sports Centre Comprehensive Training Hall, into a temporary epidemic response facility. This section highlights the integration of traffic management, spatial arrangement, and emergency accessibility, ensuring the venue can efficiently support medical operations during a health crisis.

Throughout the chapter, the emphasis remains on practical application, detailed spatial analysis, and the use of cutting-edge technology to overcome the challenges posed by irregular spaces.

## 6.1 Cabin hospital layout in hypothetical irregular space

In the previous chapter, the automatic layout algorithm used a regular rectangular space as an example for layout arrangement. In reality, it is very common for built building spaces to have irregular shapes. This section mainly explains the spatial layout division using two irregular spaces:

- Concave coordinates: (0,0) (30,0) (30,40) (20,40) (20,35) (5,35) (5,40) (0,40) (0,0)
- Convex coordinates: (30,0) (60,0) (60,20) (55,20) (55,40) (40,40) (40,20) (30,20) (30,0)

To handle an irregular space with the given coordinates, the approach is needed to modify to ensure rooms are placed within the defined polygon. A different method is used to check if the rooms fit within the polygon and avoid overlapping, which is showed in Figure 6.1. In the previous section, the space to be laid out was determined by the space size (length and width), but in this section, the boundaries of the space are determined by coordinate points, so it is necessary to introduce point in the previous code to implement coordinate point input.

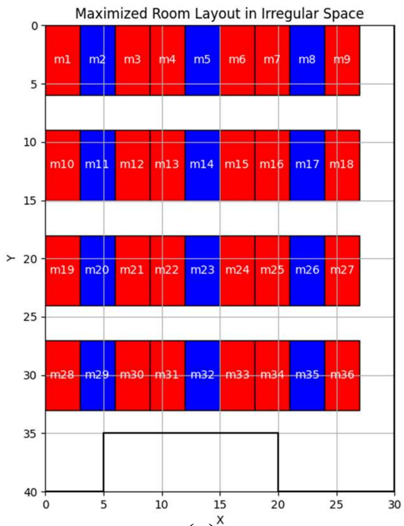
```
import matplotlib.pyplot as plt
import math
from shapely.geometry import Polygon, Point
```

Figure 6.2 Polygon method

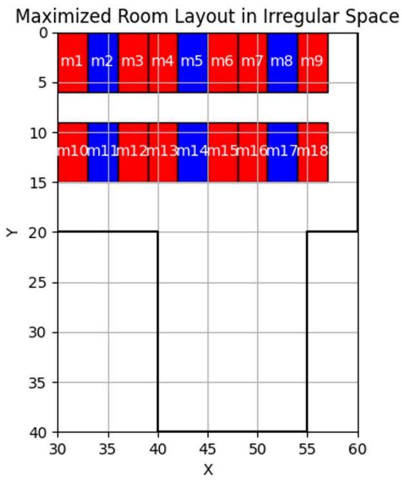
```
# Adjust the number of rooms to be a multiple of 3
row_rooms = (row_rooms // 3) * 3
# Place the rooms if the row can fit a multiple of 3
if row_rooms > 0:
    x = row_start_x
    for i in range(row_rooms):
        if room_fits(x, y, l, w, polygon):
            if i % 3 == 0 or i % 3 == 2:
                color = 'red'
            else:
                color = 'blue'
```

Figure 6.1 Code for Cluster

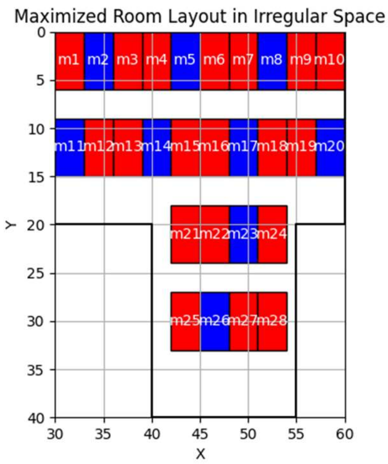
The CPS restrains in Section 5.1.2, meanwhile, according to the relationship in space ontology, the corridor needs to be connected; the two wards need to share a buffer zone and a bathroom, when implementing the CPS constraint, the three rooms need to be considered as a cluster. Furthermore, each row of space can only be arranged a multiple of 3 rooms. These additional restrictions are implemented by the code in Figure 6.2, and the rooms are distinguished by colouring them, red for wards and blue for buffer zone and bathroom. Figure 6.3 shows the layout of two polygonal spaces.




(a)




(b)




(c)

Figure 6.3 Layout Plan Result

In the results, the layout of concave (a) polygons is relatively good. Combined with the attributes and relationships defined by the ontology, this layout meets the requirements of the ontology and there is no large amount of space waste.

Among them, the spatial layout of convex polygons (b) obviously indicates the waste of space, which needs to be improved. To fit more rooms in this convex space, the restriction that the number of rooms per row must be a multiple of three has been removed. The reason for this is that code with this restriction would not be able to run out of results. The optimised space layout is shown in Figure 6.3 (c). The most notable drawbacks in this layout are the non-compliance with the IFC-space ontology location relationship, where four wards have exclusive use of the buffer zones and bathrooms, and the corridor being split into two parts, which does not satisfy the reachability of the corridor to the other buffer zones from IFC-space ontology.

The experimental results demonstrate that this research method is highly adaptable to diverse site dimensions and geometries. It enables rapid layout adjustments when initial outcomes are unsatisfactory, highlighting the approach's inherent flexibility and adaptability. Additionally, the method incorporates a validation step by cross-checking the results with the IFC-space ontology, ensuring the reliability and rationality of the findings. This dual emphasis on dynamic adjustment and rigorous verification underscores the robustness of the experimental outcomes.

## 6.2 Cabin hospital layout in selected site

The selected buildings for renovation are the Michael Sterling Building and the Wilfred Brown building of Brunel University, location shows in Figure 6.4, which are relatively new to the campus, are connected to the internal roads of the university and have the opportunity to be connected to the external transportation, and are open 24 hours a day. The ground floor of the building serves as a common area for events and activities, and the flexibility of the facility to be disassembled and moved around facilitates the installation and transportation of equipment when used as a renovation site. The building has a translucent roof and good interior lighting. The building has a number of entrances and exits that can be opened in the event of an emergency, which facilitates evacuation.

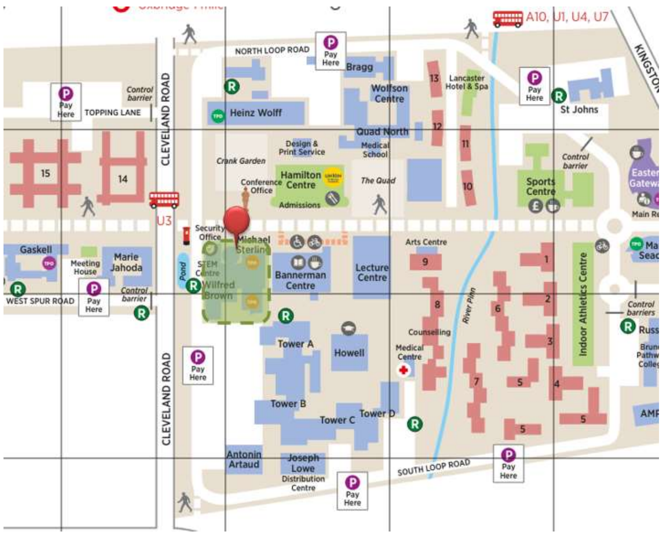



Figure 6.4 Selected site external roads (Google map)

The indoor layout is acquired by TurtleBot loaded with Kinect camera and generated simultaneously by RTAB-MAP (Real-Time Appearance-Based Mapping). RTAB-Map is a SLAM (Simultaneous Localization and Mapping) method utilizing RGB-D cameras, stereo vision, and lidar, structured around a graph-based system. It features an incremental, appearance-based loop closure detector, which employs a bag-of-words approach to assess whether a new image corresponds to a previously visited location or a novel one. When a loop closure is detected and accepted, a new constraint is incorporated into the map graph, which is then optimized to minimize mapping errors. To ensure real-time performance in extensive environments, a memory management strategy restricts the number of locations considered for loop closure detection and graph optimization. RTAB-Map supports 6DoF mapping with handheld devices like the Kinect, stereo cameras, or 3D lidars, and 3DoF mapping with robots equipped with laser rangefinders.

TurtleBot serves as a standard platform for the Robot Operating System (ROS). The name "TurtleBot" traces its origins to the Turtle robot, which was initially controlled by the Logo programming language in 1967. The turtlesim node, introduced in the basic ROS tutorials, emulates the command system of the original Logo turtle program and has become a symbolic representation of ROS. TurtleBot was developed to facilitate the teaching of ROS to beginners and to introduce the concepts of computer programming through Logo. Over time, TurtleBot has established itself as the standard platform for ROS, gaining popularity among developers and students alike.

There are three versions of the TurtleBot(Open Source Robotics Foundation). TurtleBot1, developed in 2010 by Tully (Platform Manager at Open Robotics) and Melonee (CEO of Fetch Robotics) of Willow Garage, was based on iRobot's Roomba research robot, Create, and was made available in 2011. TurtleBot2 followed in 2012, developed by Yujin Robot based on the iClebo Kobuki research robot. In 2017, TurtleBot3 was introduced, featuring enhancements to address the limitations of its predecessors and meet user demands. TurtleBot3 utilizes the ROBOTIS DYNAMIXEL intelligent actuator for its driving mechanism.

TurtleBot3 is a compact, affordable, programmable mobile robot designed for educational purposes, research, hobbies, and product prototyping. The primary aim of TurtleBot3 is to significantly reduce the size and cost of the platform while maintaining functionality and quality, along with providing expandability. The modular nature of TurtleBot3 allows for extensive customization through various mechanical configurations and optional components such as computers and sensors. It also incorporates a low-cost, small single-board computer (SBC) suitable for rugged embedded systems, a 360-degree distance sensor, and 3D printing technology.

The core technologies of TurtleBot3 include SLAM (Simultaneous Localization and Mapping), navigation, and manipulation, making it ideal for home service robots. TurtleBot3 can execute SLAM algorithms to create maps and navigate environments. It can be controlled remotely via a laptop, joypad, or Android-based smartphone and can follow a person's legs as they move around a room. Additionally, TurtleBot3 can function as a mobile manipulator by attaching devices such as the OpenMANIPULATOR, which is compatible with the TurtleBot3 Waffle and Waffle Pi. This compatibility enhances its functionality, making it a comprehensive service robot with advanced SLAM and navigation capabilities. In this section, a 3D map of the selected space is obtained by using an RGB-D camera mounted on a mobile Turtlebot robot, which shows in Figure 6.5.

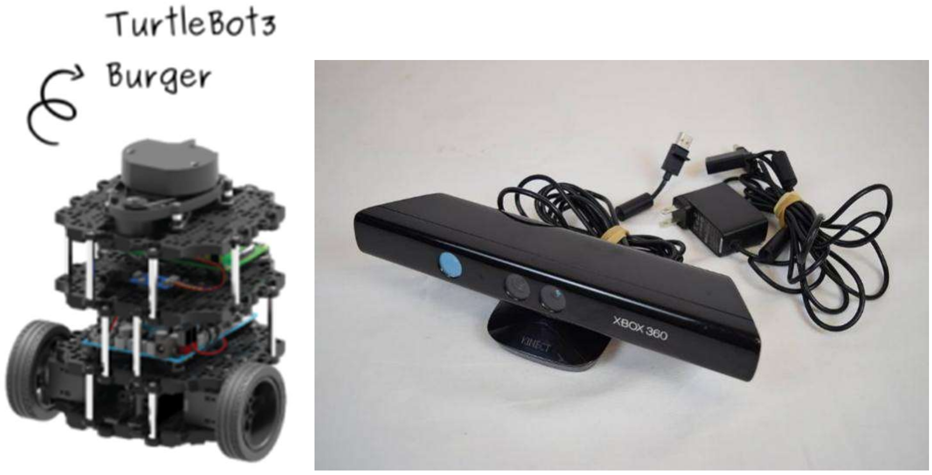



Figure 6.5 TurtleBot3 and Kinect camera

The Figure 6.6 shows the generated indoor layout, the overall mapping time was about 7 hours, the TurtleBot needed to be manually maneuverer and was not able to automatically travel after completing the mapping of the location. After this, handheld Kinect camera for mapping was also tried, the personnel need to pay attention to the display screen all the time to map the progress and then walk, so the route shown in the map is messy. The process took about 4 hours, and RTAB-MAP developed a cell phone, which can also be used for hand-held mapping with the RGB-D camera on models after the 8th generation iPhone. In this way, the map-making process can be observed through the colour change of the drawing area in the screen interface. The interior layout of the building is shown below, and the scale of the space can be well understood through the grid, gird size is 1meter by 1meter.

