AN INFORMATION EXCHANGE FRAMEWORK FOR BIM, BAS, AND IOT DATA USING SEMANTIC WEB TECHNOLOGIES

A Dissertation
Presented to
The Academic Faculty

by

Shu Tang

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Architecture
College of Design

Georgia Institute of Technology
May 2020

AN INFORMATION EXCHANGE FRAMEWORK FOR BIM, BAS, AND IOT DATA USING SEMANTIC WEB TECHNOLOGIES

Approved by:

Dr. Dennis R. Shelden, Advisor
School of Architecture
Georgia Institute of Technology

Prof. Charles M. Eastman
School of Architecture
Georgia Institute of Technology

Prof. John E. Taylor
School of Civil and Environmental Engineering
Georgia Institute of Technology

Dr. Pardis Pishdad-Bozorgi
School of Building Construction
Georgia Institute of Technology

Dr. Pieter Pauwels
Department of the Built Environment
Eindhoven University of Technology

Date Approved: April 06, 2020

To my loving parents and beloved husband

ACKNOWLEDGEMENTS

First and foremost, I want to thank my advisor, Dr. Dennis R. Shelden, for his guidance and support over the last three years. He is an impeccable mentor and has been a great inspiration to me. It is my honor to be in his team. I would like to extend my sincere gratitude to my committee members Professor Charles M. Eastman, Professor John E. Taylor, Dr. Pardis Pishdad-bozorgi, Dr. Pieter Pauwels for their advice, time, comments, and help.

I always feel lucky to be part of the Digital Building Lab (DBL) research group which has been a source of friendships, advice, and collaboration. I would like to thank all the fellow student colleagues from DBL. It is also a wonderful journey to work with research collaborators from the School of Building Construction. Special thanks to Dr. Xinghua Gao for his patient and help when I continuously raising questions through this Ph.D. process.

Also, I would like to thank my friends for their company and support, especially Chen Wu, Yuqing Hu, Rui Chen, Chen Jiang, Xiao Liu, Qi Chen, Yue Xu, and Lianzi Heyue. Thank you all for your encouragement and enjoyable time for these years.

Lastly, I would like to thank my families for all their love, support, encouragement, and trust. Many warm thanks to my four parents taking care of my little girl. And most of all, I feel extremely lucky to have my supportive, patient, and loving husband Xiang Luo. My heartfelt thanks and appreciation to all my families, I could not have finished this journey without them.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

API	Application Program Interface
AEC	Architecture Engineering and Construction
BACnet	Building Automation and Control Networks
BACS	Building Automation and Control System
BAS	Building Automation System
BEMS	Building Energy Management Systems
BIM	Building Information Modeling
BMS	Building Management Systems
BPMN	Business Process Modelling Notation
CLI	Command Line Interface
COBie	Construction Operation Building Information Exchange
CSV	Comma-Separated Value
DW	Data Warehouse
DL	Description Logic
DCMI	Dublin Core Metadata Initiative
EMCS	Energy Management Control Systems
ER	Exchange Requirements
FCU	Fan Coil Unit
FMS	Facility Management Systems
FOAF	Friend Of A Friend
FP	Functional Parts
GUID	Global Unique Identifier
GUI	Graphic User Interface
gbXML	Green Building XML
HTO	Haystack Tagging Ontology

IDM	Information Delivery Manual
IFC	Industry Foundation Class
IoT	Internet of Things
JSON	JavaScript Object Notation
LBD	Linked Building Data
MVD	Model View Definition
ODBC	Open Database Connectivity
O&M	Operation and Maintenance
RDFS	RDF Schema
RDF	Resource Description Framework
REST	Representational State Transfer
RIF	Rule Interchange Format
SSN	Semantic Sensor Network
SWRL	Semantic Web Rule Language
SKOS	Simple Knowledge Organization System
SPARQL	Simple Protocol and RDF Query Language
SAREF	Smart Appliances Reference ontology
SOA	Service-Oriented Architecture
SQL	Structured Query Language
URIs	Universal Resource Identifiers
UUID	Universal Unique Identifiers
OWL	Web Ontology Language

SUMMARY

Building Information Modeling (BIM) offers high fidelity representations of the building contextual data. By incorporating geometry, spatial location, and a scalable set of metadata properties, BIM models provide operable datasets capturing the as-designed building objects, properties, and spatial organization as a set of virtual assets. Internet of Things (IoT) data enhances this information set by providing real-time and recordable status from the actual operations in construction and operations. This data is generally organized as time-series data streams of individual sensor data point samples over time, frequently with some higher-level organization into equipment, etc. Building Automation System (BAS) groups IoT devices with logical control and management. Apart from the time-serious data of IoT devices, the metadata of sensors, actuators, equipment, and control sequences is stored in BAS.

With digital technologies like BIM, IoT, and BAS, an increasing amount of data is being created. Data silos in Architecture Engineering and Construction (AEC) industry emerged. These data silos namely building contextual data in BIM, time-series data from IoT devices, and BAS metadata lack the common data representations, thus prevent information exchange between buildings and innovative applications. This problem has been noticed for a while. Efforts to address the interoperability issue have limitations. The isolation between BIM-based building contextual information, IoT devices' time-series data, and BAS metadata still exist.

This research aims to develop a framework to facilitate information exchange between BIM-based building contextual data, IoT devices' time-series data, and BAS

metadata using Semantic Web technology. The research objectives are: **i)** conduct a comprehensive literature review on BIM and IoT integration based on domain of application and integration methods to summarized an optimal current approach; **ii)** propose a framework which enables information exchange among semantically described building contextual data, BAS metadata, and time-series data; **iii)** the proposed framework uses BOT and BRICK schema to describe building contextual data and BAS metadata; **iv)** create an MVD for BIM assisted BAS design and information exchange using BACnet and IFC use case; **v)** validate the framework with the use case and data from Georgia Tech campus building data.

Emerging technologies provide more advanced methods for data silos integration, thus facilitating and laying foundations for potential applications. The major contributions and impacts of this research include: **i)** the literature review in this research contributes to the body of knowledge by presenting an in-depth review of BIM and IoT devices integration in the AEC industry from domain application perspective and integration methodologies. Apart from the summarized application domains, the author summarized five integration methods with description, examples, discussion and suggested the current optimal approach; **ii)** the proposed framework facilitates information exchange between building contextual information in BIM, BAS metadata, and time-series data generated from IoT devices. This framework lays a foundation for future studies utilizing BIM, BAS, and IoT data such as facility management, energy bench marking, and fault detection. The overall methodology can also inspire other data silos such as construction management data, logistic data, climate data, simulation data, etc integration, thus stimulate new applications. The framework brings the higher-level connectivity between the AEC industry and other domains. It sets a foundation for data exchange between isolated information islands in the AEC industry, and brings insights for cross-domain data interlinking; **iii)** demonstrates how to reuse existing ontologies for new applications. The data serialization process can be a guideline for future researchers to implement semantic web technologies for various purposes; **iv)** the developed BACnet MVD enables information exchange for BIM assisted BAS design and operation using BACnet and IFC. This MVD lays a solid foundation for exchanging BAS information conforming to the BACnet protocol with the IFC data model for BAS design and operation; **v)** the framework validation process demonstrates the data silos interlinking for BIM assisted BAS design and operation. In this way, BAS information represented in the open BIM standard or linked data formats can unlock the potential of future smart building information exchange between various tools throughout multi-project stages and information integration with other domains.

CHAPTER 1 INTRODUCTION

This chapter introduces an overview of the thesis background and challenges of data silos isolation in the AEC industry. The motivation and brief problem statement are explained.

1.1 Overview

Building Information Modeling (BIM) has become an established paradigm for the development of enhanced project delivery practices. Properly developed and managed, BIM-centric project delivery makes available high fidelity, geometrically, and positionally accurate, uniquely identifiable building contextual data together with a wealth of descriptive and operable metadata.

According to [1], the Internet of Things (IoT) can be defined as the " interconnection of sensing and actuating devices providing the ability to share information across platforms through a unified framework, developing a common operating picture for enabling innovative applications". Driven by IoT devices, technology innovations have led to connected building systems like lighting, power, safety, occupancy, etc.

The increasing notice of IoT has also increased awareness of Building Automation System (BAS) connectivity opportunities. It is reported that 14% of the U.S. commercial buildings had adopted BAS for data collection and automatic control of building systems as of 2012 [2]. BAS is a data acquisition and control system that incorporates various functionalities provided by the control system of a building. BAS is also known as Energy Management Control Systems (EMCS), Building Management Systems (BMS), Building

Energy Management Systems (BEMS), Facility Management Systems (FMS), and Building Automation and Control System (BACS) [3]. Smart buildings empowered by smart BAS allow data collection, analysis, and control to assist with facility functions and services [4].

BIM models offer high fidelity representations of the building contextual data. By incorporating geometry, spatial location, and a scalable set of metadata properties, BIM models provide operable datasets capturing the as-designed building objects, properties, and spatial organization as a set of virtual assets. IoT data enhances this information set by providing real-time and recordable status from the actual operations in construction and operations. This data is generally organized as time-series data streams of individual sensor point samples over time, frequently with some higher-level organization into equipment, etc. BAS groups IoT devices with logical control and management. Apart from the time-serious data of IoT devices, the metadata of sensors, actuators, equipment, and control sequences is stored in BAS.

With digital technologies like BIM, IoT, and BAS, an increasing amount of data is being created. Data silos in the AEC industry emerged. These data silos namely building contextual data in BIM, time-series data from IoT devices, and BAS metadata lack the common data representations, thus prevent information exchange between buildings and innovative applications. This problem has been noticed for a while. Efforts to address the interoperability issue have limitations. The isolation between BIM-based building contextual information, IoT devices' time-series data, and BAS metadata still exist.

1.2 Motivation and Problem Statement

The AEC industry cannot escape from the pervasive digital revolution. Smart buildings are the trend of the next generation's commercial buildings, which link different building systems together with BAS. Smart buildings empowered by smart BAS allow data collection, analysis, and control to assist with facility functions and services through various IoT devices [4].

1.2.1 Motivation

The digital form of buildings, data generated from IoT devices, and data resides in BAS are valuable. BIM models offer high fidelity representations of the building contextual data. By incorporating geometry, spatial location, and a scalable set of metadata properties, BIM models provide operable datasets capturing the as-designed building objects, properties, and spatial organization as a set of virtual assets. IoT data enhances this information set by providing real-time and recordable status from the actual operations in construction and operations. This data is generally organized as time-series data streams of individual sensor point samples over time, frequently with some higher-level organization into equipment, etc. BAS groups IoT devices with logical control and management. Apart from the time-serious data of IoT devices, the metadata of sensors, actuators, equipment, and control sequences is stored in BAS. There are other AEC related data such as climate data for simulation, logistic data for construction management, and cost data for estimation are defined in various data formats. Emerging technologies provide more advanced tools for data silos integration, thus facilitating potential applications. Integrated data silos, which are serialized in both machine-readable and human-readable formats, can be linked to the cloud and stimulate cross-domain innovations.

1.2.2 Problem Statement and Current Research Gap

With digital technologies like BIM, IoT, and BAS, an increasing amount of data is being created. Data silos in the AEC industry emerged. These data silos namely building contextual data in BIM, time-series data from IoT devices, and BAS metadata lack the common data representations, thus prevent information exchange between buildings and innovative applications. This problem has been noticed for a while. Efforts to address the interoperability issue are: i) creating a common data model like Industry Foundation Class (IFC), Green Building XML (gbXML), Project Haystack, etc; ii) adapter approach enabling data transfer between various native data formats; iii) semantic web approach uses ontologies to semantically describe data in Resource Description Framework (RDF) [5].

Although previous studies have explored the above-mentioned information exchange issue in the AEC industry, there are some limitations. The current research studies are either focusing on particular applications like energy performance assessment [5][6], facility management [7], or creating new common data models for a single domain [8][9]. As technologies evolving, there are better solutions to integrate data silos using more advanced tools. Current studies that utilized out of date technologies may not be practical and inhibit implementation. Since there are various approaches to integration, the most appropriate approach should be analyzed and improved. Only a few studies that focused on the integration of BIM contextual information and time-series data are discovered, these studies are neither focusing on BAS metadata nor using most suitable standardized data models for integration. The studies that focus on information exchange between BIM, IoT data, BAS metadata are rare. The data isolation between BIM-based building contextual information, IoT devices' time-series data, and BAS metadata still exists.

CHAPTER 2 OBJECTIVE AND RESEARCH SCOPE

2.1 Research Objective, Hypothesis, and Research Questions

2.1.1 Research Objective

This research aims to develop a framework to facilitate information exchange between BIM-based building contextual data, IoT devices' time-series data, and BAS metadata using Semantic Web technology. This framework integrates semantically described building contextual data and BAS metadata together with time-series data. This framework uses BOT and BRICK schema ontologies to describe building contextual data and BAS metadata. This thesis creates an MVD for BIM assisted BAS design and information exchange using BACnet and IFC use case. A validation process of the developed framework is based on a proof-of-concept use case namely BIM assisted BAS design and operation information exchange. This use case is validated with data in the Georgia Institute of Technology campus buildings and Johnson Control BAS software. The framework also enables access and various queries over integrated data silos stored in graph database and relational database.

The research objectives are:

1. Conduct a comprehensive literature review on BIM and IoT integration;
2. Propose a framework which integrates semantically described building contextual data and BAS metadata together with time-series data;
3. Use existing ontologies BOT and BRICK schema to describe building contextual

data and BAS metadata and serialize collected data according to the selected ontologies into both human-readable and machine-readable serialization.

4. Create a BACnet MVD for BIM assisted BAS design and operation information exchange using BACnet and IFC use case

5. Validate the proposed framework with a proof-of-concept 'BIM assisted BAS design and operation information exchange' use case and data from Georgia Tech campus building

2.1.2 Research Hypothesis

The author made several **hypotheses** for the proposed study:

1. The proposed framework adopts the pros of the current integration methodology and assists information exchange between AEC data silos.
2. The proposed framework can be validated through real use case data and evaluated by the use case queries.

2.1.3 Research Questions

The objective of this research is to develop a framework to facilitate information exchange between BIM-based building contextual data, IoT devices' time-series data, and BAS metadata using Semantic Web technology. Five **research questions** are listed below to fulfill the research objective:

1. What is the most appropriate integration method for building contextual information, IoT time-series data, and BAS metadata? What is the current

research gap? What are the semantic web technologies that can be used for integration?

2. How to integrate BIM contextual data, BAS metadata, and IoT time-series data? What is the improvement from the existing approach to the proposed framework?
3. How to effectively collect use case data? what are the characteristics and schema of the collected data? Are there any reusable ontologies to semantically describe data silos? How to serialize collected data into target formats based on selected ontologies
4. What is the methodology to create an MVD for BIM assisted BAS design and operation information exchange? How to implement the MVD?
5. How to validate and evaluate the proposed framework for the BIM assisted BAS design and operation information exchange use case? What are the use case data and how to implement the validation processes?

2.2 Research Scope and Contribution

This research develops a framework to facilitate information exchange between BIM-based building contextual data, IoT devices' time-series data, and BAS metadata using Semantic Web technology. This framework integrates semantically described building contextual data and BAS metadata together with time-series data.

2.2.1 Research Scope

There are other AEC data in various formats, which can also be integrated using semantic web technology, this study only focuses on integration between BIM-based building contextual data, BAS metadata, and time-series data generated by IoT devices.

Building contextual data resides in BIM models is informatics, the complete set of building contextual data is unnecessary in the proposed framework. To achieve the information exchange goal between BIM, IoT, and BAS, building contextual data related to site, building, storey, space, part of elements and spatial relationships is within the framework's scope.

BAS contains a series of subsystems with a great number of sensors and actuators. The framework focus on data points from Georgia Tech facility management systems including Johnson Control BAS software and GT Facility ION sever. Only electricity meters and data points recorded in BAS (MetaSys) are within the scope. Those devices whose data points are not available in BAS are not within the scope. The instantiate process of the BAS metadata model is semi-automatic, strictly automatic parsing process for BAS data points is not within the scope.

2.2.2 Research Contributions

The major contributions and impacts of this research include:

- The literature review in this research contributes to the body of knowledge by presenting an in-depth review of BIM and IoT devices integration in the AEC industry from domain application perspective and integration methodologies. Apart from the summarized application domains, the authors summarized five integration methods with description, examples, discussion, and suggested the current optimal approach.
- The proposed framework facilitates information exchange between building contextual information in BIM, BAS metadata, and time-series data generated from

IoT devices. This framework lays a solid foundation for future studies utilizing BIM, BAS, and IoT data such as facility management, energy bench marking, and fault detection. The overall methodology can also inspire other data silos such as construction management data, logistic data, climate data, simulation data, etc integration, thus stimulate new applications. It sets a foundation for data exchange between isolated information islands in the AEC industry, and brings insights for cross-domain data interlinking.

- The proposed framework demonstrates how to reuse existing ontologies for new applications. The data serialization process can be a guideline for future researchers to implement semantic web technologies for various purposes.
- The developed BACnet MVD enables information exchange for BIM assisted BAS design and operation using BACnet and IFC. This MVD lays a solid foundation for exchanging BAS information conforming to the BACnet protocol with the IFC data model for BAS design and operation.
- The framework validation process demonstrates the data silos interlinking for BIM assisted BAS design and operation. In this way, BAS information represented in the open BIM standard or linked data formats can unlock the potential of future smart building information exchange between various tools throughout multi-project stages and information integration with other domains.

2.3 Overall Research Procedures

This research aims to develop a framework to facilitate information exchange between BIM-based building contextual data, IoT devices' time-series data, and BAS metadata using Semantic Web technology. The overall procedures of this research are illustrated in **Figure 2.1**:

1. The research starts with an in-depth literature review on BIM and IoT device integration by analyzing the application domains and integration methods. Instead of focusing on BAS, the author explores BIM and IoT integration in a broad perspective to profoundly discover the integration methods. The pros and cons of current integration methods are analyzed to suggest the current optimal integration approach. Based on this approach, the author evaluates the limitation and where the improvement can be made to propose the framework in this study. Besides, the literature review chapter investigates semantic web technologies related to BAS, BIM, and IoT data.
2. Based on the literature review, the second step aims to develop a framework to: i) integrate building contextual data, BAS metadata, and time-series data; ii) semantically describe building contextual data and BAS metadata based on BOT and BRICK schema ontologies; iii) validate serialized data with RDF validator and BRICK TTL viewer; iv) create links between data silos using databases.
3. The third step focuses on the methodology of data acquisition, serialization, and validation. This step focuses on effectively collect use case data including building contextual data, BAS metadata, and IoT time-series data. The data characteristics and data schemas are analyzed for data serialization based on selected ontologies. In this step, unstructured raw data is serialized into RDF serialization based on the BOT and BRICK. RDF syntax, tools, methods are explored for data source transformation. Also, serialized data are validated with several RDF validators.
4. The fourth step is to create a BACnet MVD for 'BIM assisted BAS design and

operation information exchange using BACnet and IFC' use case.

5. This step focuses on the validation of the proposed framework based on the use case. This step implements data collected from the Georgia Tech facility management system.
6. Finally, the discussion and limitations of the proposed framework and validation process are concluded.

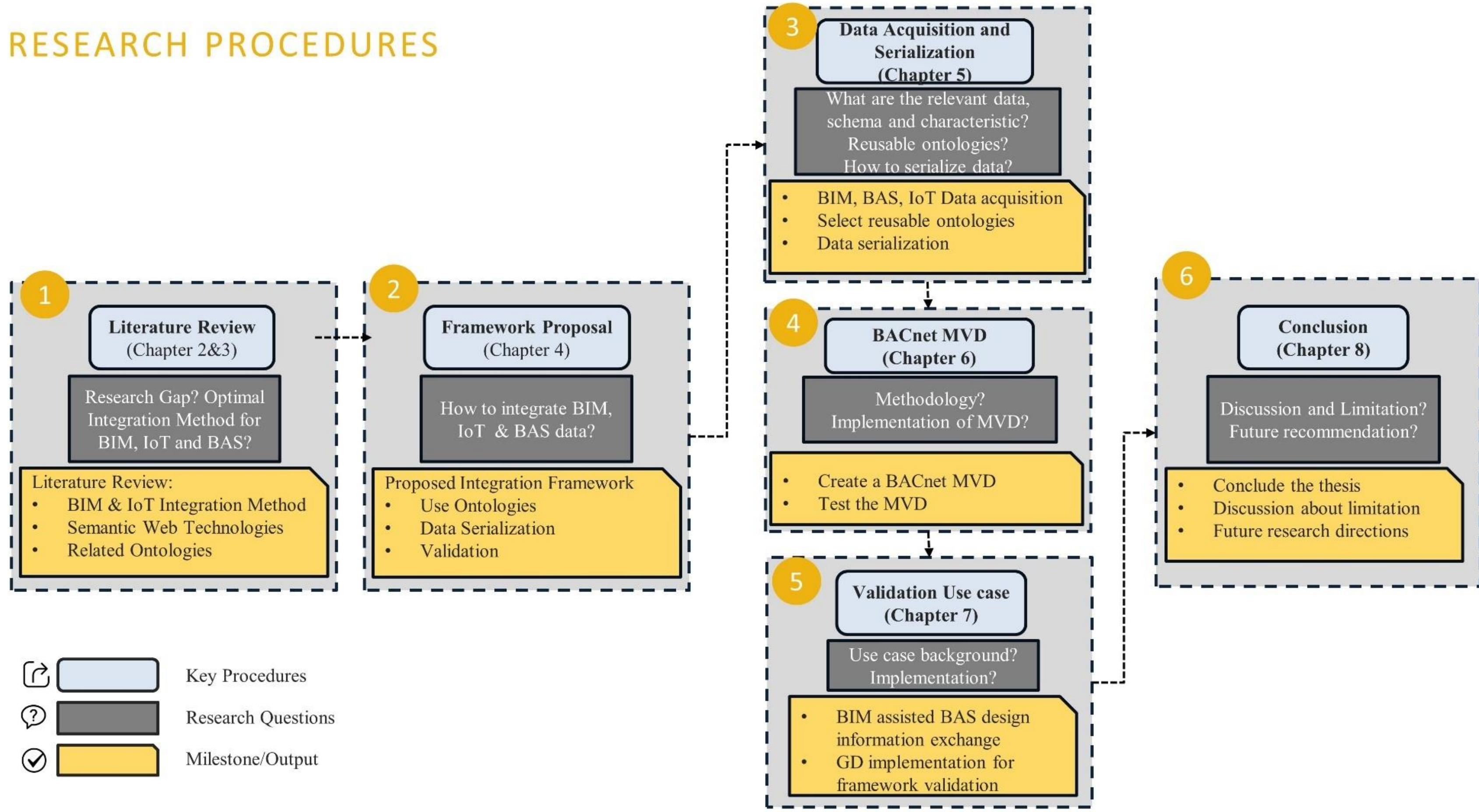


Figure 2.1 Overall Research Procedures

CHAPTER 3 LITERATURE REVIEW

This chapter first conducts a comprehensive review with the intent to identify common emerging areas of application and common design patterns in the approach to tackling BIM-IoT device integration. Instead of focusing on BAS, the author explores BIM and IoT integration in a broad perspective to profoundly discover the integration methods. Besides, this chapter investigates semantic web technologies related to BAS, BIM, and IoT data. Similar previous research studies that focus on BIM and IoT integration for BAS using semantic web technologies are analyzed and discussed in this chapter. This chapter is disseminated in [10].

3.1 Methodology for Literature Review

To ensure a comprehensive review of BIM and IoT device integration, both BIM-related articles from AEC journals and IoT related publications from electrical & electronic engineering and computer science domains were included in the review. To acquire up-to-date and high-quality papers, the following steps were taken: (i) Journal search from Web of Science; (ii) Set journal selection criteria and select high impact journals; (iii) Paper search in individual databases and libraries; (iv) elimination of duplication and irrelevance; (v) Categorization based on the results of content analysis and discussion on reviewed articles.

In the first step, high impact journals in the AEC industry were identified through the Web of Science database, Journal Citation Report, and past review articles' methodology [11–13]. Then, as shown in **Table 3-1**, 14 journals with an impact factor above 1.5 or highly endorsed by professionals were selected. Afterward, advanced searching was operated in individual journal databases including Science Direct, American Society of Civil Engineers (ASCE) Library, Taylor and Francis Online, Wiley Online Library, and ITcon. To consider publication in other industry, Electrical and Electronics Engineers (IEEE) Xplore and Association of Computing Machinery (AMC) library database were searched. Searching keywords were combinations of "Building information modeling", "BIM", "Internet of Things", "IoT", "Smart Building", "Smart City", "Sensor" and "Building Management System (BMS)". As summarized in [11], publications related to BIM has drastically increased since 2012. Publications related to IoT device integration with BIM were rare before 2012 in all searched databases. In this review, articles were limited to those published in the last decade. There were 333 searching results in the selected journals and databases. This review focused on IoT devices integration with BIM for the smart built environment, duplicated and irrelevant papers were eliminated. By reading through articles' abstracts, highlights, and key scope, this screening process reduced the article number to 97. **Table 3-1** summarized the article selection details. The final step was categorization and analysis of the reviewed articles. The categories in this paper were created based on articles' content and frequency of keywords. The authors analyzed these articles based on various domains of use and integration methods.

Table 3-1 Searching result and reviewed articles

Journals/Database	Total searching	Reviewed Paper
Automation in Construction	85	48
Advanced Engineering Informatics	25	8
Journal of Computing in Civil Engineering	53	6
Journal of Management in Engineering	12	3

Table 3-1 Continued

Journal of Construction Engineering and Management	13	2
Energy and Buildings	25	3
Journal of Information Technology in Construction	29	3
Building and Environment	7	3
Journal of Civil Engineering and Management	5	2
Building Research and Information	11	1
International Journal of Project Management	2	1
Expert Systems and Application	1	0
Project Management Journal	0	0
Computer-Aided Design	0	0
IEEE Xplore Digital Library	22	9
ACM Digital Library	43	8
Total	**333**	**97**

3.2 BIM and IoT Integration for Different Domains

To identify and provide a more in-depth analysis of the reviewed papers, the author proposed several domains based on reviewed papers' research content and keywords. The overall categorization of the reviewed paper based on domains is shown in **Table 3-2**. For each of the domains, representative research works are described and discussed in this section.

Table 3-2 Categorization of reviewed papers by domains

Domains	Subdomains	Related Research
Construction Operation and Monitoring	On-site environment monitoring	[14,15,24–27,16–23]
	Resource monitoring	
	Communication and collaboration	
	Construction performance and progress monitoring	
Health and Safety (H&S) Management	H&S training	[20–22,28–34]
	On-site monitoring for H&S	

Table 3-2 Continued

Construction Logistic and Management	Automation in prefabrication	[35,36,45,46,37–44]
	Lean construction	
Facility Management (FM)	Building Operation and Maintenance (O&M)	[47–56]
	Building Performance Management	[5,6,12,50,57–62]
	Energy Management	[63,64,73,65–72]
	Disaster and Emergency Response	[56,69,74–78]

3.2.1 Domain: Construction Operation and Monitoring

The construction industry has experienced continuous digital transformation. The emergence of BIM and IoT devices brings the integration of real-time data like environmental data and localization data to assist construction operations and management. As sensors and BIM integration can accomplish real-time information sharing and communication, construction monitoring gets benefits in various aspects. These aspects include:

<u>On-site environment monitoring:</u> On-site environment monitoring has adopted sensors and BIM. One noticeable usage was real-time site visualization through Virtual Reality (VR) [14,15]. Another aspect was the automatic equipment operation. Real-time sensory inputs that captured realistic environmental conditions were combined with BIM models to calculate the equipment operator's instruction, compactor's path [16], and automatic crane operations [17,18].