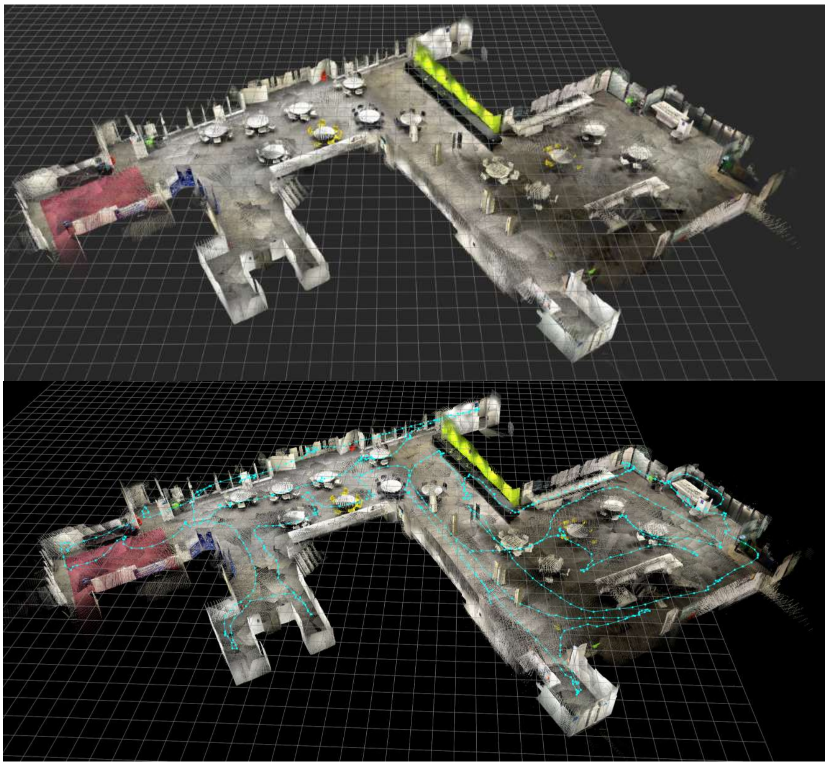


Figure 6.6 Michael Sterling Building and the Wilfred Brown building indoor layout

The data schema of this map can be briefly regarded as a collection of photos of the scene from various angles, so it does not contain spatial-physical information about the points of each element in the scene, and it is not possible to automatically generate the required BIM model of the scene, so it is necessary to build the model manually. The grid in the map provides good dimensional information for BIM, and although there will be some errors in the final model dimensions, these are within acceptable limits. The space occupied by the ward cells that need to be laid out is very regular and one unit is 3 meter by 6 meter,  errors of less than 2 meter cannot fit an unit so it do not affect the results. In the BIM model created, only two common exits were labeled, which is illustrating in Figure 6.7. The doors of these two public exits are relatively wide and are connected to the campus roads, providing sufficient conditions to support the transportation of large equipment and materials. Some of the other exits are fire evacuation passages, and some have steps for pedestrians only. During the construction and operation of the cabin Hospital, these passages will not be considered as main passages. In Figure 6.7, obviously, this is a very irregular polygon, and a series of relative coordinates are extracted for layout arrangement based on the BIM model.

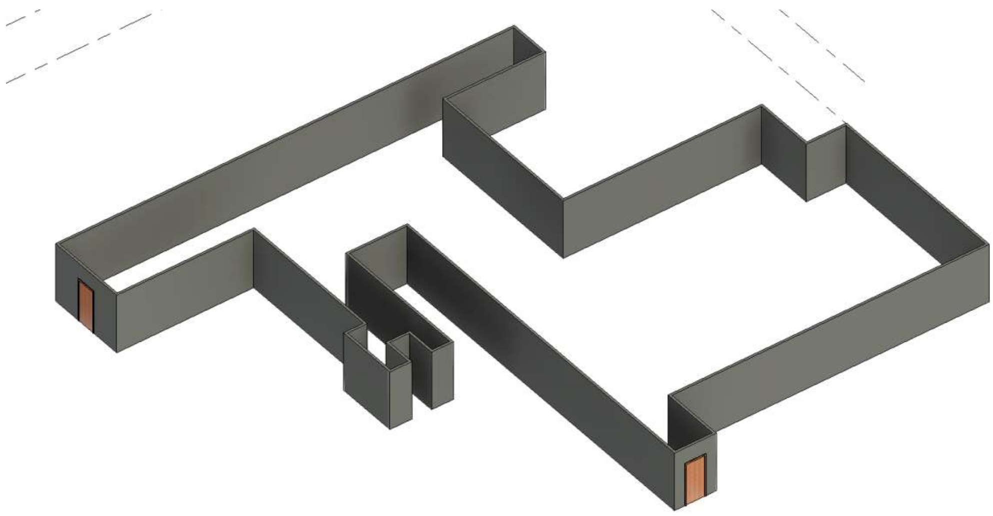
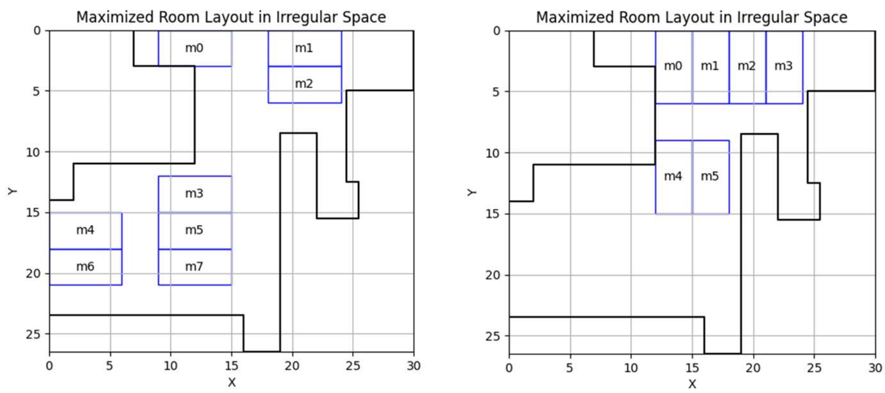


Figure 6.7 BIM model of the selected indoor space




(a) (b)

Figure 6.8 Horizontal layout (a) & Vertical layout (b)

Continuing to use the code in Section 6.1, the result of the layout results as shown in Figure 6.8, which by changing the code which variables, such as l, w, h_ij, v_jk, respectively, for the horizontal layout and vertical layout. The Figure 6.9 outlines a structured workflow for generating a cabin hospital layout using a Constraint Satisfaction Problem (CSP) algorithm. The process begins with inputting space boundary coordinates, which define the physical constraints of the site. These coordinates, along with the CSP algorithm code, drive the computational generation of a cabin hospital layout as the output. The diagram outlines a structured workflow for generating a cabin hospital layout using a Constraint Satisfaction Problem (CSP) algorithm. The process begins with inputting space boundary coordinates, which define the physical constraints of the site. These coordinates, along with the CSP algorithm code, drive the computational generation of a cabin hospital layout as the output.

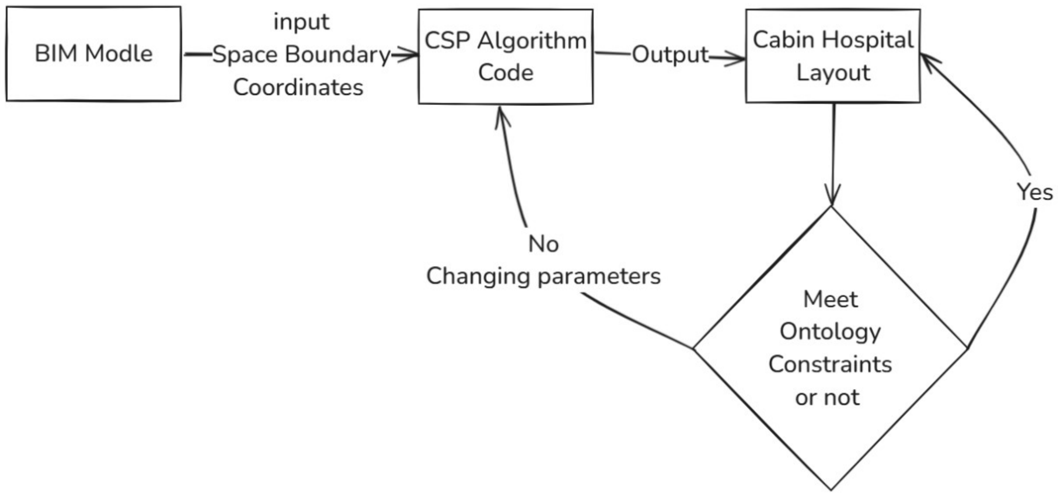



Figure 6.9 The workflow for generating a cabin hospital layout

A critical step in this workflow is evaluating whether the layout meets predefined ontology constraints (e.g., functional requirements, spatial rules). If the constraints are not satisfied, the system loops back to adjusting parameters (e.g., room dimensions, spacing) and regenerates the layout iteratively. This feedback loop ensures flexibility and adaptability in refining the design. Only when the layout aligns with all constraints does the process conclude, guaranteeing that the final output is both logically valid and practically feasible. The diagram thus highlights a systematic, iterative, and ontology-anchored approach to automated spatial planning.

Plan (a) contains 2 more rooms compared with plan (b), however, both of them are not match with the positional relationships represented in IFC-space ontology. Meanwhile considering the situation of Covid-19, less than ten rooms, which only less than four wards, that is not suitable site selection for converting to a cabin hospital.

Although this unsatisfactory result proves that the site is not suitable for the conversion of the pod hospital, such a conclusion is not entirely without practical significance. Instead, the failure of this case proves that the methodology of this study can help design stakeholders to rule out unreasonable site selection in the early stages of design. Moreover, the rule constraints of the ontology are combined to exclude unreasonable layouts, which leads to more flexibility in updating and refining the design plan, which is very beneficial for the design of epidemic hospitals with a short period of time in the design stage.

## 6.3 Implementation in sports venue

The Ningbo Olympic Sports Centre Comprehensive Training Hall is a significant sports facility in Ningbo, China, known for hosting various sports events and training sessions. In this case, converting the venue into a temporary epidemic cabin hospital would require careful planning and consideration of existing traffic conditions to ensure the smooth operation of the facility. For accessibility, this venue is well-connected by a comprehensive road network. Major roads, are showing Figure 6.9, such as the Ningbo Ring Road and Yongjiang North Road provide direct access to the facility, which would facilitate the swift transportation of medical supplies, personnel, and patients. Public transportation options, including buses and taxis, can be adjusted to serve the hospital staff and non-critical patients, minimizing the strain on private vehicle usage, which can ensure that medical vehicles, such as ambulances and medical supplies delivery vehicles, can flow freely during the operation of the temporary hospital. Ample parking spaces can be allocated for medical staff, emergency vehicles, and logistics support, also designated areas for patient drop-off and pick-up can be established to streamline the flow of vehicles. The strategic location near the city centre and other important areas facilitates the quick movement of essential supplies and personnel. It also makes it easier for local residents to access the hospital for medical needs. By leveraging the existing traffic infrastructure and implementing additional measures, the Ningbo Olympic Sports Center

Comprehensive Training Hall can be effectively converted into a temporary epidemic cabin hospital, ensuring efficient and safe operation during a health crisis. The venue building is primarily designed to host large-scale events such as sports competitions and concerts. It features excellent natural lighting, facilitated by a glass curtain wall on the upper sections, along with a reliable water and electricity supply and an efficient ventilation system. Furthermore, the venue has a proven track record in event management, and its operators possess extensive expertise in renovation and installation, ensuring smooth adaptations for cabin hospital requirements.

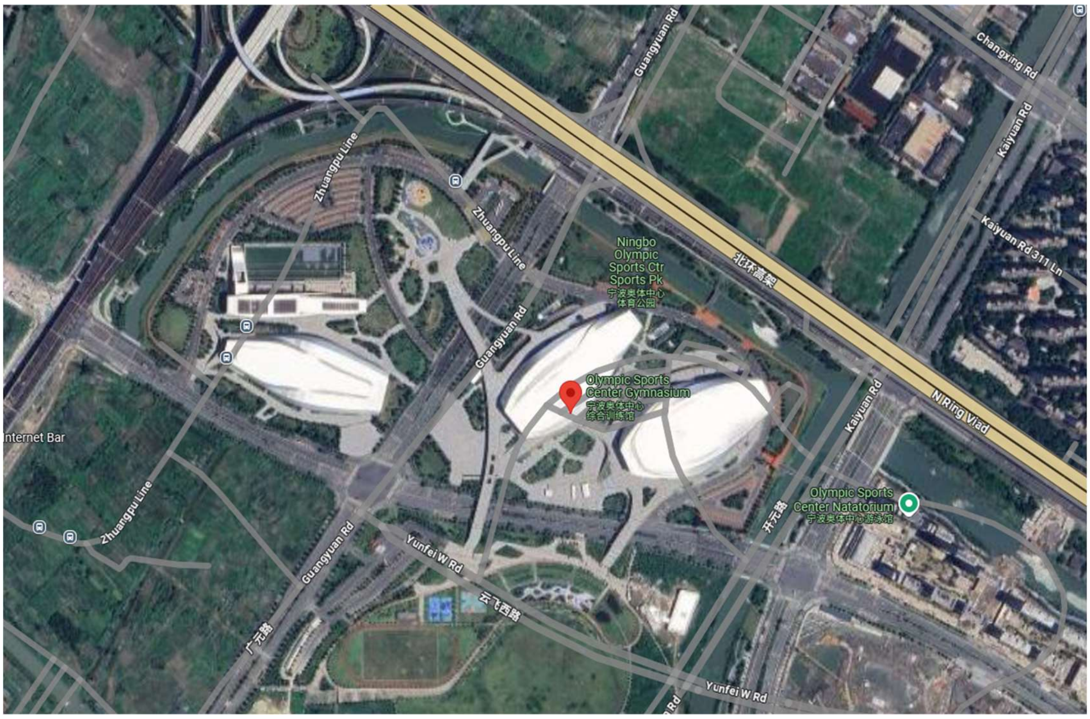



Figure 6.10 External Road of sport venue

The Figure 6.11 is a partial BIM model of the venue, and shows 6 entrances and exits. From Figure 6.11, it can be seen that in addition to the largest space in the middle, there are many spaces of different sizes around it. These spaces can be used as medical staff preparation areas and medical equipment storage areas during converting into cabin hospitals. In the previous two sections of this chapter, only the patient area was laid out, and other areas were not mentioned. In this case implementation, although these areas need to be considered, the existing small spaces around can meet the needs, so the entire space in the middle still continues the layout of the first two section, that is, the layout of the ward toilet and corridor, result is illustrated in Figure 6.11,with 72 rooms, with 48 wards, 24 buffer zones and 24 bathrooms.