<u>Resource monitoring:</u> Sensors like Bluetooth Low Energy sensors and motion sensors were used to track the movement of labors, materials, equipment in complex construction sites [19–22]. These tracking systems incorporated BIM models to visualize

moving paths, hence resource status and labor behaviors could be monitored.

Communication and Collaboration: The study from Ibem and Laryea implied that construction operation could be enhanced using real-time communication and collaboration with BIM and IoT devices [23]. [24] Also supported this opinion by suggesting that the BIM model could be correlated with real-time construction data captured from IoT tags, sensors, and mobile devices to enable in-time communication and collaboration among various parties.

Construction performance and progress monitoring: Construction performance and progress monitoring can profit from IoT devices and BIM in many aspects. Firstly, reality data including actual performance, project status (e.g. laser-scanned point-cloud data), construction activity and physical context, and other real-time project information could be captured with sensors. Integrated with models and BIM tools, these data could be leveraged to monitor the construction process and update construction schedules [24–26]. Besides, sensors have been used to detect progress data for quality control. For example, Radio-Frequency Identification (RFID) and GPS sensors were used to collect position data of construction components for comparison against BIM models [27].

The current research showed some limitations in construction operation and monitoring. Firstly, some of the reviewed articles proposed framework or workflow with only the prototype tests, single use case for a company, or limit testing scenarios [15,21,22,25,26]. Whether these solutions can be generalized or not remains to be investigated. In addition, some prototype tests [15] were based on heavy and expensive equipment in the laboratory which failed to consider the applicability in the real construction site. Furthermore, issues like the cumbersome design of the process, manual data conversion [26], the reliability of sensor collected data for calculating mechanisms [19,22,27] remains to be streamlined and addressed. Research studies have been successful on a specific use case, like visualization, crane operation, location tracking, and risk warning, however, these solutions are fragmented and there is a lack of a more cohesive framework that integrates various sensor's collected data to manifest smart construction site in the future.

3.2.2 Domain: Health and Safety Management

Health and safety are specialized high impact concerns, due to the legislative and risk exposure associated with both worker and occupant health conditions. IoT data sources are providing widely useful for detailed monitoring of environmental and human activities associated with health and safety risks, as well as programs for continuous improvement, insurance optimization, and compliance, etc. BIM datasets contain rich information on building equipment and organizational conditions (proximity, connectedness, etc.) that can be leveraged to provide context for these sensor-based data streams. Some of the prevalent applications integrating BIM and IoT data for health and safety management include:

<u>H&S Training:</u> Both in [29] and [30]' s research, H&S training systems that used BIM and sensors like Ultra-Wideband (UWB) technology were proposed. The location of trainers, trainees, materials, and equipment was tracked during the training course. Sensor collected location data and BIM models were used to analyze safety and productivity, meanwhile providing real-time and post-event visualization through a VR environment.

<u>On-site monitoring for H&S:</u> Sensors and BIM integration for H&S have been

widely implemented for safety matters. IoT devices were integrated with BIM tools to achieve real-time data queries, risk identification, visualization, and notification over BIM models. Sensor networks and BIM model were used to avoid risk in complex and confine-spaced construction sites [21,22,34]. Structure monitoring sensors, RFID tags, and BIM models enabled visualization of the malfunctioned component for structure health monitoring [31–33]. Together with the surrounding environment monitoring system, further improvement could be realized by adding a portable warning device for works to achieve early hazard warning [20].

Several limitations were identified in these studies: i) The majority of H&S applications leveraged RIDF tags and BIM for monitoring location data and sending out warnings [21,28]. However, safety issues were not only related to location data, how to identify works behaviors [30] and other potential safety hazards [31] required deployment of various types of sensors; ii) For most of the structural health monitoring research, proposed solutions were only tested with limited scenarios, further research is needed to investigate their scalability and reliability [31–33]; iii) Another issue would be the ease of implementation of the proposed solutions, considering workers' privacy in most of location-tracking required solutions; iv) Several researchers mentioned the limitation in sensor reliability and energy efficiency of battery [21,22]. A solution to this limitation would be necessary for long-term and sustainable implementation.

3.2.3 Domain: Construction Logistic and Management

In terms of Construction Logistic and Management, BIM and IoT device integration can largely facilitate automation in prefabrication and lean construction management.

Automation in Prefabrication: Advanced sensor technologies and BIM facilitate automation in prefabrication. BIM and IoT devices like RFID tags are effective tools for prefabricated component manufacturing, logistic, tracking, visualization with BIM model, and automatic assembly. Various research works have proposed solutions to demonstrate applications in prefabrication [35–42].

Lean Construction: Digital technologies like IoT and BIM can assess work progress, constraints, and productivity using constant and reliable information flow. This character expedites the development of Lean Construction. Studies like [43–46] stated that the use of IoT devices such as sensors, actuators, auto-ID system networks enable real-time information exchange. Together with BIM models, communication between systems, humans, and devices across the supply chain and construction project lifecycle could be automated.

Limitations for reviewed articles in construction logistic and management: i) Because of sensors like RFID tags deployment, a large amount of data could cause information overload. It was necessary to consider how to standardize construction process data, prefabrication data from vendors and tracking data from sensors to conform to local code and regulations [36]; ii) Solutions' generalizability: the majority of the solutions were conceptually proposed and tested in lab environments and not in real projects [36,44,45]; iii) as current studies were based on conceptual frameworks and prototype test, the challenge to overcome conservatism in the construction industry was critical. The issues like return on investment, human capital, training, technical problem in implementation might hinder implementing innovations [37,44]; iv) for prefabrication construction using RFID tracking, full BIM models were needed during the design phase. Information exchange with BIM models throughout various project stages involved serval parties. Hence, the feasibility and effectiveness of information exchange and collaboration needed further consideration [38].

3.2.4 Domain: Facility Management

Researchers have studied BIM application in supporting FM for years. The integration of BIM and IoT devices can provide facility managers with automated ways for building O&M, building performance management, energy management, and developing disaster & emergency response strategies[79].

<u>Building Operation and Maintenance:</u> BIM integrated with IoT devices can create beneficial platforms for assisting O&M practices such as real-time data access, checking maintainability, creating and updating digital assets, and space management [47]. Studies related to building O&M can be summarized into: i) Identify physical building components and link with BIM models through RFID tags [48,49] for asset tracking; ii) link physical objects with digital objects by linking BMS and BIM [50,51] using BIM tool APIs for O&M; iii) Extract real-time data, visualize problems either from Augmented Reality (AR) devices [52] or BIM tools on mobile devices on facility managers' perspective to conduct maintenance or control of assets [53–55].

<u>Building Performance Management:</u> Publications in BIM related building performance simulation, assessment, optimization, and management have experienced an exponential increase over recent years [57]. BIM enables interoperability, visualization,

automation, and integration with other systems. Consequently, BIM integration with IoT devices can bring ultimate potential in: i) Linking BIM with BMS data for visualization and management of real-time building performance data [50,51,57]; ii) monitoring Indoor Environment Quality (IEQ) by reading temperature and humidity information from BIM tools promptly [58] and improving user comfort [59]; iii) using semantic web technologies for real-time building performance monitoring and assessment [5,6,12,60,61]. Building context data in either BIM models or Industry Foundation Class (IFC) file is adapted to web standard. These studies can adapt cross-domain information and real-time sensor reading to web standards, so that cloud-based applications and services like energy management, building performance monitoring, and building operation management can be fulfilled [62].

Energy Management: Existing research on IoT devices integration with BIM for Energy Management was focused on real-time energy usage visualization and monitoring at building and city scale, energy performance analysis, and energy benchmarking. Researchers tried to propose energy management solutions and study on Wireless Sensor Networks (WSN) integration to achieve the above-mentioned goals. These research were related to: i) web-based energy management solution which allowed facility manager to view BIM models, query sensor data and get actuation suggestions at a "near real-time" basis [64–66]; ii) Geographic Information System (GIS) integrated energy management solutions that are capable of visualizing 3D models and monitoring geospatially referenced energy usage [67]; iii) software/system architecture as energy management solution to monitor real-time energy consumption and building condition with BIM [63,68,69]; iv) energy performance analysis based on real-time energy consumption using co-simulation tools like Building Controls Virtual Test Bed (BCVTB) and Cyber-Physical Building Energy Management System (CBEMS) [80,81] which extracted building data from BIM models; v) integration of WSN and BIM for energy monitoring, real-time feedback & control and energy benchmarking [70–72]. It has come to an agreement that WSN integration with BIM could improve the AEC industry, however, issues like interoperability, return on investment, the limitations with sensor devices still need future studies [73].

Disaster and Emergency Response: BIM and various types of sensors can be effective tools in disaster and emergency response both at the building and urban scale. Researchers have utilized BIM and IoT devices for: i) detecting the indoor location of trapped victims and display location in BIM models [74]; ii) calculating shortest evacuation path with building information from BIM models in real-time and location data from sensors or victims' mobile devices [75–77]; iii) creation of mobile guidance for evacuation with BIM tool API and mobile devices [78]; iv) large-scaled emergency response using BIM, IoT devices and GIS [56,69].

There were 4 limitations identified for FM studies: i) Most of the reviewed articles were theoretical, for example, several studies proposed conceptual frameworks without implementable platforms or systems [48,49,52]; ii) Some researchers only tested their prototypes under laboratory condition or limited and small-scale test scenario [54,56]; iii) Most of the research did not achieve real-time information queries, the research results were either manual information update on static information or near-real-time queries that requires user-defined time interval for information renew; iv) Although many works have achieved information acquisition from BIM and IoT devices, none of the reviewed articles provided solutions on controlling actuators through BIM. As a result, future studies focusing on tackling these limitations are necessary.

3.3 BIM and IoT Devices Integration Methods

While the BIM and IoT device integration is still nascent, there is a need to understand the current situation of BIM and IoT device integration including:

1. What are the integration methods for BIM and IoT devices?
2. What are the limitations of both application domains and integration methods?

This section attempts to address these research questions by conducting a comprehensive review of BIM and IoT device integration with the summarization of the integration methods of existing studies, examination of current limitations.

In this section, five methods were concluded from reviewed articles' methodologies and implementation. The authors described the basic steps for each integration methods, demonstrated how a method was implemented in various examples from reviewed articles and analyzed the pros, cons, and applicability of each method.

When discussing BIM and IoT device integration, there are three components. Firstly, BIM serves as a data repository for contextual information including building geometry, IoT devices' description, static information and other soft building information collected from occupancy patterns and schedule data like social media, building feedback, occupant interactions, room allocation, weather forecast and financial pricing [12]. Contextual information can be stored in BIM tools (e.g. Revit Model) or IFC formats. The second component is the time-series data which records continuous sensor readings [72]. Traditional time-series data is stored in a well-structured relational database and can be effectively queried using Structured Query Language (SQL). The third component is the integration method between contextual information and time-series data. This section explains different integration methods of contextual information (BIM data) and time-series (sensor collected) data.

3.3.1 BIM Tools' APIs + Relational Database

3.3.1.1 Description of this approach

A widely adopted approach is to use existing BIM tools' Application Program Interface (APIs) and relational database. The basic steps as shown in **Figure 3.1** can be concluded as the following points: i) Sensor collected time-series data is stored and updated in relational database (e.g. SQL server database, Microsoft Access); ii) BIM models which are constructed in BIM tools (e.g. Revit), can be exported into relational database formats using APIs (e.g. Revit DB Link, Dynamo, Grasshopper); iii) Define a database schema which clarifies the relationship between virtual objects and physical sensors. For example, virtual objects can relate to physical sensors using unique identification (Global Unique Identifier (GUID or UUID)); iv) A two-way importing and exporting of a relational database and BIM model can be achieved using APIs; v). Processing queries of sensor data through custom-built API (e.g. graphic user interface (GUI) based on Revit), third party processing engine (e.g. Unity engine), and direct query over SQL database (as the object properties).

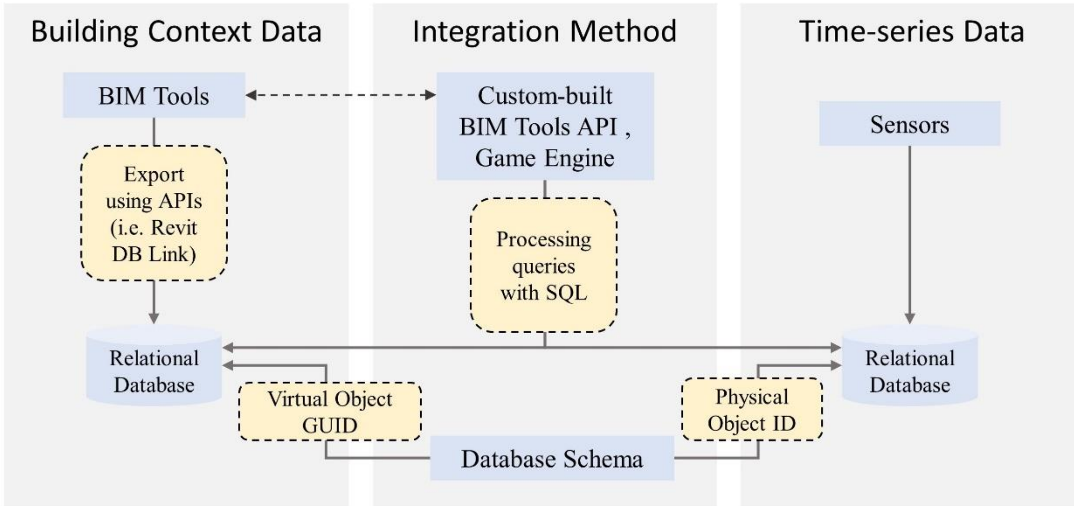


Figure 3.1 BIM Tools' APIs + Relational Database

3.3.1.2 <u>Examples of this approach</u>

The BIM and WSN system proposed by Marzouk et al. [58] allows the user to read temperature and humidity information from the Revit model promptly. Temperature and humidity data were collected by WSN, stored and updated in Microsoft Access relational database. Revit project data (contextual information) were then exported into Microsoft Access format. Data visualization, importing, and exporting were achieved using Revit DB link between Microsoft Access and Revit. Sensor readings were then displayed in Revit as object properties. Zhang and Bai's [33] method can also be categorized into this approach. Revit models were exported to MySQL or MS Access and linked with physical objects using identification. However, this study was only tested with a few structural objects under laboratory conditions. The same problem happens to [32]'s work. If the number of sensors and objects is enormous, a data mapping schema will be necessary to clarify the virtual object and physical objects linking process.

More advanced query processing was done by Arslan et al[21] and Riaz et al. [22]. The real-time H&S monitoring systems also used a relational database and Revit DB link to achieve sensor-BIM integration. They differed from [58]'s work by creating a GUI as a Revit API using C# language. The GUI once invoked can display the latest sensor values. Another example is Woo et al. [70] developed a virtual campus model that integrated BIM, WSN data in a game engine environment for energy benchmarking. WSN was first installed to collect energy data. The Revit DB link plug-in was used to export Revit model into the MySQL database. Sensor data was stored in MS Excel then exported to MS Access relational database before stored in the Revit model. The connection between query and MySQL database was processed by the Unity game engine [14,82] so that real-time energy performance data could be queried.

There were other research works implemented with Grasshopper and Dynamo. Habibi [59]developed a prototype smart system using Grasshopper to obtain and monitor real-time sensor data. A genetic algorithm was then implemented in Grasshopper to identify the optimized solution. An intelligent user interface allows the user to understand real-time sensor data and take actions based on the optimal solutions. [57] created a prototype method linking BIM with BMS data. Revit was firstly used to hold model and building design performance data that had been extracted into JSON using Dynamo. Continuous sensor data was stored in a SQL server. A Python script was used to extract sensor data stored in the SQL database and link it to JSON file.

3.3.1.3 Discussion of this approach

This approach is found in most of the related research, as there are a few obvious advantages. The integration process can be done using the existing APIs. Model data can be exported to open database connectivity (ODBC) format which is compatible with external database software (e.g. MS Access, MySQL). The second advantage is the ease in linking model data and sensor data because of they both store in the relational database. The third, time-series data can be automatically updated in BIM tools with these APIs. However, there are some drawbacks. Since only shared parameters between projects and families can be exported, there will be limitations on what to update. Besides, although sensor data can be automatically updated in the relational database, manual export of model files incurs repetitively if model change happens.

This approach is suitable for less complex BIM models with a limited number of sensors, as this approach requires the construction of virtual objects to represent physical sensors manually. It is particularly useful to integrate existing BMS with BIM. Since there are available API to export BIM data into the relational data model, it requires less expertise in IFC and programming and provides ease of use for wider adoption.

3.3.2 Transform BIM data into Relational Database using New Data Schema

3.3.2.1.1 Description of this approach

One effective way to integrate sensor data with BIM is to transform BIM data into a queryable database that allows information extraction from different users' perspective (see **Figure 3.2**). Traditional building management system store data (e.g. facility data and sensor readings.) in a relational database that is well structured and effective for SQL queries. Transforming BIM data into relational database structured data is the foundation step for binding time-series data with BIM. Once BIM data is SQL queryable, sensor data can be linked to BIM, for example connecting virtual sensor objects with physical sensors via GUID, and map sensor collected data as virtual sensor objects' properties.

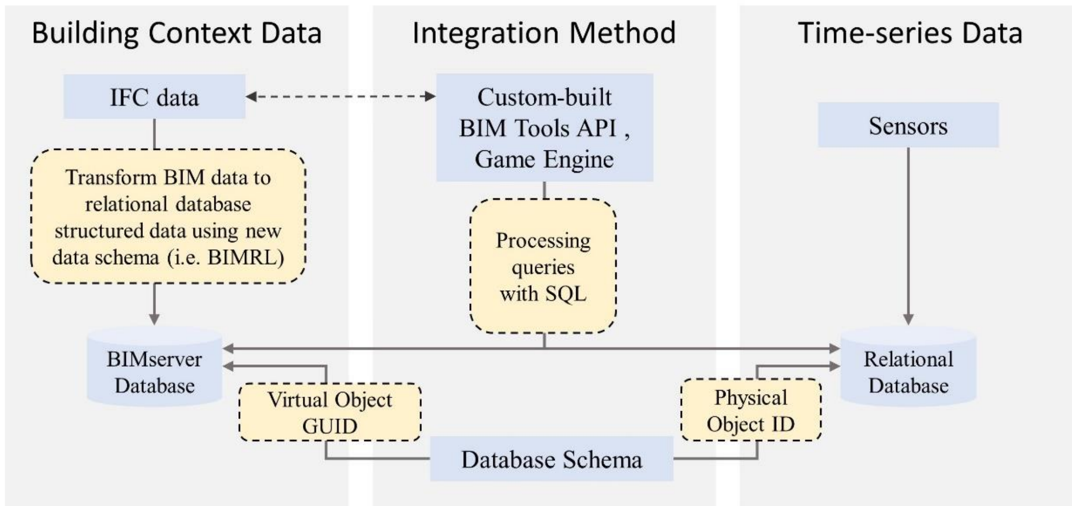


Figure 3.2 Transform BIM data into Relational Database using New Data Schema

3.3.2.1.2 Examples of this approach

Solihin et al. [83] created a new relational database schema named BIMRL to transform BIM data into a queryable database. BIMRL allowed efficient SQL queries on BIM data without storing entire IFC-STEP data in a database, BIMRL schema was generated using the relational database structure following the star schema definition in the Data Warehouse (DW) domain. BIMRL extended BIM data query capability by integrating spatial set operations into standardized SQL. BIMRL showed potential applications in rule checking, facility management, sensor data integration, and real-time optimization of building operations.

The prototype system in [54] also converted BIM data into a relational database through manipulation on IFC. Various sources of facilities' data (e.g. sensor readings stored in the relational database) were integrated with BIM. Logical and spatial relationships between entities were manually added to the IFC file and mapped to Microsoft Access relational database. GUIDs were used to link entities, related attributes, data sources throughout the whole system. User applications can be instantiated using SQL queries based on the manually coded relationships.

Kang and Choi [53] tried to transform BIM data into a database based on a proposed BIM perspective definition (BPD) metadata structure. The BPD metadata structure was used to bind BIM and facility management, and achieve data extraction based on users' queries. Multi-sources data including BIM objects, facility management data, and sensor data was extracted, transformed, and loaded (ETL) into a DW database based on BPD metadata structure. Data were linked with BIM objects via OBJECT_GUID and expressed as BIM object properties. Users were able to view requested metadata in XML format based on their purposes.

Another example was the IFC-based graph data model for topological queries created by [84]. The research proposed a new schema named graph data model extract analysis and represent topological relationships among 3D objects. Although this new data model did not transform the BIM model into a relational database, it enabled topological queries on building elements (e.g. virtual sensor objects).

3.3.2.1.3 Discussion of this approach

As a new schema or data structure is defined based on the user's perspective, this approach shows its flexibility in expanding users' perspectives while effectively extracting external system data. Time-series data retains in its original database. This approach can use existing SQL in normal database management system platforms, avoiding rewriting query interface from scratch. However, creating a new data schema requires significant efforts in mapping data which is time-consuming. Also, manipulation of SQL is necessary if special queries or operations are needed.

Although this approach requires constructing virtual objects to present physical sensors manually, it is more flexible for complex projects with complicated spatial context and a large number of sensors. The reason is that a new schema or data structure is designed based on the user's perspective. Exporting entire complex IFC data into a queryable structure is unnecessary. Compared to using existing BIM tool APIs, this method requires more expertise in language design, IFC, database, and programming knowledge.

3.3.3 Create a New Query Language

3.3.3.1 Description of this approach

Another approach identified from reviewed papers is to create a new query language to query sensor data over BIM models or IFC models instead of using SQL. As shown in **Figure 3.3** the newly developed query language is used to developing queries that process time-series data.

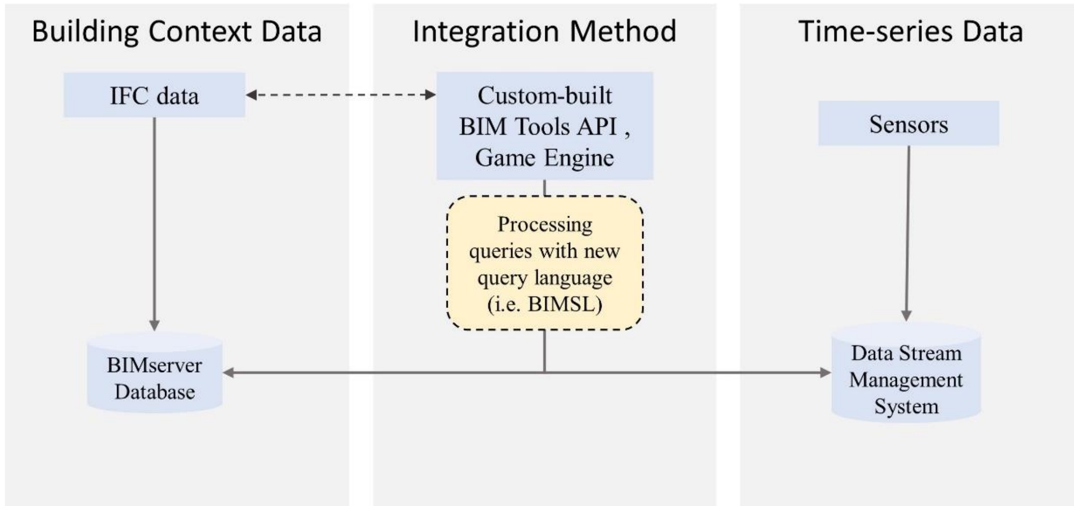


Figure 3.3 Create a New Query Language

3.3.3.2 Examples of this approach

One example could be found in Mazairac and Beetz's [85] study. They proposed a domain-specific query language named BIMQL to select and partially modify IFC-based BIM models. BIMQL allowed selection of objects and attributes based on schema names or arbitrary properties for example IfcSensor. However, this method had a limitation in query real-time sensor data. The language was developed to query IFC based BIM model, only static sensor information stored in IFC-based BIM model can be queried. Since it is almost impossible to transform real-time sensor data into IFC format and store in the BIM model, this solution had a limitation in the integration of real-time sensor data with BIM.

This problem was noticed by Alves et al.[86], so that they created another domain-specific query language named BIMSL. The implementation of the BIMSL language was through a custom-developed API named BIMSL API. BIMSL queries contained both contextual information query and dynamic components related to time-series data. While the contextual information query could be directly fed to the BIMserver database, the real-time data query utilized the open-source Java-based Esper engine for event stream processing. Esper software has its query language named event processing language (EPL) which is a SQL-standard language, the dynamic component of BIMSL invoked EPL queries to return real-time sensor readings.

3.3.3.3 Discussion of this approach

The purpose of developing a new query language over time-series sensor data is steadfast, so the advantage in expressiveness and ease of use is obvious when compared to general-purpose language. However, the newly developed query languages either lack real-time sensor data query capability or used external ELP. Moreover, to implement a newly developed query language about real-time sensor data and BIM, a corresponding platform need to be developed at the same time [86]. Furthermore, new query languages that are not standardized may not be widely adopted. There are standardized query languages (e.g. SQL, SPARQL) that have already been maturely implemented to satisfy the query needs. The standardized query languages function more effectively among various tools and users in different domains than new query languages.

This approach is seldom discussed in the reviewed articles. This approach retains both contextual data and time-series data in their original form and does not require heavy modeling or data mapping, it can be applied to various kinds of projects. However, it requires knowledge of language design and API programming to realize functions.

3.3.4 Semantic Web Approach

3.3.4.1 Description of this approach

In modern AEC processes, data sets such as building geometry and topology data, sensor data, behavior data, geospatial information data, are generated and consumed across a building's lifecycle. The integration of BIM and Semantic Web Technologies has the potential to meet the requirements for storing, sharing, and using heterogeneous data sets. The key concept is to have those data sets to be represented as or tagged using RDF. Linking BIM ontologies e.g. IfcOWL and ontologies in other domain e.g. Smart Appliances Reference ontology (SAREF), SSN for sensor devices domain is an effective approach for BIM and IoT device integration.

This approach requires both building context data and time-series sensor data to be represented in a homogenous format (see **Figure 3.4**). The basic steps are listed as follows: i) represent contextual information including building context data, sensor information, and other soft building information into a web interchanging standard named RDF using semantic web approach; ii). Sensor collected time-series data is extracted from the relational database and represented to RDF format using the semantic web approach; iii). Linking data silos across different domains via unique identification; iv). Conduct contextual building information queries or real-time sensor data queries using an RDF query language named SPARQL; v). Query results can be shown on applications in different forms such as Command Line Interface (CLI), dashboards, GUI, API, and other tools.