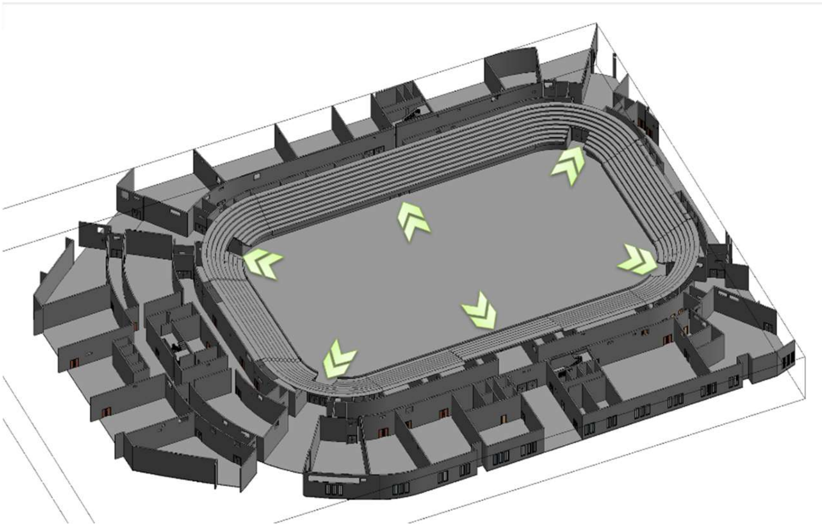


Figure 6.11 BIM model for sport venue

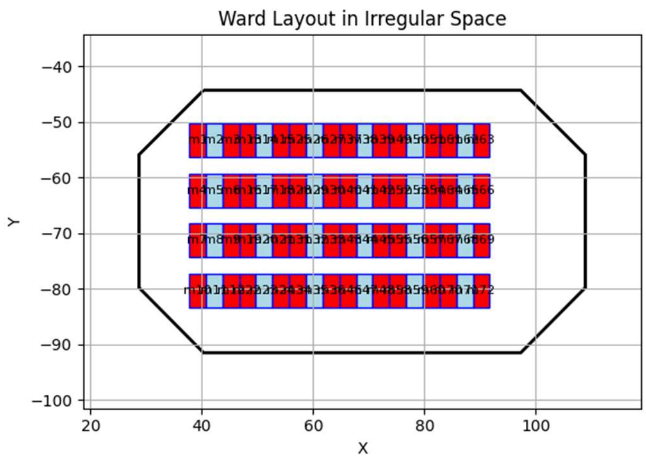



Figure 6.12 Cabin hospital layout

In IFC-space ontology, containment relationships are defined. For example, a ward contains two beds, a buffer contains a sink, and a bathroom contains two toilets, two showers, and two sinks. Now that the number of rooms is obtained through automatic layout, the corresponding facilities required for the conversion of the venue can be easily calculated, that is, 96 beds, 48 toilets and 72 sinks are needed.

In the actual renovation process, managers need to control the overall situation and evaluate the project from the aspects of construction cost, equipment and facility cost, and time. Jalel and Linda(Akaichi & Mhadhbi, 2016) established an ontology related to hospital emergency care, extracted the content related to medical equipment from the ontology and combined it with the IFC-space ontology. In response to the covid-19 epidemic, what is needed in the ward is ventilators. Instances such as doors, windows, and walls have been defined in the IFC-space ontology, and the number of doors, windows, and walls can be obtained as shown in the Table 6-1.

Table 6-1 Facilities accounting

| **Wall type** | **A** | **B** | **C** | **D** | **E** | **F** |
|---|---|---|---|---|---|---|
| **Quantity** | 72 | 48 | 48 | 48 | 24 | 48 |
| **Window** | fixed | Sliding | | | | |
| **Quantity** | 48 | 96 | | | | |
| **Door** | Single | Double | | | | |
| **Quantity** | 96 | 24 | | | | |

From the perspective of time, add data property in the IFC-space ontology. Assuming that the pipeline laying of the ward is considered, it takes 15 hours for one person to assemble the ward, 10 hours for the buffer zone, and 15 hours for the bathroom. It can be concluded that the assembly time for one person to transform the project is 1320 hours. Managers can decide how many people to hire by limiting the time limit for renovation.

According to the above conclusions, the venue-to-cabin hospital project can be evaluated by designers and administrators based on time and required materials. If there are multiple alternative sites, the method of this study can provide an overall reference opinion, which has a positive impact on the rapid decision-making of site selection.

## 6.4 Case Results and discussions

The application of automatic layout algorithms to irregular spaces presented in this chapter yields several key insights and areas for improvement. The analysis of the layouts in concave and convex spaces, as well as the practical implementations at Brunel University and the Ningbo Olympic Sports Centre, offers a novel method for optimizing hospital design in unconventional environments.

The layout for the concave polygon was constrained by the need to fit rooms within a non-overlapping boundary. The algorithm effectively placed rooms, color-coded as red for wards and blue for buffer zones and bathrooms, ensuring that the spatial constraints were respected. The main challenge in concave polygons is the irregularity of the boundary, which can lead to inefficient use of space. The layout needs to balance the placement of rooms while maintaining connectivity and access, particularly in ensuring corridors are continuous and accessible. For the future improvements, enhanced algorithms that can better handle the intricacies of concave shapes should be considered, possibly incorporating machine learning to predict and adjust layouts dynamically, could improve efficiency.

The initial layout of the convex polygon resulted in significant space wastage. The restriction that rooms must be arranged in multiples of three exacerbated this issue, leading to an inefficient design. Removing the restriction allowed for a more optimized layout, although this led to deviations from the IFC-space ontology's positional relationships. The trade-off between space efficiency and compliance with ontology constraints needs careful consideration. The improved layout, while better in terms of room utilization, still highlighted the difficulty in maintaining corridor connectivity and buffer zone sharing.

The Michael Sterling and Wilfred Brown buildings provided a unfitted test case for renovation into a cabin hospital. The use of TurtleBot and RTAB-Map facilitated the generation of accurate indoor layouts. However, the manual manoeuvring and handheld Kinect camera mapping revealed limitations in automation and the precision of the mapping process. The mapping process, though effective, highlighted the need for more automated and accurate technologies to reduce manual intervention and errors. The irregular polygon derived from the BIM model posed challenges in maintaining the required spatial relationships for effective hospital operation. The resulting layout, despite its regular ward units, did not fully comply with the IFC-space ontology. Although the results are not satisfactory, this method can eliminate inappropriate site selection schemes in the early stage of design and shorten the trial and error time of designers. Experienced personnel can rely on their years of work experience and intuition to judge whether the site selection is appropriate, but this method at least provides a reliable method for industry novices to eliminate errors.

Further advancements in mapping technologies, like enhanced SLAM algorithms and automated mapping robots, could improve accuracy and reduce manual intervention. Additionally, refining the layout algorithms to better handle irregular shapes and spatial constraints would be beneficial. Autodesk Recap is an up-to-date software application developed by Autodesk for reality capture and 3D scanning workflows. It allows users to process point clouds and 3D models from laser scans and photos, enabling the capture of real-world data using laser scanners and photogrammetry. This data is processed to create accurate 3D models and point clouds, with features for noise reduction, registration of multiple scans, and data alignment. Recap also offers precise measurement and analysis tools within 3D models, making it valuable for architecture, construction, and engineering industries. The software integrates seamlessly with other Autodesk products such as AutoCAD, Revit, and Civil 3D, facilitating smooth workflows between reality capture and design. Additionally, Recap supports cloud-based collaboration, allowing teams to share and review 3D models and point clouds in real-time. It provides tools for visualizing and navigating complex 3D datasets, making it easier to understand and interact with the captured data. Users can annotate and tag specific areas of the 3D models or point clouds, adding notes and metadata for better documentation and communication. Recap supports various export formats, making data transfer to other applications and platforms straightforward. Widely used in industries where accurate 3D modeling and reality capture are essential, such as construction, architecture, civil engineering, and heritage preservation, Autodesk Recap's ability to process and manage large datasets from various sources makes it a powerful tool for professionals working with 3D data.

The sports venue’s conversion into a temporary hospital demonstrated effective use of existing infrastructure. The layout included 72 rooms with a mix of wards, buffer zones, and bathrooms. The strategic location and connectivity of the venue can be ensured smooth operation during an epidemic scenario. The conversion showcased the importance of integrating traffic management and spatial arrangement for emergency operations.

The selected buildings at Brunel University—the Michael Sterling Building and the Wilfred Brown Building—were initially considered for conversion into a temporary cabin hospital due to their modern design, good lighting, and flexible layout. These buildings feature multiple entrances and exits, which are advantageous for emergency evacuations, and their open ground floor spaces allow for easy equipment transportation. However, despite these benefits, the generated layouts using advanced mapping technology (TurtleBot3 with Kinect and RTAB-MAP) revealed significant limitations. The irregular shape of the space and the constraints of the layout algorithm resulted in fewer than 10 rooms, which was insufficient to meet the needs of a COVID-19 cabin hospital. This outcome demonstrated that while the site had some favorable features, it was ultimately unsuitable for large-scale conversion. Nevertheless, the exercise proved valuable in showcasing how early-stage feasibility checks, guided by ontology-based constraints, can help designers quickly identify and eliminate impractical options.

In contrast, the Ningbo Olympic Sports Centre Comprehensive Training Hall emerged as a highly suitable candidate for conversion into a temporary epidemic hospital. The venue's large, open spaces, excellent infrastructure (including natural lighting, ventilation, and reliable utilities), and adaptable smaller surrounding rooms made it ideal for accommodating medical wards, staff preparation areas, and equipment storage. The layout optimization process successfully generated 72 rooms, including 48 wards, 24 buffer zones, and 24 bathrooms, meeting the functional requirements for a cabin hospital. Additionally, the venue's strategic location, with strong road connectivity and public transport access, ensured efficient logistics for medical supplies and personnel. The project also benefited from precise calculations of required facilities (e.g., beds, sinks, toilets) and construction timelines, enabling managers to make informed decisions about labor and resource allocation.

The comparison between these two locations underscores the importance of scalability, infrastructure, and spatial regularity in selecting sites for temporary medical facilities. While the Brunel University buildings highlighted the challenges of irregular layouts and limited space, the Ningbo sports venue demonstrated how large, adaptable spaces with robust infrastructure can be effectively repurposed for emergency healthcare needs. Both cases validated the study's methodology, showing its effectiveness in evaluating site suitability and optimizing layouts through iterative design and constraint-based validation. Ultimately, the Ningbo venue stood out as a practical and efficient choice, while the Brunel case served as a useful example of how early-stage assessments can prevent costly and impractical design commitments.

By addressing the challenges in these cases, it can be demonstrated this method to developing more effective and adaptable solutions for emergency medical facilities in unconventional environments, increasing the resilience and flexibility of hospital design. During the implementation process, IFC-space integrated other ontologies and added more detailed data attributes, which demonstrated the flexibility and extensibility of the IFC-space ontology proposed in this research method and enabled data reuse, which is also the focus of the ontology-based automatic layout method.

# Chapter 7 Conclusions and Future Work

## 7.1 Conclusions

This research effectively tackles the key challenges outlined in the introduction chapter by proposing an ontology-based automatic layout design approach integrated with Building Information Modeling. The first problem, the complexity and inefficiency of traditional cabin hospital design, is addressed through automation. Rather than relying on manual CAD drafting and large teams of designers, the ontology-driven system generates optimized layouts using predefined rules and constraints. This reduces human error, accelerates the design process, and ensures consistency across different projects.

The second problem, cost and time inefficiencies, is mitigated by the dynamic nature of the ontology-BIM framework. Since cabin hospitals are temporary structures, the system accounts for modularity and reusability of components, such as medical equipment and prefabricated building elements. By integrating cost constraints into the design algorithm, the method helps minimize waste and optimize resource allocation. Additionally, real-time adjustments in BIM allow for rapid modifications, which is crucial in emergency scenarios where design requirements may change abruptly.

The third problem, heterogeneous and fragmented knowledge from multiple stakeholders, is resolved through the structured nature of ontologies. By formalizing medical, architectural, and logistical requirements into a unified knowledge model, the system ensures that all stakeholders (e.g., architects, engineers, healthcare providers) work from a single source of truth. This eliminates inconsistencies, reduces redundant data, and improves collaboration.

Building upon foundational concepts, this research investigates the practical application and significant benefits of integrating ontology-based automatic layout design within cabin hospitals. Cabin hospitals, which are temporary or rapidly deployable healthcare facilities, necessitate precise and efficient design due to their specialized medical functions and the urgency of their deployment in emergency scenarios. The adoption of Building Information Modeling (BIM) as a core tool is pivotal in this context, as it provides a sophisticated platform for merging various aspects of building design into a unified and interactive digital model.

Building Information Modeling (BIM) serves as an advanced digital framework that allows for the detailed visualization and management of building projects. In the context of cabin hospitals, BIM offers several advantages:

1) Integrated Visualization: BIM enables a comprehensive 3D visualization of the hospital layout, including spatial arrangements, medical equipment placements, and functional areas. This holistic view supports better planning and coordination among architects, engineers, and healthcare professionals.
2) Data Integration: BIM integrates diverse data types, such as architectural plans, structural elements, and facilities, into a cohesive model. This integration is crucial for cabin hospitals, where accurate coordination between different systems is essential for functionality and compliance.
3) Data Reusability and Extensibility: The IFC format serves as an open, standardized data schema specifically designed to enable seamless interoperability between AEC-FM software platforms. Developed to overcome proprietary data silos, IFC provides a neutral structure that ensures consistent data exchange across the project lifecycle. By structuring building information in a unified model, it allows multidisciplinary teams, including architects, engineers, and contractors, to collaboratively access, modify, and extend project data without compatibility barriers. This interoperability not only streamlines workflows but also preserves data integrity for reuse in downstream applications.