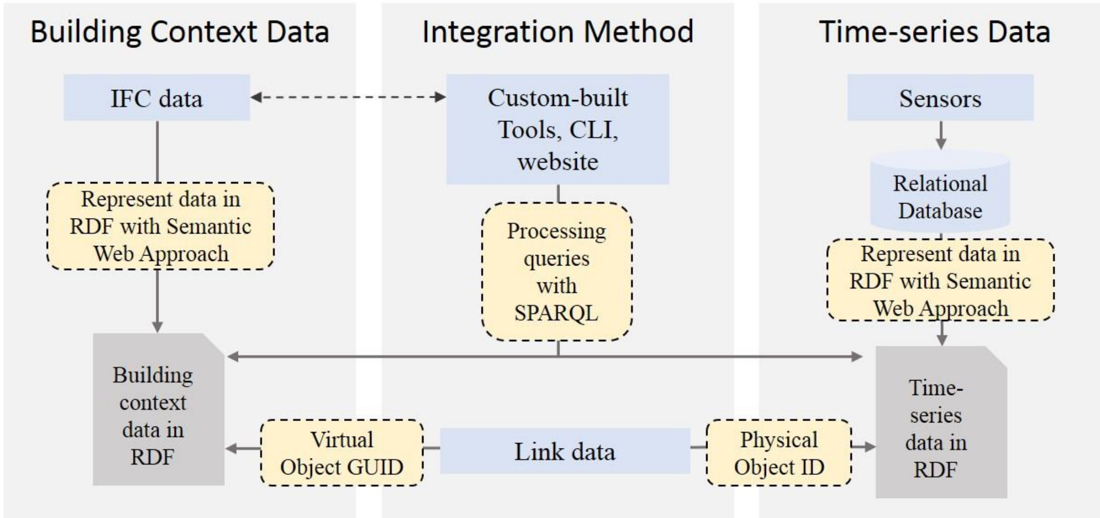


Figure 3.4 Using Semantic Web Technologies

3.3.4.2 Examples of this approach

The ontology framework for intelligent sensor-based building monitoring fitted into this approach [61]. The framework named OntoFM contained a building ontology-based on IFC, a sensors ontology generated from OntoSensor and a general-purpose ontology for domain-independent concepts capture. IFC was used to represent building geometry and converted to Web Ontology Language (OWL). SPARQL was used to conduct ontology queries. However, this study focused on the process of ontology development rather than how sensor real-time data was represented into RDF.

A more explicit example from Curry et al.[62] was using RDF and Linked Data in the cloud to integrate cross-domain building data. Building context data representing in IFC was first converted into RDF. All data related to building operations (sensor data) need to be expressed in RDF. Universal Resource Identifiers (URIs) were used to globally identify resources and associate isolated data silos. SPARQL was implemented to query

RDF data and resulting data can be visualized via applications. The authors stated that SPARQL queries can be translated from other query languages such as SQL and XQuery, while Hu et al. [5] held an opposite point of view as discussed in **Section 3.3.5**.

3.3.4.3 Discussion of this approach

This approach shows its advantage in linking cross-domain data in a homogeneous format and the ease of interlinking silos. Although some existing data silos can be directly utilized for various purposes, this approach can be problematic. These issues are: i) most of time-series sensor data was stored in well-structured and relatively mature relational database, the way the relational database store sensor data is more effective for query than store sensor data in RDF format; ii) Duplication of data may incur when converting time-series data into RDF; iii) The performance of RDF representing fixed-structured data is inefficient and storage consuming [5].

Although this approach requires modeling virtual objects, knowledge of semantic web technologies, and heavy data transformation, it is useful for projects with a broader scope that connects various kinds of data sources. Since data silos can be represented in RDF format, this approach extends the possibility to achieve the real concept of IoT, which requires interlinking with the internet through a unified framework. However, heavy data transformation is cumbersome for complex building and BMS with continuous real-time readings.

3.3.5 *Hybrid approach: Semantic Web + Relational database*

3.3.5.1 Description of this approach

In this approach, both the Semantic Web and relational databases are used to store cross-domain data. The authors conclude the key steps to implement this approach (see **Figure 3.5**) as follows: i) represent contextual information including building context data, sensor information, and other soft building information into RDF format using semantic web approach; ii) Retain sensor collected time-series data in the relational database; iii) Map contextual information with time-series data, in particular, time-series data can be referenced using sensor ID described in RDF.

This approach brings two technologies together, hence results in integrated query methods. Contextual information represented in RDF is queried by SPARQL, while time-series data stored in the relational database is queried using SQL. Since contextual information and time-series data are mapped, SQL queries can be created using SPARQL queries on RDF data.

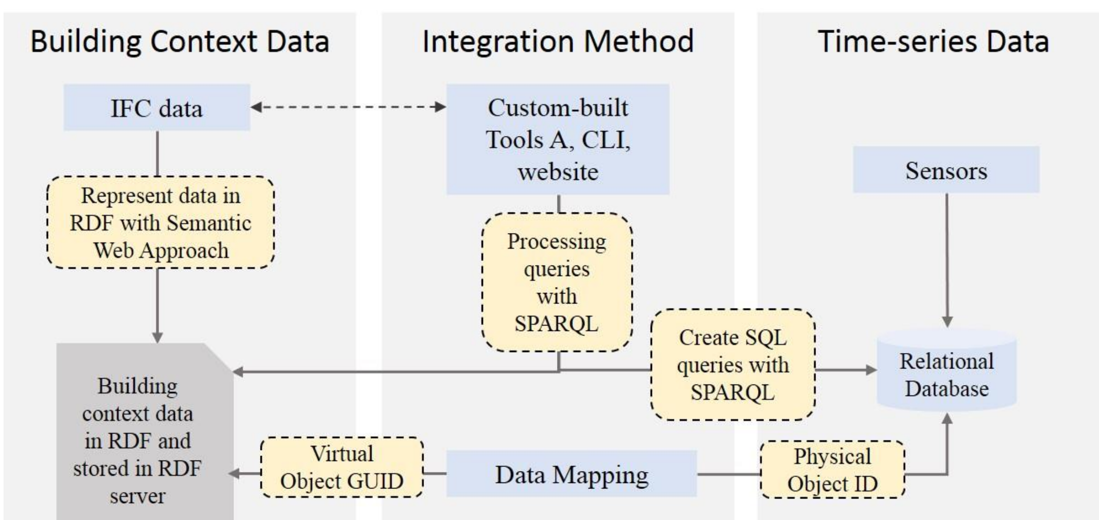


Figure 3.5 Hybrid Approach: Semantic Web + Relational Database

3.3.5.2 Examples of this approach

Hu et.al [5] introduced a hybrid architecture that integrated building performance data with semantically-described building context data. In this hybrid architecture, contextual building data – typically represented as BIM model or IFC format – converted to RDF using the semantic web approach. Static sensor information such as sensor type, vender, and identification was also converted to RDF format using SSN ontology. However, sensor collected time-series data was stored in a relational database and maintained its original form. Time series data (sensor collected data) was then cross-referenced with SSN ontology using sensor ID. In this way, building data and time-series data stayed in their appropriate platform and format. The same approach has also been implemented in another example. McGlinn et al.[64] presented a building energy management solution that used Semantic Web and Relational database technology to integrate BIM, sensor data, and actuator infrastructure. The solution contained a knowledge base as a central integration component of heterogeneous data sources. Building a semantic model that represented the RDF format was uploaded to a SPARQL server. Application in a proposed interface was used to conduct SPARQL queries about building elements. Instead of converting sensor data into RDF format, sensor data was stored in the relational database and ID-referenced with the semantic model. The sensor monitoring application could query sensors' time-series data via SQL queries.

3.3.5.3 Discussion of this approach

The highlight of this approach is that different data sources retain their most suitable platforms and formats while achieving interlinking. This approach retains the effectiveness of storing time-series data in the relational model, flexibility to link building contextual data using a semantic web approach and query using standardized language without heavy data conversion. The highlight of this approach contributes to several advantages: i) Time-saving: as time-series data still stored in the relational database, duplicating time-series data into RDF format is not necessary; ii) Storage saving: the way that RDF data is stored (triple stores) requires more storage than relational database; iii) Better performance: relational database performs better in data lookup than triple stores [5]; iv) Effective query language: the integrated query method use existing SPARQL and SQL to query RDF data and sensor data respectively.

This approach is one of the most promising methods to facilitate IoT deployment in the construction industry. It retains different data sources in their most suitable platforms and formats while achieving interlinking with the Internet to achieve the real concept of IoT. It is suitable for different kinds of projects without heavy data conversion. As this approach utilized standardized data formats and query language, it offers an opportunity to integrate other domain data sources to extend project scope.

3.4 Semantic Web and Linked Data

The semantic web is a web of data in which all information is described in machine-readable language, so that cross-domain knowledge can be represented and connected flexibly and generically. It enables data storage on the web, building vocabularies, writing rules for data handling, and query information [87]. In this semantic network, information is represented and combined using labeled graphs with nodes and labeled arc. The nodes in graphs represent a concept or object in reality while the edges represent logical relations between the nodes [88]. Another term that frequently relates to the semantic web is linked data. Linked data requires data to be published on the web while linking to the outside data which is the core idea behind the semantic web. Linked data can be considered as part of the semantic web when data is cross-domain linked [89]. Linked data are empowered by semantic web technologies like Resource Description Framework (RDF), Simple Protocol and RDF Query Language (SPARQL), RDF Schema (RDFS), Web Ontology Language (OWL), Rule Interchange Format (RIF), Semantic Web Rule Language (SWRL), and Simple Knowledge Organization System (SKOS).

3.4.1 Semantic Web Technologies

3.4.1.1 Resource Description Framework

RDF is a standard graph data model, which is the foundation for describing semantic domain knowledge in a uniform representation. Thus, the semantic web is an information network of labeled RDF graphs. The nodes represent concepts or objects in reality while the edges are the connections between the nodes, representing relations between them. The nodes and edges, assigned with unique identification namely the Unique Resource Identifier (URI), form the RDF graphs. Logical statements named RDF triples, which are consisted by subject, predicate and object (**Figure 3.6**), construct an RDF graph. The URIs are assigned to subjects, predicated, and objects. The subjects are the resource being described while the predicates are the properties of a resource. The objects are the values of properties (predicates), which can be represented either as URIs or literals (text, numbers, date, etc.).

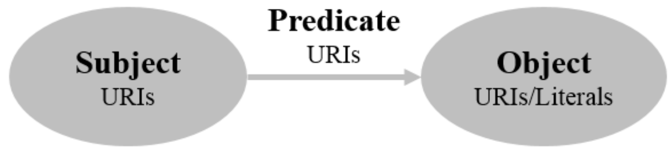


Figure 3.6 RDF Triple

The RDF graph can be represented in a various syntax like RDF/XML (*.RDF), N-Triple (*.N_T), Turtle (*.TTL-), Notation-3 (*.N3), and TriG (*.trig). **Figure 3.7** shows examples of RDF serializations in Turtle and RDF/XML.

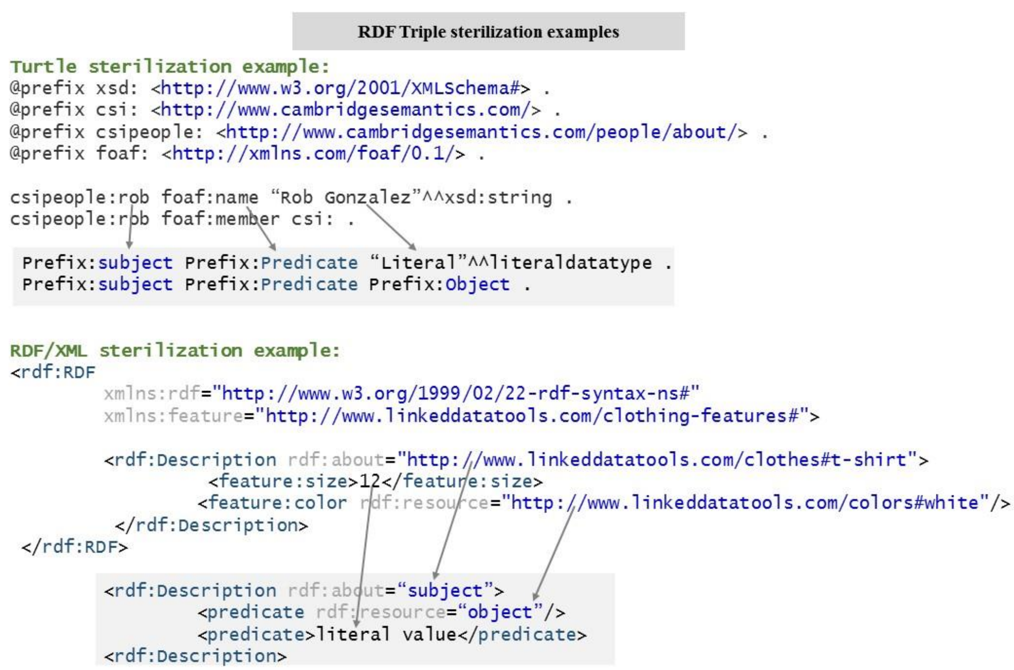


Figure 3.7 RDF Serialization Examples

3.4.1.2 Ontology, RDFS, OWL, RIF, SWRL, and SPARQL

The semantic structure of RDF graphs can be further described using ontologies or

RDF vocabularies. Vocabulary is a collection of terms used to describe and represent an area of concern in a consistent form across context. Ontologies add contextual relationships behind defined vocabularies. Ontologies are used to classify the concepts and relationships in a particular application, characterize relationships, and define constraints in between. Standard vocabularies or formal ontologies are available for some domains of knowledge like the Dublin Core Metadata Initiative (DCMI) and the Friend Of A Friend (FOAF). Since ontologies describe concepts and relationships between them, they act as cornerstones of the Linked Data initiatives for formally represent data on the web. The basic components of ontologies are classes (concepts, type of objects), properties (attributes, parameters of objects), individual (instances or objects), relations, and axioms (assertions and rules).