While BIM provides a robust digital platform, its effectiveness can be significantly enhanced by incorporating ontology-based approaches. Ontologies are formal representations of knowledge that define the relationships between concepts within a specific domain. Here's how ontology-based methods augment BIM for cabin hospitals:

1) Semantic Enrichment: Ontologies enhance the semantic depth of BIM models by providing detailed definitions and relationships for various building components and their functions. This semantic enrichment allows for more accurate and meaningful data representation, which is crucial for specialized medical environments.
2) Knowledge Integration: This study develops an IFC-Space Ontology by integrating multidisciplinary knowledge (architecture, medicine, AI) through a two-phase process: (a) semantic mapping to align concepts and relationships, and (b) harmonization to merge ontologies using tools like Protégé. The resulting approach encodes design rules, medical workflows, and spatial constraints via logic programming (OWL/RDF), enabling automated layout optimization and inconsistency detection. Validated through empirical testing, the ontology enhances interoperability, adaptability to medical needs, and decision-making accuracy, offering a scalable model for hospital design and beyond.

One of the key advantages of an ontology-based approach is its contribution to the flexibility and adaptability of cabin hospital designs. The approach in this research can dynamically accommodate changes:

1) Dynamic Layout Adjustments: In emergency situations, the needs and constraints of a cabin hospital may evolve rapidly. An ontology-based design allows for quick adjustments to the layout by providing a flexible structure that can adapt to new requirements or constraints, such as changes medical equipment. The results can be obtained quickly based on the automatic layout. Combined with the verification of the ontology, effective feedback can be obtained to adjust the constraints or change the parameters to optimize the layout design. Even more, this feedback can support better decision-making by allowing stakeholders to understand and analyze the implications of design choices within the context of

medical needs. This leads to more informed decisions and fewer errors.

2) Scenario Planning: Ontologies enable scenario planning by incorporating different design possibilities and their implications. This allows for the rapid development of alternative layouts that can be evaluated and implemented as needed. Furthermore, this study uses the layout of the COVID-19 cabin hospital as knowledge input and uses the algorithm to achieve automatic layout. This proves that ontology-based automatic layout is feasible. If other knowledge inputs are used to change the constraints, this method can also be applied to other scenarios.For example, in response to public safety emergencies such as tsunamis and earthquakes, by changing the constraints on virus transmission control in the space entity (removing the buffer zone, changing access restrictions), the automated layout of medical facilities corresponding to the profile can be carried out.

The research also highlights the potential for automating various aspects of the cabin hospital design process using ontology-based methods:

1) Automated Layout Generation: Leveraging ontologies, algorithms can be developed to automatically generate layout options based on predefined criteria. For example, an algorithm could create different configurations for patient rooms, medical stations, and support areas while adhering to regulatory standards.
2) Evaluation and Optimization: Automated systems can evaluate and optimize layout options based on factors such as space utilization, accessibility, and compliance with health regulations. This reduces the manual effort required from designers and ensures that the final layout is both functional and compliant.
3) Design Consistency: Automation helps maintain consistency across different design iterations and projects by applying the same ontological framework. This consistency is essential for ensuring that all cabin

hospital designs meet the necessary standards and requirements.

This research presents a groundbreaking ontology-based framework for automated cabin hospital design, offering transformative advancements in emergency healthcare infrastructure and digital construction. By automating layout generation, the system slashes design time from days to mere hours—a critical advantage for rapid pandemic or disaster response—while its dynamic adjustment capabilities enable real-time adaptations to evolving medical needs, such as equipment reconfigurations.

A pivotal innovation lies in its ontology-driven standardization, which synthesizes multidisciplinary stakeholder knowledge into a unified model, bridging architects, engineers, and clinicians while ensuring strict compliance with medical and safety regulations. The integration of IFC-based interoperability further streamlines data exchange across AEC-FM platforms, enhancing collaboration and reducing errors.

Beyond efficiency gains, the framework embeds domain-specific expertise directly into the digital workflow, ensuring cabin hospitals are not only rapidly deployable but also optimized for operational effectiveness in emergencies. Its exceptional flexibility allows seamless scalability to diverse crisis scenarios—from earthquake shelters to refugee housing—through modular constraint adjustments, with demonstrated COVID-19 applications underscoring its potential for global standardization in temporary healthcare design.

The research also propels intelligent construction forward by merging BIM's advanced modeling with ontology-driven semantics, introducing new paradigms for error detection, safety optimization, and knowledge reuse in permanent structures. By harmonizing medical requirements with computational design and construction practices, this work not only redefines emergency facility planning but also lays the foundation for adaptive, intelligence-driven infrastructure solutions across crisis management and architectural design.

## 7.2 Limitations and future work

Despite the substantial promise of integrating ontology-based methods with

Building Information Modeling (BIM) in the design of cabin hospitals, several significant challenges and limitations persist. Addressing these issues is crucial for realizing the full potential of these advanced technologies and achieving widespread adoption across the construction industry.

A major challenge is ensuring interoperability between various BIM tools and platforms. While ontologies can significantly enhance data integration and the semantic richness of BIM models, achieving seamless communication across different software systems remains complex. Each BIM tool often employs its own data structures, standards, and protocols, which can lead to compatibility issues and hinder effective data exchange. To mitigate these challenges, there is a pressing need for standardized protocols and frameworks that can facilitate smoother interoperability. This involves developing universal standards for data formats, communication protocols, and integration methods. Such standardization requires extensive collaboration among software developers, industry stakeholders, and standardization bodies to create and implement common guidelines that enable different systems to work together seamlessly.

Resistance to adopting new technologies is another significant limitation. The construction industry is traditionally conservative, and introducing ontology-based BIM systems may encounter resistance due to a lack of familiarity and perceived risk. Convincing stakeholders to invest in these advanced systems requires compelling evidence of their benefits. Demonstrating the advantages of ontology-based BIM, such as cost savings, improved efficiency, and enhanced compliance, can help overcome skepticism and resistance. Additionally, the high initial costs associated with these technologies, including the expense of software development, training, and maintenance, can be a substantial barrier, particularly for smaller firms with limited budgets. To address this, it is essential to provide a clear understanding of the long-term return on investment (ROI) and to explore ways to reduce initial implementation costs through scalable solutions and financial incentives for early adopters.

Maintaining up-to-date ontologies presents another critical challenge. As the healthcare industry evolves with new medical technologies, treatments, and regulations, ontologies must be continuously updated to reflect these changes accurately. This ongoing maintenance is essential to ensure that the ontological framework remains relevant and precise. However, maintaining up-to-date ontologies requires a coordinated effort among industry experts, IT professionals, and regulatory bodies. This collaborative effort ensures that the ontologies incorporate the latest advancements and comply with updated standards and regulations.

Looking forward, future research and development should focus on several key areas to address these challenges. Simplifying the integration process between different BIM tools and reducing implementation costs are crucial steps for broader adoption. Developing standardized protocols and creating user-friendly tools and interfaces can make ontology-based BIM systems more accessible to firms of varying sizes and levels of expertise. Comprehensive training programs and support resources will also be essential in helping industry professionals effectively utilize these technologies.

Expanding research to include a wider range of healthcare facilities and different types of construction projects will help validate the methodology and demonstrate its versatility. Conducting validation studies in diverse real-world scenarios can provide valuable insights into the performance and impact of ontology-based BIM systems, thereby building confidence and support for their broader adoption. Additionally, fostering a culture of innovation and openness within the construction industry is critical. Promoting the benefits of advanced methodologies and encouraging a willingness to explore new technologies can drive progress and improve industry practices.

Collaborative efforts among academia, industry, and regulatory bodies will be essential in advancing ontology-based BIM systems. Such collaborations can drive innovation, establish standards, and address common challenges, ultimately contributing to the successful integration and utilization of these technologies.

In summary, while integrating ontology-based methods with BIM offers transformative potential for the design and construction of cabin hospitals, overcoming the identified challenges is essential for achieving widespread adoption.

By focusing on simplifying integration, reducing costs, developing user-friendly tools, and fostering a culture of innovation and collaboration, the construction industry can leverage these advanced methodologies to create more efficient, flexible, and compliant healthcare facilities. This progress will ultimately lead to improved patient care and operational efficiency, particularly in emergency healthcare settings where rapid and effective design solutions are crucial.

# REFERENCES


Asia Weekly: Beijing built Xiaotangshan Hospital in 8 days. (2020) Available at: (Accessed: .

BuildingSMART: Linked Data Working Group . (2016) Available at: (Accessed: .

Extensible Markup Language (XML). (2016) Available at: (Accessed: .

Introduction to: RDF vs XML . (2016) Available at: (Accessed: .

W3C SEMANTIC WEB. (2016) Available at: (Accessed: .

BuildingSMART Proposed Recommendation: EXPRESS-to-OWL conversion routine. (2015) Available at: (Accessed: .

W3C RDF Working Group, Resource Description Framework (RDF) . (2015) Available at: http://www.w3.org/RDF/ (Accessed: .

W3C Standard, Ontologies . (2015) Available at: (Accessed: .

About Advanced Work Packaging (AWP) . (2014) Available at: http://www.coaa.ab.ca/construction/AWPWFP/AWPWFPOverviewandDefinitions.aspx (Accessed: .

Ajam, M., Alshawi, M. and Mezher, T. (2010) 'Augmented process model for e-tendering: Towards integrating object models with document management systems', Automation in Construction, 19(6), pp. 762-778.

Akaichi, J. and Mhadhbi, L. (2016) 'A clinical decision support system: Ontology-driven approach for effective emergency management'Improving health management through clinical decision support systems IGI Global, pp. 270-294.

Akbarnezhad, A., Ong, K.C.G. and Chandra, L.R. (2014) 'Economic and environmental assessment of deconstruction strategies using building information modeling', Automation in Construction, 37, pp. 131-144.

Alshawi, M. and Ingirige, B. (2003) 'Web-enabled project management: an emerging paradigm in construction', Automation in Construction, 12(4), pp. 349-364.

Antoniou, G. and Van Harmelen, F. (2004) A semantic web primer. MIT press.

Anumba, C.J., Issa, R.R., Pan, J. and Mutis, I. (2008) 'Ontology-based information and knowledge management in construction', Construction Innovation, .

Becerik-Gerber, B., Jazizadeh, F., Li, N. and Calis, G. (2012) 'Application areas and data requirements for BIM-enabled facilities management', Journal of Construction Engineering and Management, 138(3), pp. 431-442.

Becerik-Gerber, B. and Rice, S. (2010) 'The perceived value of building information modeling in the US building industry', Journal of Information Technology in Construction (ITcon), 15(15), pp. 185-201.

Beetz, J., Van Leeuwen, J. and De Vries, B. (2009) 'IfcOWL: A case of transforming EXPRESS schemas into ontologies', Ai Edam, 23(1), pp. 89-101.

Bernstein, H.M., Jones, S.A., Russo, M.A., Laquidara-Carr, D., Taylor, W., Ramos, J., Healy, M., Lorenz, A., Fujishima, H. and Fitch, E. (2012) 'The business value of BIM in North America', SmartMarket Report, 64.

Bonsma, P., Bonsma, I., Ziri, A.E., Parenti, S., Lerones, P.M., Hernández, J.L., Maietti, F., Medici, M., Turillazzi, B. and Iadanza, E. (2016) Inception standard for heritage bim models. Springer, pp. 590.

Brambilla, A., Brusamolin, E., Johnson, A.A., Scullica, F. and Capolongo, S. (2023) 'Lessons from the first wave of COVID-19 in Italy: A collection of design strategies to face pandemic situations in healthcare facilities', Journal of Emergency Management, 21(7), pp. 167-184.

Caneparo, L. (2022a) 'Semantic knowledge in generation of 3D layouts for decision-making', Automation in Construction, 134, pp. 104012 Available at: 10.1016/j.autcon.2021.104012.

Caneparo, L. (2022b) 'Semantic knowledge in generation of 3D layouts for decision-making', Automation in Construction, 134, pp. 104012 Available at: 10.1016/j.autcon.2021.104012.

Cao, D., Li, H. and Wang, G. (2014) 'Impacts of isomorphic pressures on BIM adoption in construction projects', Journal of Construction Engineering and Management, 140(12), pp. 04014056.

Chan, A.P.C., Scott, D. and Chan, A.P.L. (2004) 'Factors affecting the success of a construction project', Journal of Construction Engineering and Management, 130(1), pp. 153-155.

Chen, C., Chacón Vega, R.J. and Kong, T.L. (2021) 'Using genetic algorithm to automate the generation of an open-plan office layout', International Journal of Architectural Computing, 19(3), pp. 449-465.

Cicmil, S. and Marshall, D. (2005) 'Insights into collaboration at the project level: complexity, social interaction and procurement mechanisms', Building Research & Information, 33(6), pp. 523-535.

Cubukcuoglu, C., Nourian, P., Tasgetiren, M.F., Sariyildiz, I.S. and Azadi, S. (2021) 'Hospital layout design renovation as a Quadratic Assignment Problem with geodesic distances', Journal of Building Engineering, 44, pp. 102952 Available at: 10.1016/j.jobe.2021.102952.

Curry, E., O'Donnell, J., Corry, E., Hasan, S., Keane, M. and O'Riain, S. (2013) 'Linking building data in the cloud: Integrating cross-domain building data using linked data', Advanced Engineering Informatics, 27(2), pp. 206-219 Available at: 10.1016/j.aei.2012.10.003.

De Farias, T.M., Roxin, A. and Nicolle, C. (2015) 'IfcWoD, semantically adapting IFC model relations into OWL properties', arXiv preprint arXiv:1511.03897, .

Eastman, C.M., Jeong, Y., Sacks, R. and Kaner, I. (2010) 'Exchange model and exchange object concepts for implementation of national BIM standards', Journal of Computing in Civil Engineering, 24(1), pp. 25-34.

Eastman, C.M., Eastman, C., Teicholz, P., Sacks, R. and Liston, K. (2011) BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors. John Wiley & Sons.

Fernández-López, M., Gómez-Pérez, A. and Juristo, N. (1997a) 'Methontology: from ontological art towards ontological engineering', .

Fernández-López, M., Gómez-Pérez, A. and Juristo, N. (1997b) 'Methontology: from ontological art towards ontological engineering', .

Froese, T. (2003) 'Future directions for IFC-based interoperability', Journal of Information Technology in Construction (ITcon), 8(17), pp. 231-246.

Gray, M., Gray, J., Teo, M., Chi, S. and Cheung, F. (2013) Building Information Modeling. pp. 5.

Gruber, T.R. (1993) 'A translation approach to portable ontology specifications', Knowledge acquisition, 5(2), pp. 199-220.

Gruber, T.R. (1995) 'Toward principles for the design of ontologies used for knowledge sharing?', International Journal of Human-Computer Studies, 43(5), pp. 907-928 Available at: 10.1006/ijhc.1995.1081.

Gu, N. and London, K. (2010) 'Understanding and facilitating BIM adoption in the AEC industry', Automation in Construction, 19(8), pp. 988-999.

Guizzardi, G. (2005) 'Ontological foundations for structural conceptual models', .

Hamadah, H., Alahmad, B., Behbehani, M., Al-Youha, S., Almazeedi, S., Al-Haddad, M., Jamal, M.H. and Al-Sabah, S. (2020) 'COVID-19 clinical outcomes and nationality: results from a Nationwide registry in Kuwait', BMC Public Health, 20, pp. 1-9.

Hendler, J., Gandon, F. and Allemang, D. (2020) Semantic Web for the Working Ontologist: Effective Modeling for Linked Data, RDFS, and OWL. Morgan & Claypool.

HKS Architects and Arup (2021) The Pandemic-Resilient Hospital: How Design Can Help Facilities Stay Operational and Safe. Available at:(Accessed: .

Jia, Z., Nourian, P., Luscuere, P. and Wagenaar, C. (2023) 'Spatial decision support systems for hospital layout design: A review', Journal of Building Engineering, 67, pp. 106042 Available at: 10.1016/j.jobe.2023.106042.

Jung, Y. and Joo, M. (2011) 'Building information modelling (BIM) framework for practical implementation', Automation in Construction, 20(2), pp. 126-133.

Kang, T. and Choi, H. (2015a) 'BIM perspective definition metadata for interworking facility management data', Advanced Engineering Informatics, 29(4), pp. 958-970 Available at: 10.1016/j.aei.2015.09.004.

Kang, T. and Choi, H. (2015b) 'BIM perspective definition metadata for interworking facility management data', Advanced Engineering Informatics, 29(4), pp. 958-970.

Karshenas, S. and Niknam, M. (2013) 'Ontology-based building information modeling'Computing in Civil Engineering (2013), pp. 476-483.

Knublauch, H., Oberle, D., Tetlow, P., Wallace, E., Pan, J.Z. and Uschold, M. (2006)

'A semantic web primer for object-oriented software developers', W3c working group note, W3C, .
Lee, G., Lee, J. and Jones, S.A. (2012) 'The business value of BIM in South Korea', SmartMarket Rep, .
Lucas, J., Bulbul, T. and Thabet, W. (2013) 'An object-oriented model to support healthcare facility information management', Automation in Construction, 31, pp. 281-291.
MA Longxin, TANG Jie, LIN Jingsheng and (2022) ' Construction and application of clinical database in large shelter hospital', Journal of Diagnostics Concepts & Practice, 21(02), pp. 205.
Maunula, A. and Smeds, R. (2008) The implementation of building information modeling (BIM): A process perspective. Teknillinen korkeakoulu.
Maurer, I. (2010) 'How to build trust in inter-organizational projects: The impact of project staffing and project rewards on the formation of trust, knowledge acquisition and product innovation', International Journal of Project Management, 28(7), pp. 629-637.
Michalek, J., Choudhary, R. and Papalambros, P. (2002) 'Architectural layout design optimization', Engineering optimization, 34(5), pp. 461-484.
Naik, S.B. and Kallurkar, S. (2016) 'A literature review on efficient plant layout design', International Journal of Industrial Engineering, 7(2), pp. 43-51.
Nicolle, C. and Cruz, C. (2010) Semantic building information model and multimedia for facility management. Springer, pp. 14.
Niknam, M. and Karshenas, S. (2015) 'Sustainable design of buildings using semantic BIM and semantic web services', Procedia Engineering, 118, pp. 909-917.
Niknam, M. and Karshenas, S. (2014) 'A social networking website for AEC projects'Computing in Civil and Building Engineering (2014), pp. 2208-2215.
Nisbet, N. and Dinesen, B. (2010) 'Constructing the business case: Building information modelling', London, UK: British Standards Institution, .
Nitithamyong, P. and Skibniewski, M.J. (2004) 'Web-based construction project management systems: how to make them successful?', Automation in Construction,

13(4), pp. 491-506.

Noy, N.F. and McGuinness, D.L. (2001) 'Ontology development 101: A guide to creating your first ontology', .

Parsia, Y. and Tamyez, P.F. (2018) 'Role of healthcare-facilities layout design, healing architecture, on quality of services', International Journal of Civil Engineering and Technology, 9(4), pp. 598-601.

Pauwels, P., de Farias, T.M., Zhang, C., Roxin, A., Beetz, J., De Roo, J. and Nicolle, C. (2017) 'A performance benchmark over semantic rule checking approaches in construction industry', Advanced Engineering Informatics, 33, pp. 68-88 Available at: https://doi.org/10.1016/j.aei.2017.05.001.

Pauwels, P., De Meyer, R. and Van Campenhout, J. (2010) Interoperability for the design and construction industry through semantic web technology. Springer, pp. 143.

Pauwels, P., Krijnen, T., Terkaj, W. and Beetz, J. (2017) 'Enhancing the ifcOWL ontology with an alternative representation for geometric data', Automation in Construction, 80, pp. 77-94 Available at: https://doi.org/10.1016/j.autcon.2017.03.001.

Pauwels, P. and Roxin, A. (2017) 'SimpleBIM: From full ifcOWL graphs to simplified building graphs'Ework and ebusiness in architecture, engineering and construction CRC Press, pp. 11-18.

Pauwels, P. and Terkaj, W. (2016) 'EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology', Automation in Construction, 63, pp. 100-133 Available at: https://doi.org/10.1016/j.autcon.2015.12.003.

Pauwels, P., Terkaj, W., Krijnen, T. and Beetz, J. (2015) Coping with lists in the ifcOWL ontology. pp. 113.

Pauwels, P., Törmä, S., Beetz, J., Weise, M. and Liebich, T. (2015) 'Linked Data in Architecture and Construction', Automation in Construction, 57, pp. 175-177 Available at: https://doi.org/10.1016/j.autcon.2015.06.007.

Pauwels, P., Zhang, S. and Lee, Y. (2017) 'Semantic web technologies in AEC industry: A literature overview', Automation in Construction, 73, pp. 145-165 Available at: https://doi.org/10.1016/j.autcon.2016.10.003.

Penttilä, H., Rajala, M. and Freese, S. (2007) 'Building information modelling of

modern historic buildings', .
Rahbar, M., Mahdavinejad, M., Markazi, A.H.D. and Bemanian, M. (2022a) 'Architectural layout design through deep learning and agent-based modeling: A hybrid approach', Journal of Building Engineering, 47, pp. 103822 Available at: 10.1016/j.jobe.2021.103822.
Rahbar, M., Mahdavinejad, M., Markazi, A.H. and Bemanian, M. (2022b) 'Architectural layout design through deep learning and agent-based modeling: A hybrid approach', Journal of Building Engineering, 47, pp. 103822.
Reshetnikov, V., Mitrokhin, O., Shepetovskaya, N., Belova, E. and Jakovljevic, M. (2020) 'Organizational measures aiming to combat COVID-19 in the Russian Federation: the first experience', Expert Review of Pharmacoeconomics & Outcomes Research, 20(6), pp. 571-576.
RICS, R. (2013) 'BIM survey results', The Royal Institute of Charted Surveyors, .
Sabol, L. (2008) 'Building information modeling & facility management', IFMA World workplace, , pp. 2-13.
Sears, S.K., Clough, R.H. and Sears, G.A. (2008) Construction project management: a practical guide to field construction management. John Wiley & Sons.
Shu-Hsien Liao, (2005) 'Expert system methodologies and applications—a decade review from 1995 to 2004', Expert Systems with Applications, 28(1), pp. 93-103 Available at: 10.1016/j.eswa.2004.08.003.
Succar, B. (2009) 'Building information modelling framework: A research and delivery foundation for industry stakeholders', Automation in Construction, 18(3), pp. 357-375.
Sugumaran, V. and Storey, V.C. (2006) 'The role of domain ontologies in database design: An ontology management and conceptual modeling environment', ACM Transactions on Database Systems (TODS), 31(3), pp. 1064-1094.
Sydora, C. and Stroulia, E. (2020) 'Rule-based compliance checking and generative design for building interiors using BIM', Automation in Construction, 120, pp. 103368.
Taxén, L. and Lilliesköld, J. (2008) 'Images as action instruments in complex projects', International Journal of Project Management, 26(5), pp. 527-536.

Terkaj, W. and Pauwels, P. (2017) A method to generate a modular ifcOWL ontology.

Terkaj, W. and Šojić, A. (2015) 'Ontology-based representation of IFC EXPRESS rules: An enhancement of the ifcOWL ontology', Automation in Construction, 57, pp. 188-201 Available at: https://doi.org/10.1016/j.autcon.2015.04.010.

Underwood, J. and Isikdag, U. (2011) 'Emerging technologies for BIM 2.0', Construction Innovation, .

Uschold, M. and King, M. (1995) Towards a methodology for building ontologies. Citeseer.

Uschold, M. and Gruninger, M. (1996) 'Ontologies: Principles, methods and applications', The knowledge engineering review, 11(2), pp. 93-136.

Wang, L., Liu, J., Zeng, Y., Cheng, G., Hu, H., Hu, J. and Huang, X. (2023) 'Automated building layout generation using deep learning and graph algorithms', Automation in Construction, 154, pp. 105036 Available at: 10.1016/j.autcon.2023.105036.

Wassouf, Z., Egger, M., Neuberg, F., van Treeck, C. and Rank, E. (2006) 'Produktmodell basierte Simulation des Ressourcenbedarfs von Bauwerken', Bauingenieur, 81.

Waterhouse, R. and Philp, D. (2013) 'National BIM Report 2013', London, UK: RIBA Enterprise Ltd, .

Whitehead, B. and Eldars, M.Z. (1965) 'The planning of single-storey layouts', Building Science, 1(2), pp. 127-139.

Williams, T.M. (2013) Managing and modelling complex projects. Springer.

Wise, J. (2021) 'Covid-19: London's Nightingale Hospital will reopen for non-covid cases', BMJ, 372, pp. n15 Available at: 10.1136/bmj.n15.

Wong, S.S.Y. and Chan, K.C.C. (2009) 'EvoArch: An evolutionary algorithm for architectural layout design', Computer-Aided Design, 41(9), pp. 649-667 Available at: 10.1016/j.cad.2009.04.005.

Yong, L. and Chibiao, H. (2022) 'A generative design method of building layout generated by path', Applied Mathematics and Nonlinear Sciences, 7(2), pp. 825-848.

Zampetakis, S., Tzitzikas, Y., Leonidis, A. and Kotzinos, D. (2012a) 'Star-like auto-configurable layouts of variable radius for visualizing and exploring RDF/S ontologies', Journal of Visual Languages & Computing, 23(3), pp. 137-153 Available at: 10.1016/j.jvlc.2012.01.002.

Zampetakis, S., Tzitzikas, Y., Leonidis, A. and Kotzinos, D. (2012b) 'Star-like auto-configurable layouts of variable radius for visualizing and exploring RDF/S ontologies', Journal of Visual Languages & Computing, 23(3), pp. 137-153 Available at: 10.1016/j.jvlc.2012.01.002.

Zangrillo, A., Beretta, L., Silvani, P., Colombo, S., Scandroglio, A.M., Dell'Acqua, A., Fominskiy, E., Landoni, G., Monti, G. and Azzolini, M.L. (2020) 'Fast reshaping of intensive care unit facilities in a large metropolitan hospital in Milan, Italy: facing the COVID-19 pandemic emergency', Critical care and resuscitation, 22(2), pp. 91-94.

Zhang, D., Zhang, J., Guo, J. and Xiong, H. (2019) 'A semantic and social approach for real-time green building rating in BIM-based design', Sustainability, 11(14), pp. 3973.

Zhang, S., Teizer, J., Lee, J., Eastman, C.M. and Venugopal, M. (2013) 'Building Information Modeling (BIM) and Safety: Automatic Safety Checking of Construction Models and Schedules', Automation in Construction, 29, pp. 183-195 Available at: 10.1016/j.autcon.2012.05.006.

Zhao, Y., Chen, G., Liao, C. and Shen, X. (2016) Towards ontology-based program analysis. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, .

Zhipeng, Z., Goh, Y.M. and Lijun, S. (2016) 'Overview and Analysis of Ontology Studies Supporting Development of the Construction Industry', Journal of Computing in Civil Engineering, 30(6), pp. 04016026 Available at: 10.1061/(ASCE)CP.1943-5487.0000594.

Zhong, B., Wu, H., Li, H., Sepasgozar, S., Luo, H. and He, L. (2019) 'A scientometric analysis and critical review of construction related ontology research', Automation in Construction, 101, pp. 17-31 Available at: https://doi.org/10.1016/j.autcon.2018.12.013.

Zhou, M., Chen, Y., Su, X. and An, L. (2020) Rapid construction and advanced technology for a Covid-19 field hospital in Wuhan, China. Thomas Telford Ltd, pp. 29.