Ontologies can be implemented by various syntaxes (languages). There are many formal syntaxes (standard format of vocabularies and logic) to define and describe ontologies, for example, RDFS, OWL, RIF, SWRL, and SKOS [87]. RDFS provides the specification of class, subclass, comments, and data types. OWL further extends RDFS with Description Logic (DL) which is the logical concepts allowing a description of more complex RDF statements and restrictions. RIF and SWRL add rules for concepts and relations in an IF-THEN form.

Similar to SQL for relational database and XQuery for XML, SPARQL is the query language to create, read, update, and delete RDF data.

3.4.2 Ontology for BAS/BMS Metadata

Many buildings today have BAS to monitor building subsystems through sensors, actuators, and networking. These sensors, actuators, and control points in the BAS are normally referred as “data points” whose metadata is described with “labels”. The data points’ information such as location, type, function, and subsystem relations is revealed from their labels as alphanumeric representation. Hence, the BAS metadata in a form of alphanumeric labels is neither standardized nor machine-readable. Since the alphanumeric representation can be custom-made, it is inconsistent between vendors and buildings, thus add difficulty to understand. Standardized semantic models that describe BAS metadata are necessary. Several projects have focused on developing semantic metadata models for BAS.

3.4.2.1 Project Haystack

Project Haystack is an open-source initiative to create standardized sets of semantic tags to model BAS metadata. It defines standardized vocabularies of tags to describe building data points. This tagging system provides a consistent naming convention to all building sensors, equipment, and control variables. Haystack describes building data points in a flat meta-model: tag model. Tags are name/value pairs, which can be associated with any entity like a sensor. Haystack defines several basic concepts: i) Entity: an entity is an abstraction of a physical object in reality. Entities, for example, are sites, equipment, sensor points, weather stations, etc; ii) Tags: tags are name/value pairs applied to an entity. They also define attributes about an entity. Tags have Tag Names and Tag kinds (value type of tags); iii) Structure: the primary structure of the Haystack model is based on three basic entities namely `Site`, `Equip`, and `Point` in a hierarchy way. Relationships between entities can be represented using the `ref` tag. **Figure 3.8** shows an example of Haystack Tags and Structure. Each box represents an entity and associated attributes. Relationships between entities are described using ‘`ref: @entityid`’.

The previous study also explored wrapping Project Haystack vocabulary into an ontology named Haystack Tagging Ontology (HTO) [90]. HTO creates ontology class for each entity and tag, and model relations using ontology properties.

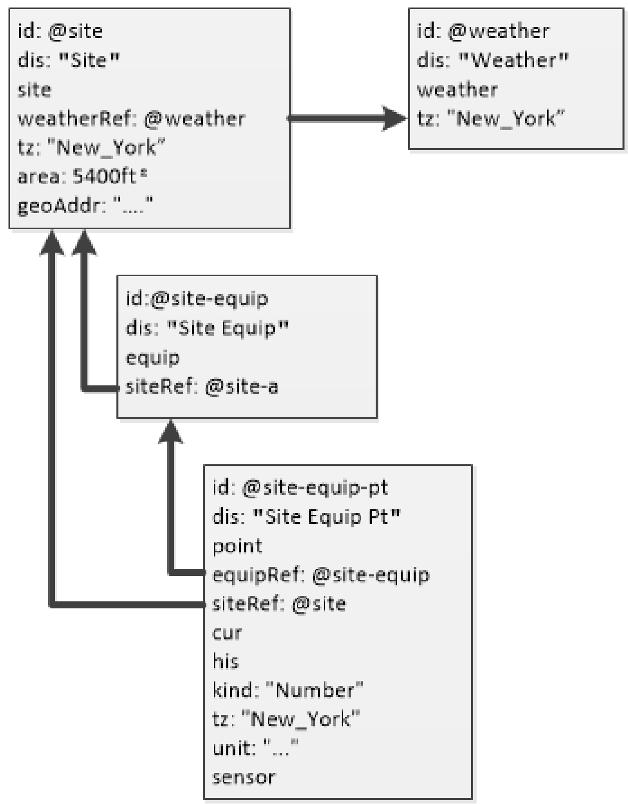


Figure 3.8 Project Haystack Tagging Meta-model Example [91]

3.4.2.2 BRICK

BRICK schema [9] borrowed the concept tags from Project Haystack and enhanced the schema with an underlying ontology that enriched with concepts from the tags, relationships, hierarchies, and properties. The BRICK schema defines some high-level classes and relationships between them as shown in **Figure 3.9**. Nodes represent classes, while relationships and their inverse relationships are labels above the edges. Some basic definitions are:

- Point: are the physical or virtual entities that create time-series data. For example, points are sensors, data points, and set points.
- Equipment: physical devices controlled by points belonging to it. For example, fan AHU, light.
- Location: space, area in the buildings. For example, rooms, floors.
- Resource: physical resource or material controlled by equipment and measured by points. For example, air, water, electricity.

Apart from the relationships in **Figure 3.9**, BRICK schema also reused common relationships defined in RDF, RDFS, and OWL like `rdfs:subClassOf` and `rdf:type`.

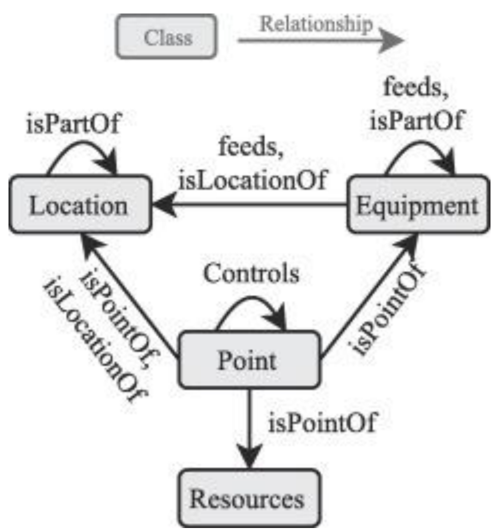


Figure 3.9 Classes and Relationships in BRICK schema [9]

Since BRICK schema is an ontology, actual building BAS metadata using BRICK schema can be modeled as collection triples in RDF. BAS metadata modeled with BRICK schema can be queried using SPARQL.

3.4.2.3 ASHRAE Standard 223P

Starting from early 2018, an ongoing collaboration between ASHRAE, Project Haystack, and Brick Schema Initiative aims at developing a standardized application data modeling solution across the building industry. The ASHRAE Standard 223P will provide a dictionary of semantic tags for descriptive tagging of building data including BAS and control data. In this way, different BAS vendors can follow the semantic tagging standard to exchange data over various communication protocols and enable broader interoperability among applications [92][93].

3.4.2.4 Other related Ontologies for BAS metadata

Smart Appliance Reference (SAREF) [94] aims to semantically describe home automation where capture generic sensors and smart devices. W3C Consortium creates the Semantic Sensor Network Ontology (SSN) [95] to describe sensors and their observations.

3.4.3 Ontology for Building Contextual Information

The most commonly used data exchange format for open BIM is IFC that has been accepted as ISO 16739 standard [96]. Building contextual information such as topology, geometry, location, material properties, as-built construction details, HVAC specification, and spatial relationships can be effectively represented using IFC [5].

3.4.3.1 ifcOWL

buildingSMART International released an OWL representation of IFC schema named ifcOWL. It is generated directly from the EXPRESS schema. Similar to EXPRESS schema for IFC STEP Physical File (IFC/SPF) and XSD schema for XML, ifcOWL is the schema to represent building data using RDF serialization. The ultimate goal is to enable data linking between building contextual data represented using IFC with data in other domains like sensors, GIS, material, etc. Tools (IFC-to-RDF converter) are available to convert IFC/SPF files into RDF serializations like Turtle files [88].

3.4.3.2 Linked Building Data ontologies

While the ifcOWL ontology directly generated from IFC EXPRESS schema, other development initiatives focus on building ontologies inspired by IFC instead of direct translation. These initiatives mainly focused on utilizing existing W3C's Linked Building Data (LBD). The four main ontologies that capture most IFC data sets are Building Topology Ontology (BOT), Product Ontology (PRODUCT), Property Ontology (PROP), and Geometry Ontology (GEOM). BOT captures most IFC building topologies and can be further extended by PRODUCT, PSET, and GEOM to capture the semantics of tangible objects, product data, and 3D building data [97]. BOT allows the definition of building's topology with BOT classes like bot:Site, bot:Building, bot:Storey, bot:Element and bot:containsElement. The PRODUCT ontology aims to describe individual building objects while the PSET ontology is used to assign properties to building elements. The GEOM ontology captures 2D and 3D geometric representations of concepts and elements in buildings. Some concepts in ifcOWL can be directly aligned with BOT concepts as shown below:

- ifc:IfcSite→bot:Site
- ifc:IfcBuilding→bot:Building

- ifc:IfcBuildingStorey→bot:Storey
- ifc:IfcSpace→bot:Space
- ifc:IfcElement→bot:Element

3.4.4 *Ontology for IoT Device Communication*

IoT device contextual information such as identification, type, vendors, and other properties can be stored in the BIM model as virtual objects or can be represented as a collection of BACnet objects. Although BAS metadata ontologies like BRICK schema is capable of describing IoT device contextual in RDF, there are some ontologies developed particularly for sensor and actuator description.

3.4.4.1 Semantic Sensor Network Ontology (SSN) & Sensor, Observation, Sample and Actuator (SOSA) Ontology

W3C consortium introduced SSN and SOSA ontologies to represent the entities, relations, and activities involved in sensing, sampling, and actuation. The objective is to describe sensors and their observations, the involved procedures, the studied feature of interest, the samples used to do so, and the observed properties, as well as actuators. These ontologies create a set of classes such as actuator, sensor, system along with a set of object properties like isPropertyOf, isResultOf, and a set of datatype properties [95].

3.4.4.2 BACnet OWL (BACowl)

There is a non-official effort in creating a BACnet ontology named BACowl. The BACowl specification follows the BACnet protocol by creating primitive data, composite data, objects, and protocol data unit (PDU). Since it is not part of the ASHRAE standard, the implementation is limited [98].

3.5 Similar Previous Research

As summarized in **Section 3.3**, the author observed five methods of integrated building contextual information in BIM and time-series data generated by IoT devices. The hybrid approach that uses both semantic web technology and relational database for BIM and IoT integration is considered the most appropriate way. Detail description of the hybrid approach, examples, and discussion is in **Section 3.3.5**. As shown in [64][5], these research studies utilized the hybrid approach to present building context information and IoT time-series data. However, these studies do not include the BAS metadata description. These studies are neither focusing on BAS metadata nor using the most suitable standardized data models for integration. Some technologies used in these studies are out of date. This research aims to develop a framework to facilitate information exchange between BIM-based building contextual data, IoT devices' time-series data, and BAS metadata using Semantic Web technology. The author proposed a framework based on the hybrid approach and adding descriptions of BAS metadata using standardized data schema to facilitate further cross-domain data interlinking.

CHAPTER 4 BIM, IOT, BAS INFORMATION EXCHANGE FRAMEWORK

This chapter describes the overall architecture of the proposed framework. This research aims to develop a framework to facilitate information exchange between BIM-based building contextual data, IoT devices' time-series data, and BAS metadata using Semantic Web technology. The proposed framework is illustrated in **Figure 4.1** with several sub-sections:

Data Acquisition: The framework starts with collecting raw data from different systems in various formats. This process prepares building contextual data in an IFC-SPF file, BAS metadata in CSV file, and time-series data in the relational database. Building contextual data that stored in BIM contains information about building topology, building component spatial relationships, geometry, material properties, subsystems specification, and IoT device contextual information. The framework does not require full building contextual information, only information like space, room, and geometry of some components is needed for integration. BAS metadata identifies, describes, locates, and contextualizes sensors and actuators in the buildings. The metadata about data points in the BAS system usually contains tag string descriptor, units, data type, communication protocol names, associated space, unique identifiers, subsystems, control logics, etc. This information stores in BAS software like Johnson Control MetaSys. As the naming convention for BAS data points varies between vendors, systems, and buildings, the effort to understand the BAS metadata is significant. Times-series data generated from IoT devices describes continuous records from deployed sensors and meters in the buildings.

The time-series data can be obtained from relational databases, existing BAS software like Niagara, and facility management website.

Using Ontology: The proposed framework aims to integrate building contextual information in BIM, BAS metadata, and IoT time-series data. Based on these data silos, their native data schemas are analyzed and compared to standardized ontologies like ifcOWL, BOT, and BRICK. This step aims to select reusable ontologies to semantically describe the data silos and assigns standardized vocabularies and relationships from the reusable ontologies to the raw data silos. The ontology selection process is based on reusable ontologies, native terms, and data schemas from raw data silos.

Target Data Serialization: Building contextual data and BAS metadata serialization is based on the selected ontologies (BOT and BRICK), their target data serialization is in RDF serialization like Turtle and RDF/XML using conversion tools and scripts in Python. Time-series data from IoT devices retains its original format and stores in the relational database.