# Appendix

## Cabin Hospital Ontology


```
<rdf:RDF xmlns="http://www.w3.org/2002/07/owl#" xmlns:ont="http://www.co-
ode.org/ontologies/ont.owl#" xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:cabin="http://example.org/covid19-cabin#"
xml:base="http://www.w3.org/2002/07/owl">
<Ontology/>
<AnnotationProperty rdf:about="http://www.co-
ode.org/ontologies/ont.owl#hasName">
<rdfs:subPropertyOf rdf:resource="http://www.w3.org/2000/01/rdf-
schema#comment"/>
</AnnotationProperty>
<ObjectProperty rdf:about="http://example.org/covid19-cabin#adjacentTo"/>
<ObjectProperty rdf:about="http://example.org/covid19-cabin#contains"/>
<ObjectProperty rdf:about="http://example.org/covid19-cabin#reachableFrom"/>
<ObjectProperty rdf:about="http://www.co-ode.org/ontologies/ont.owl#attachTo"/>
<DatatypeProperty rdf:about="http://example.org/covid19-cabin#length">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#decimal"/>
</DatatypeProperty>
<DatatypeProperty rdf:about="http://example.org/covid19-cabin#width">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#decimal"/>
</DatatypeProperty>
<DatatypeProperty rdf:about="http://www.co-
ode.org/ontologies/ont.owl#Assemblytime">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#decimal"/>
```

```
</DatatypeProperty>
<DatatypeProperty rdf:about="http://www.co-
ode.org/ontologies/ont.owl#PurchasePrice">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#decimal"/>
</DatatypeProperty>
<DatatypeProperty rdf:about="http://www.co-
ode.org/ontologies/ont.owl#hasGUID"/>
<DatatypeProperty rdf:about="http://www.co-ode.org/ontologies/ont.owl#hight">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#decimal"/>
</DatatypeProperty>
<Class rdf:about="http://example.org/covid19-cabin#Bathroom">
<rdfs:subClassOf rdf:resource="http://www.co-
ode.org/ontologies/ont.owl#SpaceArea"/>
</Class>
<Class rdf:about="http://example.org/covid19-cabin#Bed">
<rdfs:subClassOf rdf:resource="http://www.co-
ode.org/ontologies/ont.owl#GeneralFacilities"/>
</Class>
<Class rdf:about="http://example.org/covid19-cabin#BufferZone">
<rdfs:subClassOf rdf:resource="http://www.co-
ode.org/ontologies/ont.owl#SpaceArea"/>
</Class>
<Class rdf:about="http://example.org/covid19-cabin#BuildingElement"/>
<Class rdf:about="http://example.org/covid19-cabin#Corridor">
<rdfs:subClassOf rdf:resource="http://www.co-
ode.org/ontologies/ont.owl#SpaceArea"/>
</Class>
<Class rdf:about="http://example.org/covid19-cabin#Door">
<rdfs:subClassOf rdf:resource="http://example.org/covid19-
cabin#BuildingElement"/>
```

```
</Class>
<Class rdf:about="http://example.org/covid19-cabin#Facility"/>
<Class rdf:about="http://example.org/covid19-cabin#Shower">
<rdfs:subClassOf rdf:resource="http://www.co-
ode.org/ontologies/ont.owl#GeneralFacilities"/>
</Class>
<Class rdf:about="http://example.org/covid19-cabin#Sink">
<rdfs:subClassOf rdf:resource="http://www.co-
ode.org/ontologies/ont.owl#GeneralFacilities"/>
</Class>
<Class rdf:about="http://example.org/covid19-cabin#Toilet">
<rdfs:subClassOf rdf:resource="http://www.co-
ode.org/ontologies/ont.owl#GeneralFacilities"/>
</Class>
<Class rdf:about="http://example.org/covid19-cabin#Wall">
<rdfs:subClassOf rdf:resource="http://example.org/covid19-
cabin#BuildingElement"/>
</Class>
<Class rdf:about="http://example.org/covid19-cabin#Ward">
<rdfs:subClassOf rdf:resource="http://www.co-
ode.org/ontologies/ont.owl#SpaceArea"/>
</Class>
<Class rdf:about="http://example.org/covid19-cabin#WardA">
<rdfs:subClassOf rdf:resource="http://example.org/covid19-cabin#Ward"/>
</Class>
<Class rdf:about="http://example.org/covid19-cabin#WardB">
<rdfs:subClassOf rdf:resource="http://example.org/covid19-cabin#Ward"/>
</Class>
<Class rdf:about="http://example.org/covid19-cabin#Window">
<rdfs:subClassOf rdf:resource="http://example.org/covid19-
```

```
cabin#BuildingElement"/>
</Class>
<Class rdf:about="http://www.co-ode.org/ontologies/ont.owl#CallingDevice">
<rdfs:subClassOf rdf:resource="http://www.co-ode.org/ontologies/ont.owl#MedicalEquipment"/>
</Class>
<Class rdf:about="http://www.co-ode.org/ontologies/ont.owl#GeneralFacilities">
<rdfs:subClassOf rdf:resource="http://example.org/covid19-cabin#Facility"/>
</Class>
<Class rdf:about="http://www.co-ode.org/ontologies/ont.owl#Light">
<rdfs:subClassOf rdf:resource="http://www.co-ode.org/ontologies/ont.owl#GeneralFacilities"/>
</Class>
<Class rdf:about="http://www.co-ode.org/ontologies/ont.owl#MedicalContaminantCollectingBin">
<rdfs:subClassOf rdf:resource="http://www.co-ode.org/ontologies/ont.owl#MedicalEquipment"/>
</Class>
<Class rdf:about="http://www.co-ode.org/ontologies/ont.owl#MedicalEquipment">
<rdfs:subClassOf rdf:resource="http://example.org/covid19-cabin#Facility"/>
</Class>
<Class rdf:about="http://www.co-ode.org/ontologies/ont.owl#MedicalMonitor">
<rdfs:subClassOf rdf:resource="http://www.co-ode.org/ontologies/ont.owl#MedicalEquipment"/>
</Class>
<Class rdf:about="http://www.co-ode.org/ontologies/ont.owl#Negative-pressureVentilation">
<rdfs:subClassOf rdf:resource="http://www.co-ode.org/ontologies/ont.owl#MedicalEquipment"/>
</Class>
```

```
<Class rdf:about="http://www.co-ode.org/ontologies/ont.owl#Socket">
<rdfs:subClassOf rdf:resource="http://www.co-
ode.org/ontologies/ont.owl#GeneralFacilities"/>
</Class>
<Class rdf:about="http://www.co-ode.org/ontologies/ont.owl#SpaceArea"/>
<Class rdf:about="http://www.co-ode.org/ontologies/ont.owl#UVGermicidalLamp">
<rdfs:subClassOf rdf:resource="http://www.co-
ode.org/ontologies/ont.owl#MedicalEquipment"/>
</Class>
<Class rdf:about="http://www.co-ode.org/ontologies/ont.owl#Ventilator">
<rdfs:subClassOf rdf:resource="http://www.co-
ode.org/ontologies/ont.owl#MedicalEquipment"/>
</Class>
<NamedIndividual rdf:about="http://example.org/covid19-cabin#Bathroom1">
<rdf:type rdf:resource="http://example.org/covid19-cabin#Bathroom"/>
<cabin:adjacentTo rdf:resource="http://example.org/covid19-cabin#BufferZone1"/>
<cabin:adjacentTo rdf:resource="http://example.org/covid19-cabin#WardA1"/>
<cabin:adjacentTo rdf:resource="http://example.org/covid19-cabin#WardB1"/>
<cabin:contains rdf:resource="http://example.org/covid19-cabin#Shower1"/>
<cabin:contains rdf:resource="http://example.org/covid19-cabin#Shower2"/>
<cabin:contains rdf:resource="http://example.org/covid19-cabin#Sink1"/>
<cabin:contains rdf:resource="http://example.org/covid19-cabin#Sink2"/>
<cabin:contains rdf:resource="http://example.org/covid19-cabin#Toilet1"/>
<cabin:contains rdf:resource="http://example.org/covid19-cabin#Toilet2"/>
<cabin:contains rdf:resource="http://www.co-
ode.org/ontologies/ont.owl#CallingDevice3"/>
<cabin:contains rdf:resource="http://www.co-
ode.org/ontologies/ont.owl#CallingDevice4"/>
<cabin:contains rdf:resource="http://www.co-
ode.org/ontologies/ont.owl#GeneralLighting1"/>
```

<cabin:contains rdf:resource="http://www.co-ode.org/ontologies/ont.owl#GeneralLighting2"/>
<cabin:contains rdf:resource="http://www.co-ode.org/ontologies/ont.owl#WallTypeE1"/>
<cabin:contains rdf:resource="http://www.co-ode.org/ontologies/ont.owl#WallTypeF3"/>
<cabin:contains rdf:resource="http://www.co-ode.org/ontologies/ont.owl#WallTypeF4"/>
<cabin:length rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">3</cabin:length>
<cabin:width rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">3</cabin:width>
<ont:Assemblytime rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">10</ont:Assemblytime>
</NamedIndividual>
<NamedIndividual rdf:about="http://example.org/covid19-cabin#Bed1">
<rdf:type rdf:resource="http://example.org/covid19-cabin#Bed"/>
<cabin:contains rdf:resource="http://example.org/covid19-cabin#Bed1"/>
<cabin:contains rdf:resource="http://example.org/covid19-cabin#Bed2"/>
</NamedIndividual>
<NamedIndividual rdf:about="http://example.org/covid19-cabin#Bed2">
<rdf:type rdf:resource="http://example.org/covid19-cabin#Bed"/>
</NamedIndividual>
<NamedIndividual rdf:about="http://example.org/covid19-cabin#Bed3">
<rdf:type rdf:resource="http://example.org/covid19-cabin#Bed"/>
</NamedIndividual>
<NamedIndividual rdf:about="http://example.org/covid19-cabin#Bed4">
<rdf:type rdf:resource="http://example.org/covid19-cabin#Bed"/>
</NamedIndividual>

```
<NamedIndividual rdf:about="http://example.org/covid19-cabin#BufferZone1">
<rdf:type rdf:resource="http://example.org/covid19-cabin#BufferZone"/>
<cabin:adjacentTo rdf:resource="http://example.org/covid19-cabin#WardA1"/>
<cabin:adjacentTo rdf:resource="http://example.org/covid19-cabin#WardB1"/>
<cabin:contains rdf:resource="http://example.org/covid19-cabin#Sink3"/>
<cabin:contains rdf:resource="http://www.co-
ode.org/ontologies/ont.owl#GeneralLighting5"/>
<cabin:contains rdf:resource="http://www.co-
ode.org/ontologies/ont.owl#MedicalContaminantCollectingBin1"/>
<cabin:contains rdf:resource="http://www.co-ode.org/ontologies/ont.owl#Socket5"/>
<cabin:contains rdf:resource="http://www.co-
ode.org/ontologies/ont.owl#UVGermicidalLamp1"/>
<cabin:contains rdf:resource="http://www.co-
ode.org/ontologies/ont.owl#WallTypeA1"/>
<cabin:length
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">3</cabin:length>
<cabin:width
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">3</cabin:width>
<ont:Assemblytime
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">10</ont:Assemblyti
me>
</NamedIndividual>
<NamedIndividual rdf:about="http://example.org/covid19-cabin#Corridor1">
<rdf:type rdf:resource="http://example.org/covid19-cabin#Corridor"/>
<cabin:adjacentTo rdf:resource="http://example.org/covid19-cabin#BufferZone1"/>
<cabin:adjacentTo rdf:resource="http://example.org/covid19-cabin#WardA1"/>
<cabin:adjacentTo rdf:resource="http://example.org/covid19-cabin#WardB1"/>
<cabin:reachableFrom rdf:resource="http://example.org/covid19-
cabin#BufferZone1"/>
<cabin:length
```

```
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">3</cabin:length>
</NamedIndividual>
<NamedIndividual rdf:about="http://example.org/covid19-cabin#Shower1">
<rdf:type rdf:resource="http://example.org/covid19-cabin#Shower"/>
</NamedIndividual>
<NamedIndividual rdf:about="http://example.org/covid19-cabin#Shower2">
<rdf:type rdf:resource="http://example.org/covid19-cabin#Shower"/>
</NamedIndividual>
<NamedIndividual rdf:about="http://example.org/covid19-cabin#Sink1">
<rdf:type rdf:resource="http://example.org/covid19-cabin#Sink"/>
</NamedIndividual>
<NamedIndividual rdf:about="http://example.org/covid19-cabin#Sink2">
<rdf:type rdf:resource="http://example.org/covid19-cabin#Sink"/>
</NamedIndividual>
<NamedIndividual rdf:about="http://example.org/covid19-cabin#Sink3">
<rdf:type rdf:resource="http://example.org/covid19-cabin#Sink"/>
</NamedIndividual>
<NamedIndividual rdf:about="http://example.org/covid19-cabin#Toilet1">
<rdf:type rdf:resource="http://example.org/covid19-cabin#Toilet"/>
</NamedIndividual>
<NamedIndividual rdf:about="http://example.org/covid19-cabin#Toilet2">
<rdf:type rdf:resource="http://example.org/covid19-cabin#Toilet"/>
</NamedIndividual>
<NamedIndividual rdf:about="http://example.org/covid19-cabin#WardA1">
<rdf:type rdf:resource="http://example.org/covid19-cabin#WardA"/>
<cabin:adjacentTo rdf:resource="http://example.org/covid19-cabin#Bathroom1"/>
<cabin:adjacentTo rdf:resource="http://example.org/covid19-cabin#BufferZone1"/>
<cabin:contains rdf:resource="http://example.org/covid19-cabin#Bed1"/>
<cabin:contains rdf:resource="http://example.org/covid19-cabin#Bed2"/>
<cabin:contains rdf:resource="http://www.co-
```

ode.org/ontologies/ont.owl#CallingDevice1"/>
<cabin:contains rdf:resource="http://www.co-ode.org/ontologies/ont.owl#GeneralLighting3"/>
<cabin:contains rdf:resource="http://www.co-ode.org/ontologies/ont.owl#MedicalMonitor1"/>
<cabin:contains rdf:resource="http://www.co-ode.org/ontologies/ont.owl#MedicalMonitor2"/>
<cabin:contains rdf:resource="http://www.co-ode.org/ontologies/ont.owl#Negative-pressureVentilation1"/>
<cabin:contains rdf:resource="http://www.co-ode.org/ontologies/ont.owl#Socket1"/>
<cabin:contains rdf:resource="http://www.co-ode.org/ontologies/ont.owl#Socket2"/>
<cabin:contains rdf:resource="http://www.co-ode.org/ontologies/ont.owl#WallTypeB1"/>
<cabin:contains rdf:resource="http://www.co-ode.org/ontologies/ont.owl#WallTypeC1"/>
<cabin:contains rdf:resource="http://www.co-ode.org/ontologies/ont.owl#WallTypeD1"/>
<cabin:contains rdf:resource="http://www.co-ode.org/ontologies/ont.owl#WallTypeF1"/>
<cabin:reachableFrom rdf:resource="http://example.org/covid19-cabin#Bathroom1"/>
<cabin:reachableFrom rdf:resource="http://example.org/covid19-cabin#BufferZone1"/>
<cabin:length rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">6</cabin:length>
<cabin:width rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">3</cabin:width>
<ont:Assemblytime rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">15</ont:Assemblytime>
</NamedIndividual>