Validation and Evaluation: This step contains both data validation and framework validation. For data validation, the RDF validator guarantees the data validity of the RDF serialization for the building contextual data and BAS metadata. The framework validation process is based on the use case queries. Graph database tools like Neo4j and Stardog stores the validated RDF serialization and enables SPARQL queries. SQL queries for time-series data in the relational database are connected to SPARQL quires using IoT devices ID.

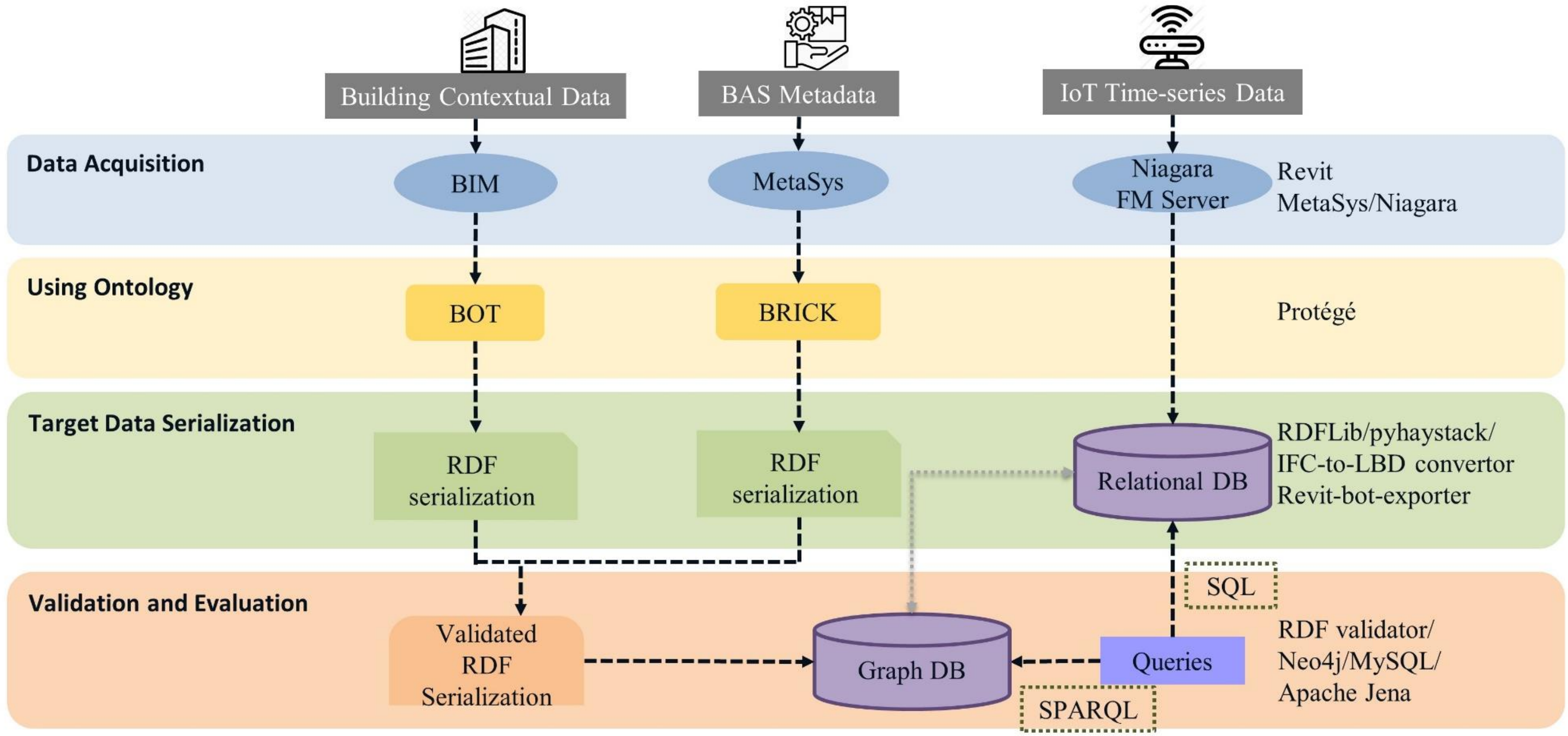


Figure 4.1 Proposed Framework for BIM, BAS and IoT Integration

CHAPTER 5 DATA ACQUISITION AND SERIALIZATION

In this chapter, the methodology for the framework implementation is detailly explained. The chapter starts with a description of collecting data silos resides in various AEC systems. Then, the chapter explains the process of ontologies selection and how to manipulate data silos using semantic web technologies.

5.1 Data Acquisition

The proposed framework starts with data acquisition as the first layer. Data silos from various AEC systems require extraction and export. The overall data acquisition process is shown in **Figure 5.1**. Building contextual data resides in BIM models is exported into the IFC SPF file. The BIM model needs modification and a clean process to delete unnecessary entities such as trees, people, and decorations. Entities related to the basic building topology and geometry are the priority. Some export settings are necessary to decide an IFC version, exported categories, and entities mapping. For example, in Revit, the in-session export set up is processed based on the use case's objective. The next step is to collect raw BAS metadata which stores in different BAS software, project information database like drawings and models, and facility managers' knowledge. The data points in BAS software are usually named according to vendors, for example, "B60A.NAE227.Bacnet1.Room110.FC-1-1.ZoneTemp". The data points names need to be extracted from BAS software into a comma-separated file (*.CSV). Data points names, project information, and facility managers' knowledge help to identify parsing rules like naming convention, relationships between equipment and data points, and implied relationships between equipment. The third step is to get access to all the data points readings. Historical time-series data and real-time time-series data can stores in a relational database in vendor tools or announced through websites. Based on the different choices, time-series data can be prepared as a CSV file or an HTTP link to the database server.

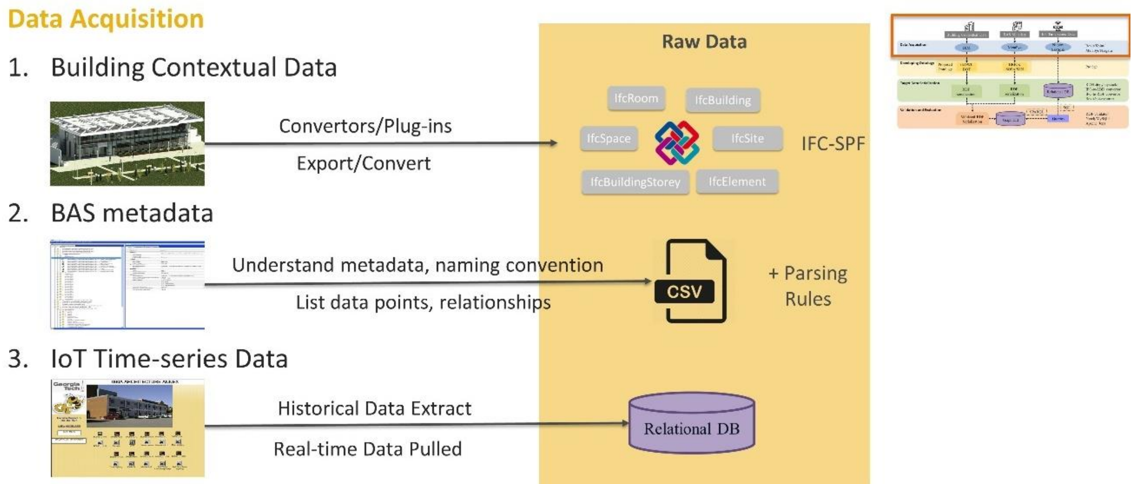


Figure 5.1 Data Acquisition Process

5.2 Ontologies Selection and Parsing Rules

The ontology selection depends on the raw data source collected in the previous section. There are several consecutive steps illustrated in **Figure 5.2** to select ontologies and generate parsing rules. The grey boxes represent key procedures connecting the overall steps to achieve the proposed framework. Data flow that includes outputs is represented by blue boxes, while the yellow boxes describe the ontology selection steps.

Based on the previous section, the selected raw data include building contextual data from the BIM model, BAS metadata in BAS software, and time-series data relational database. The data source access requires credentials login into the university's facility management system (BAS systems).

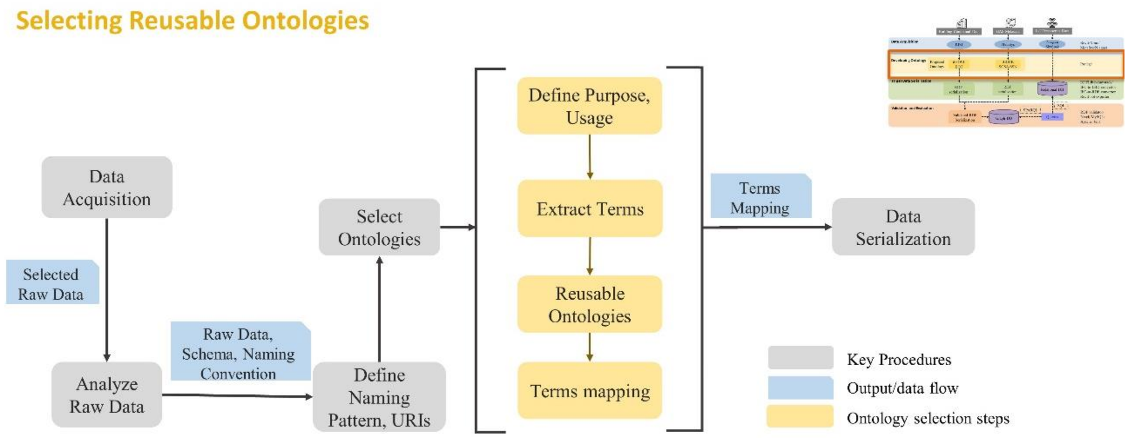


Figure 5.2 Ontology Selection Steps

After obtaining the raw data source, the next key procedure is to analyze the raw data's characteristics (structure quantity, value range, etc.) and schema (domain concepts, relationships). The building contextual data is represented in IFC, the entities and relationships that are necessary for the proposed framework are analyzed. The IoT time-series data character and schema is simple and can be directly used. The naming convention for BAS metadata requires domain knowledge to understand the concepts and relationships.

The next procedure is to define the URIs to be used to name the data source. This naming pattern should follow guidelines like Linked Data patterns [99]. In this project, the URIs (Namespace) is defined as:

- For the building contextual data: <https://dbl.gatech.edu/project/BIM_BAS_IoT_Metadata#>
- For BAS metadata: <https://dbl.gatech.edu/project/BIM_BAS_IoT_Contextual#>

They consecutive ontology selection steps are the key points for this section.

- First of all, the objective is integration for BIM, BAS, and IoT data.

- The second step is to extract terms from raw data and the raw data schemas including the domain concepts, relationships, and identification of synonyms. These terms are mapped to the selected ontologies' concept and relationships to form a terms-mapping documented as a draft in an excel spreadsheet. Later, this terms-mapping excel spreadsheet is converted into JSON files as key-value pairs.
- Reusable ontologies should be explored to cover previously extracted terms. In this case, BRICK schema and BOT are detail examined to identify terms overlapping. The goal is to identify as much overlap as possible between class/properties in selected ontologies and previously extracted terms from raw data schemas. Detail explanation of BRICK schema and BOT is in **Section 3.4**.
- The next step is to find the mapping between the raw data terms and ontologies class/properties and relationships. The BRICK and BOT ontologies in *ttl are available online for downloading [100][101]. These *ttl ontologies files are imported into a tool name Protégé for visualizing class/properties and relationships. Based on the analysis of raw data terms and ontologies concepts, a mapping table can be produced to show the alignment. The terms-mapping spreadsheet forms the parsing rules in JSON files for data serialization in the next section.

The ontologies selection procedure creates parsing rules in JSON files as outputs which are used in the next section for data serialization.

5.3 Data Serialization

The data serialization procedure aims to transform the raw data sources into RDF serializations. Based on the raw data characteristics and schema, different RDF serialization should be analyzed to choose an optimal solution for simplicity, human-readability, and processing speed. Depending on the raw data format, the transformation tools convert various data sources into target RDF serialization. A mapping between the data and the selected ontologies forms the parsing rules. There are many tools available for the transformation process depending on the raw data formats. For example, from data in a relational database to RDF uses morph-RDB, from XML file to RDF uses XML2RDF and OpenRefine, from spreadsheets to RDF uses Excel2RDF and OpenRefine. Scripting and libraries also enable instantiate of RDF serialization.

5.3.1 Building Contextual Data Serialization

Building contextual data is represented using IFC schema, the native data format is IFC SPFF. shows an example of building contextual data represented in IFC SPFF. It is possible to serialize IFC STPFF into RDF according to the BOT ontology using the IFCtoLBD converter [102] at https://github.com/jyrkioraskari/IFCtoLBD. A more convenient way is to export *.ttl file from Revit models using a plug-in named revit-bot-exporter at https://github.com/MadsHolten/revit-bot-exporter. Other tools are also available [103]. Another choice is to convert IFC SPFF into ifcOWL RDF serialization using automatic tools. The official recommended tool for such a conversion process is a Java component named IFCtoRDF at https://github.com/pipauwel/IFCtoRDF.

In this project, the IFCtoLDB converter is used to serialize building contextual data according to BOT. IFCtoLBD converts IFC SPFF into RDF triples according to linked building data ontology BOT. It is a reusable Java component. There are options in this Jar component that allows choosing levels of detail for serialization based on PRODUCT and

PROPS. The output is a *ttl file as shown in **Figure 5.3,** which can be represented as a graph. Instance data is represented in light blue boxes, while other colors are BOT concepts. The detailed example of building contextual data serialized in *ttl is shown in **APPENDIX B. Contextual Data Serialized in *ttl**.