<NamedIndividual rdf:about="http://example.org/covid19-cabin#WardB1">
<rdf:type rdf:resource="http://example.org/covid19-cabin#WardB"/>
<cabin:adjacentTo rdf:resource="http://example.org/covid19-cabin#WardA1"/>
<cabin:contains rdf:resource="http://example.org/covid19-cabin#Bed3"/>
<cabin:contains rdf:resource="http://example.org/covid19-cabin#Bed4"/>
<cabin:contains rdf:resource="http://www.co-ode.org/ontologies/ont.owl#CallingDevice2"/>
<cabin:contains rdf:resource="http://www.co-ode.org/ontologies/ont.owl#GeneralLighting4"/>
<cabin:contains rdf:resource="http://www.co-ode.org/ontologies/ont.owl#MedicalMonitor3"/>
<cabin:contains rdf:resource="http://www.co-ode.org/ontologies/ont.owl#MedicalMonitor4"/>
<cabin:contains rdf:resource="http://www.co-ode.org/ontologies/ont.owl#Negative-pressureVentilation2"/>
<cabin:contains rdf:resource="http://www.co-ode.org/ontologies/ont.owl#Socket3"/>
<cabin:contains rdf:resource="http://www.co-ode.org/ontologies/ont.owl#Socket4"/>
<cabin:contains rdf:resource="http://www.co-ode.org/ontologies/ont.owl#WallTypeB2"/>
<cabin:contains rdf:resource="http://www.co-ode.org/ontologies/ont.owl#WallTypeC2"/>
<cabin:contains rdf:resource="http://www.co-ode.org/ontologies/ont.owl#WallTypeD2"/>
<cabin:contains rdf:resource="http://www.co-ode.org/ontologies/ont.owl#WallTypeF2"/>
<cabin:reachableFrom rdf:resource="http://example.org/covid19-cabin#Bathroom1"/>
<cabin:reachableFrom rdf:resource="http://example.org/covid19-cabin#BufferZone1"/>
<cabin:length rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">6</cabin:length>

<cabin:width
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">3</cabin:width>
<ont:Assemblytime
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">15</ont:Assemblyti
me>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-
ode.org/ontologies/ont.owl#CallingDevice1">
<rdf:type rdf:resource="http://www.co-ode.org/ontologies/ont.owl#CallingDevice"/>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-
ode.org/ontologies/ont.owl#CallingDevice2">
<rdf:type rdf:resource="http://www.co-ode.org/ontologies/ont.owl#CallingDevice"/>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-
ode.org/ontologies/ont.owl#CallingDevice3">
<rdf:type rdf:resource="http://www.co-ode.org/ontologies/ont.owl#CallingDevice"/>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-
ode.org/ontologies/ont.owl#CallingDevice4">
<rdf:type rdf:resource="http://www.co-ode.org/ontologies/ont.owl#CallingDevice"/>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-ode.org/ontologies/ont.owl#Double-
sidedDoor1">
<rdf:type rdf:resource="http://example.org/covid19-cabin#Door"/>
<ont:attachTo rdf:resource="http://www.co-
ode.org/ontologies/ont.owl#WallTypeA1"/>
<cabin:length
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">1.6</cabin:length>
<ont:hight

rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">2</ont:hight>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-ode.org/ontologies/ont.owl#FixedWindow1">
<rdf:type rdf:resource="http://example.org/covid19-cabin#Window"/>
<ont:attachTo rdf:resource="http://www.co-ode.org/ontologies/ont.owl#WallTypeB1"/>
<cabin:length
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">0.6</cabin:length>
<ont:hasGUID
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">15z0zbrp53VAzjXXBtIrVa</ont:hasGUID>
<ont:hight
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">0.8</ont:hight>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-ode.org/ontologies/ont.owl#FixedWindow2">
<rdf:type rdf:resource="http://example.org/covid19-cabin#Window"/>
<ont:attachTo rdf:resource="http://www.co-ode.org/ontologies/ont.owl#WallTypeB2"/>
<cabin:length
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">0.6</cabin:length>
<ont:hasGUID
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">15z0zbrp53VAzjXXBtIrVd</ont:hasGUID>
<ont:hight
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">0.8</ont:hight>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-ode.org/ontologies/ont.owl#GeneralLighting1">

<rdf:type rdf:resource="http://www.co-ode.org/ontologies/ont.owl#Light"/>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-ode.org/ontologies/ont.owl#GeneralLighting2">
<rdf:type rdf:resource="http://www.co-ode.org/ontologies/ont.owl#Light"/>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-ode.org/ontologies/ont.owl#GeneralLighting3">
<rdf:type rdf:resource="http://www.co-ode.org/ontologies/ont.owl#Light"/>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-ode.org/ontologies/ont.owl#GeneralLighting4">
<rdf:type rdf:resource="http://www.co-ode.org/ontologies/ont.owl#Light"/>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-ode.org/ontologies/ont.owl#GeneralLighting5">
<rdf:type rdf:resource="http://www.co-ode.org/ontologies/ont.owl#Light"/>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-ode.org/ontologies/ont.owl#MedicalContaminantCollectingBin1">
<rdf:type rdf:resource="http://www.co-ode.org/ontologies/ont.owl#MedicalContaminantCollectingBin"/>
<ont:PurchasePrice rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">30</ont:PurchasePrice>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-ode.org/ontologies/ont.owl#MedicalMonitor1">
<rdf:type rdf:resource="http://www.co-ode.org/ontologies/ont.owl#MedicalMonitor"/>

<ont:PurchasePrice
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">1050</ont:Purchase
Price>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-
ode.org/ontologies/ont.owl#MedicalMonitor2">
<rdf:type rdf:resource="http://www.co-
ode.org/ontologies/ont.owl#MedicalMonitor"/>
<ont:PurchasePrice
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">1050</ont:Purchase
Price>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-
ode.org/ontologies/ont.owl#MedicalMonitor3">
<rdf:type rdf:resource="http://www.co-
ode.org/ontologies/ont.owl#MedicalMonitor"/>
<ont:PurchasePrice
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">1050</ont:Purchase
Price>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-
ode.org/ontologies/ont.owl#MedicalMonitor4">
<rdf:type rdf:resource="http://www.co-
ode.org/ontologies/ont.owl#MedicalMonitor"/>
<ont:PurchasePrice
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">1050</ont:Purchase
Price>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-ode.org/ontologies/ont.owl#Negative-
pressureVentilation1">

<rdf:type rdf:resource="http://www.co-ode.org/ontologies/ont.owl#Negative-pressureVentilation"/>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-ode.org/ontologies/ont.owl#Negative-pressureVentilation2">
<rdf:type rdf:resource="http://www.co-ode.org/ontologies/ont.owl#Negative-pressureVentilation"/>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-ode.org/ontologies/ont.owl#SingleLeaf_Door1">
<rdf:type rdf:resource="http://example.org/covid19-cabin#Door"/>
<ont:attachTo rdf:resource="http://www.co-ode.org/ontologies/ont.owl#WallTypeC1"/>
<cabin:length
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">0.8</cabin:length>
<ont:hasGUID
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">15z0zbrp53VAzjXXBtIrTH</ont:hasGUID>
<ont:hight
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">2</ont:hight>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-ode.org/ontologies/ont.owl#SingleLeaf_Door2">
<rdf:type rdf:resource="http://example.org/covid19-cabin#Door"/>
<ont:attachTo rdf:resource="http://www.co-ode.org/ontologies/ont.owl#WallTypeC1"/>
<cabin:length
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">0.8</cabin:length>
<ont:hasGUID
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">15z0zbrp53VAzjXXB

tIrTQ</ont:hasGUID>
<ont:hight
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">2</ont:hight>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-ode.org/ontologies/ont.owl#SingleLeaf_Door3">
<rdf:type rdf:resource="http://example.org/covid19-cabin#Door"/>
<ont:attachTo rdf:resource="http://www.co-ode.org/ontologies/ont.owl#WallTypeC2"/>
<cabin:length
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">0.8</cabin:length>
<ont:hasGUID
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">15z0zbrp53VAzjXXB
tIrTS</ont:hasGUID>
<ont:hight
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">2</ont:hight>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-ode.org/ontologies/ont.owl#SingleLeaf_Door4">
<rdf:type rdf:resource="http://example.org/covid19-cabin#Door"/>
<ont:attachTo rdf:resource="http://www.co-ode.org/ontologies/ont.owl#WallTypeC2"/>
<cabin:length
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">0.8</cabin:length>
<ont:hasGUID
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">15z0zbrp53VAzjXXB
tIrTT</ont:hasGUID>
<ont:hight
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">2</ont:hight>
</NamedIndividual>

<NamedIndividual rdf:about="http://www.co-
ode.org/ontologies/ont.owl#SlidingWindow1">
<rdf:type rdf:resource="http://example.org/covid19-cabin#Window"/>
<ont:attachTo rdf:resource="http://www.co-
ode.org/ontologies/ont.owl#WallTypeB1"/>
<cabin:length
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">0.8</cabin:length>
<ont:hasGUID
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">15z0zbrp53VAzjXXB
tIrTG</ont:hasGUID>
<ont:hight
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">1.1</ont:hight>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-
ode.org/ontologies/ont.owl#SlidingWindow2">
<rdf:type rdf:resource="http://example.org/covid19-cabin#Window"/>
<ont:attachTo rdf:resource="http://www.co-
ode.org/ontologies/ont.owl#WallTypeB2"/>
<cabin:length
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">0.8</cabin:length>
<ont:hasGUID
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">15z0zbrp53VAzjXXB
tIrTG</ont:hasGUID>
<ont:hight
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">1.1</ont:hight>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-
ode.org/ontologies/ont.owl#SlidingWindow3">
<rdf:type rdf:resource="http://example.org/covid19-cabin#Window"/>
<cabin:adjacentTo rdf:resource="http://www.co-

ode.org/ontologies/ont.owl#WallTypeE1"/>
<cabin:length
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">0.8</cabin:length>
<ont:hasGUID
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">15z0zbrp53VAzjXXB
tIrTG</ont:hasGUID>
<ont:hight
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">1.1</ont:hight>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-
ode.org/ontologies/ont.owl#SlidingWindow4">
<rdf:type rdf:resource="http://example.org/covid19-cabin#Window"/>
<ont:attachTo rdf:resource="http://www.co-
ode.org/ontologies/ont.owl#WallTypeE1"/>
<cabin:length
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">0.8</cabin:length>
<ont:hasGUID
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">15z0zbrp53VAzjXXB
tIrTG</ont:hasGUID>
<ont:hight
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">1.1</ont:hight>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-ode.org/ontologies/ont.owl#Socket1">
<rdf:type rdf:resource="http://www.co-ode.org/ontologies/ont.owl#Socket"/>
<rdfs:comment
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">220V/10A 6-hole
socket</rdfs:comment>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-ode.org/ontologies/ont.owl#Socket2">
<rdf:type rdf:resource="http://www.co-ode.org/ontologies/ont.owl#Socket"/>

<rdfs:comment
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">220V/10A 6-hole
socket</rdfs:comment>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-ode.org/ontologies/ont.owl#Socket3">
<rdf:type rdf:resource="http://www.co-ode.org/ontologies/ont.owl#Socket"/>
<rdfs:comment
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">220V/10A 6-hole
socket</rdfs:comment>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-ode.org/ontologies/ont.owl#Socket4">
<rdf:type rdf:resource="http://www.co-ode.org/ontologies/ont.owl#Socket"/>
<rdfs:comment
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">220V/10A 6-hole
socket</rdfs:comment>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-ode.org/ontologies/ont.owl#Socket5">
<rdf:type rdf:resource="http://www.co-ode.org/ontologies/ont.owl#Socket"/>
<rdfs:comment
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">220V/10A 2-hole
socket</rdfs:comment>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-
ode.org/ontologies/ont.owl#UVGermicidalLamp1">
<rdf:type rdf:resource="http://www.co-
ode.org/ontologies/ont.owl#UVGermicidalLamp"/>
<ont:PurchasePrice
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">277</ont:PurchaseP
rice>
</NamedIndividual>

<NamedIndividual rdf:about="http://www.co-ode.org/ontologies/ont.owl#Ventilator1">
<rdf:type rdf:resource="http://www.co-ode.org/ontologies/ont.owl#Ventilator"/>
<ont:PurchasePrice
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">3500</ont:PurchasePrice>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-ode.org/ontologies/ont.owl#Ventilator2">
<rdf:type rdf:resource="http://www.co-ode.org/ontologies/ont.owl#Ventilator"/>
<ont:PurchasePrice
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">3500</ont:PurchasePrice>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-ode.org/ontologies/ont.owl#WallTypeA1">
<rdf:type rdf:resource="http://example.org/covid19-cabin#Wall"/>
<cabin:length
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">3</cabin:length>
<ont:hasGUID
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">15z0zbrp53VAzjXXBtIq91</ont:hasGUID>
<ont:hight
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">3</ont:hight>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-ode.org/ontologies/ont.owl#WallTypeB1">
<rdf:type rdf:resource="http://example.org/covid19-cabin#Wall"/>
<cabin:length
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">3</cabin:length>

<ont:hasGUID
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">15z0zbrp53VAzjXXB
tIqEx</ont:hasGUID>
<ont:hight
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">3</ont:hight>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-
ode.org/ontologies/ont.owl#WallTypeB2">
<rdf:type rdf:resource="http://example.org/covid19-cabin#Wall"/>
<cabin:length
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">3</cabin:length>
<ont:hasGUID
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">15z0zbrp53VAzjXXB
tIqEx</ont:hasGUID>
<ont:hight
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">3</ont:hight>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-
ode.org/ontologies/ont.owl#WallTypeC1">
<rdf:type rdf:resource="http://example.org/covid19-cabin#Wall"/>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-
ode.org/ontologies/ont.owl#WallTypeC2">
<rdf:type rdf:resource="http://example.org/covid19-cabin#Wall"/>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-
ode.org/ontologies/ont.owl#WallTypeD1">
<cabin:length
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">6</cabin:length>
<ont:hasGUID

```
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">15z0zbrp53VAzjXXB
tIqEl</ont:hasGUID>
<ont:hight
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">3</ont:hight>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-
ode.org/ontologies/ont.owl#WallTypeD2">
<cabin:length
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">6</cabin:length>
<ont:hasGUID
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">15z0zbrp53VAzjXXB
tIqEl</ont:hasGUID>
<ont:hight
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">3</ont:hight>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-
ode.org/ontologies/ont.owl#WallTypeE1">
<rdf:type rdf:resource="http://example.org/covid19-cabin#Wall"/>
<cabin:length
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">3</cabin:length>
<ont:hasGUID
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">15z0zbrp53VAzjXXB
tIqEn</ont:hasGUID>
<ont:hight
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">3</ont:hight>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-
ode.org/ontologies/ont.owl#WallTypeF1">
<cabin:length
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">3</cabin:length>
```

<ont:hasGUID
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">15z0zbrp53VAzjXXB
tIq9S</ont:hasGUID>
<ont:hight
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">3</ont:hight>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-
ode.org/ontologies/ont.owl#WallTypeF2">
<cabin:length
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">3</cabin:length>
<ont:hasGUID
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">15z0zbrp53VAzjXXB
tIq9S</ont:hasGUID>
<ont:hight
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">3</ont:hight>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-
ode.org/ontologies/ont.owl#WallTypeF3">
<cabin:length
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">3</cabin:length>
<ont:hasGUID
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">15z0zbrp53VAzjXXB
tIq9S</ont:hasGUID>
<ont:hight
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">3</ont:hight>
</NamedIndividual>
<NamedIndividual rdf:about="http://www.co-
ode.org/ontologies/ont.owl#WallTypeF4">
<cabin:length
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">3</cabin:length>

```
<ont:hasGUID
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">15z0zbrp53VAzjXXB
tIq9S</ont:hasGUID>
<ont:hight
rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">3</ont:hight>
</NamedIndividual>
<rdf:Description rdf:about="http://www.w3.org/2001/XMLSchema#unsignedShort">
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">VL-
115.G,1×15 W,254 nm,VWR Catalog Number:VILB312111511_P</rdfs:comment>
</rdf:Description>
</rdf:RDF>
```

## Coding for Data conversion

```
import ifcopenshell
from rdflib import Graph, Namespace, RDF, Literal

# Load the IFC file
ifc_file = ifcopenshell.open('D12137.ifc')

# Define namespaces
IFC = Namespace("http://ifcowl.openbimstandards.org/IFC2X3_TC1#")
EX = Namespace("http://example.com/ifc2owl#")

# Create an RDF graph
g = Graph()

# Bind namespaces
g.bind("ifc", IFC)
g.bind("ex", EX)
```

```python
# Function to map IFC entities to OWL
def map_ifc_entity_to_owl(entity, ifc_type):
    entity_id = EX[entity.GlobalId]
    g.add((entity_id, RDF.type, IFC[ifc_type]))
    if entity.Name:
        g.add((entity_id, EX.hasName, Literal(entity.Name)))
    if entity.Description:
        g.add((entity_id, EX.hasDescription, Literal(entity.Description)))
    # Add more mappings as needed

# Extract and map IfcWall entities
walls = ifc_file.by_type('IfcWall')
for wall in walls:
    map_ifc_entity_to_owl(wall, 'IfcWall')

# Extract and map IfcDoor entities
doors = ifc_file.by_type('IfcDoor')
for door in doors:
    map_ifc_entity_to_owl(door, 'IfcDoor')

# Extract and map IfcWindow entities
windows = ifc_file.by_type('IfcWindow')
for window in windows:
    map_ifc_entity_to_owl(window, 'IfcWindow')

# Extract and map IfcBuilding entities
buildings = ifc_file.by_type('IfcBuilding')
for building in buildings:
    map_ifc_entity_to_owl(building, 'IfcBuilding')
```

```
# Extract and map IfcSite entities
sites = ifc_file.by_type('IfcSite')
for site in sites:
    map_ifc_entity_to_owl(site, 'IfcSite')

# Extract and map other IFC entities similarly...
# Add more entity types as needed

# Serialize the graph to an OWL file
g.serialize("ward.owl", format="turtle")

print("Conversion complete. OWL file created: output.owl")
```

## Coding for Convex site laytou

```
import matplotlib.pyplot as plt
import math
from shapely.geometry import Polygon, Point

# Define the irregular polygon coordinates
polygon_coords = [(30,0), (60,0), (60,20), (55,20), (55,40), (40,40), (40,20), (30,20),
(30,0)]

# Create the polygon
polygon = Polygon(polygon_coords)

# Define the dimensions and space parameters
l = 3
```

```
w = 6
h_ij = 0
v_jk = 3

# Calculate the bounding box of the polygon
min_x, min_y, max_x, max_y = polygon.bounds

# Function to check if a room fits within the polygon
def room_fits(x, y, l, w, polygon):
    room = Polygon([(x - l / 2, y - w / 2), (x + l / 2, y - w / 2), (x + l / 2, y + w / 2), (x -
l / 2, y + w / 2)])
    return polygon.contains(room)

# Function to draw the layout within the polygon
def draw_max_layout_irregular(polygon, l, w, h_ij, v_jk):
    fig, ax = plt.subplots()
    x, y = polygon.exterior.xy
    ax.plot(x, y, color='black')

    ax.set_xlim(min_x, max_x)
    ax.set_ylim(min_y, max_y)
    ax.set_aspect('equal')
    plt.gca().invert_yaxis()

    room_index = 1  # Start numbering from 1
    y = min_y + w / 2

    while y + w / 2 <= max_y:
        x = min_x + l / 2
        row_rooms = 0
```

```
        row_start_x = x
        # Calculate how many rooms fit in this row
        while x + l / 2 <= max_x:
            if room_fits(x, y, l, w, polygon):
                row_rooms += 1
            x += l + h_ij
        # Adjust the number of rooms to be a multiple of 3 for clusters
        row_rooms = (row_rooms // 3) * 3
        # Place the rooms if the row can fit a multiple of 3
        if row_rooms > 0:
            x = row_start_x
            for i in range(row_rooms):
                if room_fits(x, y, l, w, polygon):
                    # Determine color based on cluster logic
                    if i % 3 == 0 or i % 3 == 2:  # Start and end of a cluster
                        color = 'red'
                    else:  # Middle of a cluster
                        color = 'blue'
                    rect = plt.Rectangle((x - l / 2, y - w / 2), l, w, fill=True, edgecolor='black',
facecolor=color)
                    ax.add_patch(rect)
                    plt.text(x, y, f"m{room_index}", ha='center', va='center', color='white')
                    room_index += 1
                x += l + h_ij
        y += w + v_jk

    plt.xlabel("X")
    plt.ylabel("Y")
    plt.title("Maximized Room Layout in Irregular Space")
    plt.grid(True)
```

```
    plt.show()

# Draw the layout
draw_max_layout_irregular(polygon, l, w, h_ij, v_jk)
```

## Coding for Concave site layout

```
import matplotlib.pyplot as plt
import math
from shapely.geometry import Polygon, Point

# Define the irregular polygon coordinates
polygon_coords = [(0, 0), (30, 0), (30, 40), (20, 40), (20, 35), (5, 35), (5, 40), (0, 40),
(0, 0)]

# Create the polygon
polygon = Polygon(polygon_coords)

# Define the dimensions and space parameters
l = 3
w = 6
h_ij = 0
v_jk = 3

# Calculate the bounding box of the polygon
min_x, min_y, max_x, max_y = polygon.bounds

# Function to check if a room fits within the polygon
def room_fits(x, y, l, w, polygon):
```

```
    room = Polygon([(x - l / 2, y - w / 2), (x + l / 2, y - w / 2), (x + l / 2, y + w / 2), (x - l / 2, y + w / 2)])
    return polygon.contains(room)

# Function to draw the layout within the polygon
def draw_max_layout_irregular(polygon, l, w, h_ij, v_jk):
    fig, ax = plt.subplots()
    x, y = polygon.exterior.xy
    ax.plot(x, y, color='black')

    ax.set_xlim(min_x, max_x)
    ax.set_ylim(min_y, max_y)
    ax.set_aspect('equal')
    plt.gca().invert_yaxis()

    room_index = 1
    y = min_y + w / 2

    while y + w / 2 <= max_y:
        x = min_x + l / 2
        row_rooms = 1
        row_start_x = x
        # Calculate how many rooms fit in this row
        while x + l / 2 <= max_x:
            if room_fits(x, y, l, w, polygon):
                row_rooms += 1
            x += l + h_ij
        # Adjust the number of rooms to be a multiple of 3
        row_rooms = (row_rooms // 3) * 3
        # Place the rooms if the row can fit a multiple of 3
```

```
        if row_rooms > 0:
            x = row_start_x
            for i in range(row_rooms):
                if room_fits(x, y, l, w, polygon):
                    if i % 3 == 0 or i % 3 == 2:
                        color = 'red'
                    else:
                        color = 'blue'
                    rect = plt.Rectangle((x - l / 2, y - w / 2), l, w, fill=True, edgecolor='black',
facecolor=color)
                    ax.add_patch(rect)
                    plt.text(x, y, f"m{room_index}", ha='center', va='center', color='white')
                    room_index += 1
                x += l + h_ij
        y += w + v_jk

    plt.xlabel("X")
    plt.ylabel("Y")
    plt.title("Maximized Room Layout in Irregular Space")
    plt.grid(True)
    plt.show()

# Draw the layout
draw_max_layout_irregular(polygon, l, w, h_ij, v_jk)
```

## Coding for sport venue experimental layout

```
import matplotlib.pyplot as plt
import matplotlib.patches as patches
```

```
from matplotlib.path import Path
import numpy as np

# Define the problem parameters
l = 3   # Length of each space
w = 6   # Width of each space

hij = 0  # Minimum horizontal distance between spaces
vjk = 3  # Minimum vertical distance between spaces

hi11 = 9  # Minimum horizontal distance between space and boundary
vi11 = 6  # Minimum vertical distance between space and boundary

# Define the irregular boundary coordinates
boundary_coords = [
    (40.3, -44.3),
    (28.7, -55.9),
    (28.7, -79.8),
    (40.3, -91.5),
    (97.3, -91.5),
    (108.9, -79.8),
    (108.9, -55.9),
    (97.3, -44.3),
    (40.3, -44.3)  # Closing the loop
]

# Convert boundary coordinates to numpy arrays for easier manipulation
boundary_coords = np.array(boundary_coords)

# Create the path for the irregular boundary
```

```
boundary_path = Path(boundary_coords)

# Function to check if a point is within the boundary
def is_within_boundary(x, y, path):
    return path.contains_point((x, y))

# Find the bounding box of the irregular space
min_x, min_y = np.min(boundary_coords, axis=0)
max_x, max_y = np.max(boundary_coords, axis=0)

# Generate coordinates and labels for each space
spaces = []
label_index = 1

x_start = min_x + hi11 + l / 2
y_start = max_y - vi11 - w / 2

x = x_start
while x + l/2 <= max_x - hi11:
    y = y_start
    while y - w/2 >= min_y + vi11:
        cluster = []
        for cluster_index in range(3):
            x_cluster = x + cluster_index * (l + hij)
                if x_cluster + l/2 <= max_x - hi11 and is_within_boundary(x_cluster, y,
boundary_path):
                cluster.append((x_cluster, y, f"m{label_index}"))
                label_index += 1
        if len(cluster) == 3:
            spaces.extend(cluster)
```

```
        y -= (w + vjk)
    x += 3 * (l + hij)  # Move to the next cluster position

# Print the number of rooms
num_rooms = len(spaces)
print(f"Number of rooms: {num_rooms}")

# Draw the layout
fig, ax = plt.subplots()
plt.xlim(min_x - 10, max_x + 10)
plt.ylim(min_y - 10, max_y + 10)

# Draw the boundary
patch = patches.PathPatch(boundary_path, edgecolor='black', facecolor='none', lw=2)
ax.add_patch(patch)

# Draw each space
for i, (x, y, label) in enumerate(spaces):
    # Determine the color of the space
    if i % 3 == 0 or i % 3 == 2:  # First and last in the cluster
        color = 'red'
    else:
        color = 'lightblue'
    rect = patches.Rectangle((x - l/2, y - w/2), l, w, linewidth=1, edgecolor='blue',
facecolor=color)
    ax.add_patch(rect)
    plt.text(x, y, label, ha='center', va='center', fontsize=8)

plt.gca().set_aspect('equal', adjustable='box')
plt.grid(True)
```

```
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Ward Layout in Sport Venue')
plt.show()

# Print the number of rooms
num_rooms = len(spaces)
num_rooms
```