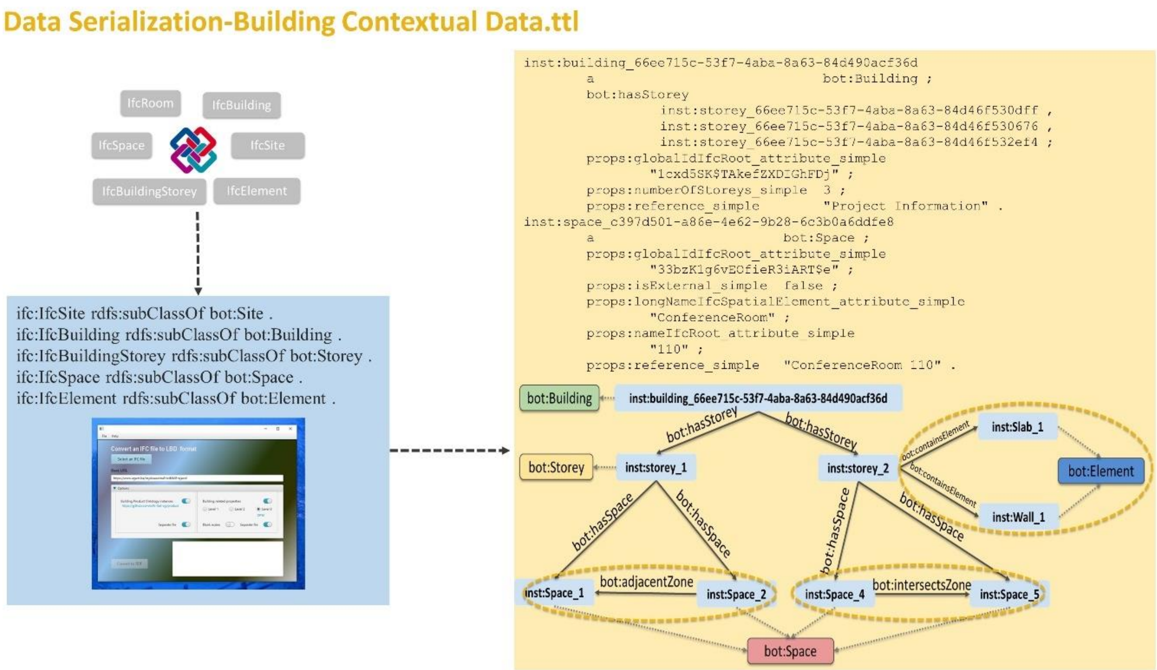


Figure 5.3 Data Serialization-Building Contextual Data

5.3.2 BAS Metadata Serialization

The overall process for BAS metadata serialization is shown in **Figure 5.4**. BAS metadata serialization requires both raw BAS metadata in CSV file and parsing rules JSON files formed from terms-mapping during the ontology selection step in **Section 5.2**. For raw BAS metadata, it is represented as a list of understandable building data points from the BAS software. The list could be a CSV file or an Excel spreadsheet. Parsing rules are created from the terms-mapping and represented in JSON files as shown in **APPENDIX**

D. BAS Metadata Parsing Rules. Parsing these data points in the list using Python script and RDFLib (Python package for RDF work) as shown in **APPENDIX E. Python Script for BAS Metadata Parsing** can semi-automate the data serialization process for BAS metadata. This process may iterate to fully instantiate a BAS metadata RDF model [104]. An example of the serialized BAS metadata in *.ttl is shown in **APPENDIX F. An Example of BAS Metadata serialized in *ttl**.

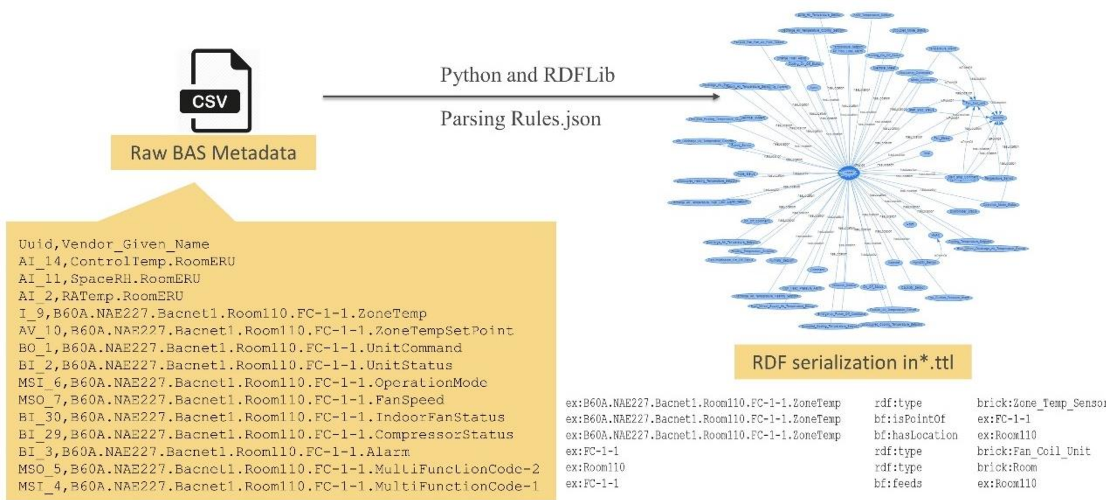


Figure 5.4 Data Serialization-BAS Metadata

5.3.3 Time-series data

For IoT devices generated time-series data, the historical record is stored in relational databases while the real-time reading can be pulled with Python script to store in a relational database. The time-series data in the relational database is announced through server or HTTP links. Hence, the data conversion process is not required for time-series data [5].

5.4 Serialized Data Validation and Framework Evaluation Methods

Before validating the proposed framework, it is crucial to validate the RDF serializations based on the chosen ontologies. The validation process for RDF serialization can be done using the RDF validator [105][106].

The success queries between the RDF serialization including both building contextual data and BAS metadata, and the IoT time-series data prove the evaluation process of the proposed framework. To achieve this goal, the RDF serializations are sent graph database using Neo4j. The unique identifiers are the links between virtual sensors in IFC entities, BAS data points, and ID in relational databases. RDF serializations in graph databases allow SPARQL queries. To query time-series data in relational databases, there are several options:

- Relational Database to RDF mapper: With Apache Jena API and libraries like http://d2rq.org/ allows SPARQL queries on non-RDF databases. This library generates a virtual RDF graph from the relational database. A mapping table between the proposal ontology and relational database schema is needed by writing with the mapper's mapping language. Then, an engine evaluates SPARQL queries based on the mapping table to generate SQL queries.
- Generate SQL queries from SPARQL [5][64].
- Create unique links to the relational database for each data point, and set these links as a property for the nodes in the RDF graph.

Detail discussion on the framework proof-of-concept evaluation and validation is in **CHAPTER 7**.

5.5 Publication

The publication of data depends on the data license and is optional. The legal compliance about raw data licensing and access control are analyzed. If the license allows, the serialized data will be published online for potential exploration of other data connectivity. The RDF data can be stored in a public repository with unique HTTP URIs, which enables dataset download. The repository contains both machine-readable metadata descriptions and human-readable documentation of RDF data.

CHAPTER 6 BIM ASSISTED BAS INFORMATION EXCHANGE USING BACNET AND IFC

This chapter presents research which aims to set a fundamental step to facilitate information exchange for BIM assisted BAS design and operation using one of the BAS open communication protocol BACnet and open BIM standard IFC. This research leverages IDM and MVD methodologies to define an IFC subset schema (a BACnet MVD) so that BAS information conforming to the BACnet protocol can be represented in the IFC data model for information exchange throughout various project stages with BIM tools. Revit and a web browser were used to demonstrate the implementation of the BACnet MVD for the BAS information exchange. This chapter is disseminated in [107].

The BIM assisted BAS design and operation information exchange using the developed MVD is utilized as a validation use case in the next chapter.

6.1 Introduction

The Architecture, Engineering and Construction industry cannot escape from the pervasive digital revolution. Smart buildings are the trend of the next generation's commercial buildings, which link different building systems together with the BAS. Smart buildings empowered by smart BAS allow data collection, analysis, and control to assist with facility functions and services [4]. BAS is a data acquisition and control system that incorporates various functionalities provided by the control system of a building. BAS is also known as Energy Management Control Systems (EMCS), Building Management Systems (BMS), Building Energy Management Systems (BEMS), Facility Management

Systems (FMS), and Building Automation and Control System (BACS) [3]. Common functionalities of BAS are temperature and air quality monitoring, lighting system control, HVAC system control, electricity control, access control, security control, fire control, and sending signals when faults occur [108]. BAS relies on sensors to collect the condition or status of control parameters and actuators to conduct physical actions. Different subsystems in BAS and devices manufactured by various vendors need to communicate with each other. Data communication protocols play key roles in information exchange in the BAS domain. Recent protocols, such as Building Automation and Control Networks (BACnet), LonWorks, EIB/KNX, and MODBUS dominate BAS communication networks[3]. Building information modeling (BIM) assists data exchange and information flow among architects, engineers, clients, and contractors throughout various project stages. The Industry Foundation Class (IFC) standard acts as a medium for data exchange across domains, stages, and parties for BIM applications [109]. Integration of BAS and BIM has been explored in previous research for energy management [8], building design optimization and operation [50], and building fault detection and diagnostics [81].

However, it is rarely seen to design BAS (i.e. construct 3D BAS models in BIM tools) or exchange BAS information (i.e. exchange BAS information with IFC) in different project stages using BIM tools [110]. The current design of the BAS system is either using 2D drawings based on AutoCAD or customized tools [111][112]. Unlike the other building systems, BAS seldom participates in the design-build BIM cycle but blends into facility management (FM) in the later stage. Without BIM, issues like information loss, inefficient collaboration, and error-prone construction happen. BAS is part of the building system, if BAS does not participate in 3D design coordination, error-prone design may happen due to various complex building systems. Normally, BAS is the last system to be built in the construction phase, it may suffer from error and corrections made during the setup of other systems. The information sharing between BAS designers, all sub-contractors, and building's general contractors without BIM may be inefficient [113][114].

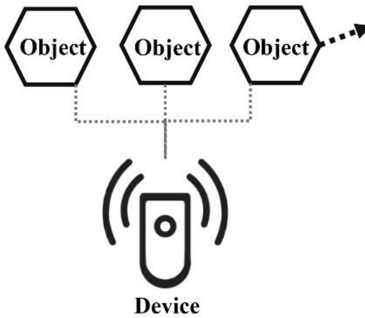
The great potential can be exploited if BAS information can be declared into BIM from the design stage throughout the operation and maintenance stage. BAS is designed based on communication protocol like BACnet, while BIM information can be represented using open standards like IFC. With the object-oriented modeling characteristic of the BAS protocols and open BIM standard, it is possible to design and modify BAS using BIM tools without specifying device vendors during the design phase [114]. In this way, the BAS information represented in open BIM standard (IFC) can be shared among different stakeholders for construction, software, and various project stages. BAS information does not require the whole IFC schema for data representation, an IFC subschema that corresponds to BAS communication protocol is sufficient for BAS information exchange between various parties, software, and project stages.

To tackle the above-mentioned issue, this research aims to set a fundamental step to facilitate information exchange for BIM assisted BAS design and operation using one of the BAS open communication protocol named BACnet and BIM open standard, known as IFC. This research leveraged the Information Delivery Manual (IDM) and the Model View Definition (MVD) methodologies to define an IFC subset schema (a BACnet MVD) so that BAS information conforming to the BACnet protocol can be represented in IFC data model for information exchange throughout various project stages. In doing so, a BAS system can be modeled based on BIM tools without specifying actual devices. The BAS information, which is included in the models, can be exchanged among different stakeholders or BIM tools using the proposed BACnet MVD. The paper is structured as follows: **Section 6.2** provides background information about BACnet, IFC, IDM/MVD. **Section 6.3** explains the methodology and detailed process of creating the BACnet MVD and demonstrates the MVD implementation process. **Section 6.4** discusses the results and limitations. **Section 6.5** concludes the primary outcomes and next steps.

6.2 Background

6.2.1 BACnet Overview

The American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) has developed the BACnet protocol to address the communication needs of BAS for different applications like heating, lighting control, and fire detection systems. BACnet aims to solve interoperability issues among different devices vendors by modeling exchanged information with object-oriented representations [3]. The function of BAS can be modeled as a collection of BACnet Objects. Currently, there are 60 Object Types defined in ANSI/ASHRAE Standard 135-2016 [115]. The instances of Object Type are Objects. As shown in **Figure 6.1** a device can be represented as a group of BACnet Objects. BACnet Objects hold information, which relates to a device (i.e. sensor, actuator) as properties sets. Each property has an identifier, a data type, and a conformance code indicating whether this property is required or optional [116].



Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Present_Value	REAL	R
…	…	…
Description	CharacterString	O
Device_Type	CharacterString	O

Figure 6.1 BACnet Device, Object, and Properties

With the characteristic of the BACnet protocol, it is possible to declare BAS information into BIM throughout the project lifecycle [114]. Starting from the design phase, BAS can be modeled using BIM tools without specifying device vendors (i.e. BAS represented by a collection of BACnet Objects virtually). In this way, BAS design can be conveniently modified if any function or communication design varies. During the construction phase when physical devices are chosen based on vender specification, the BAS model which contains all device-specific information and devices interconnection information, facilitates specifying physical devices. An updated BAS model can be handed to the owners and facility managers during the handover phase for operation and maintenance (O&M). One potential application benefitting O&M is accessing BAS information through the extended BACnet Web Service (BACnet/WS) and integrating it with BIM online. The BACnet/WS is capable of using technologies like REST, JSON, and OAuth2 within BACnet and integrating with the BIM models [114]. The information resides in BIM can be accessed through BACnet/WS for energy saving, maintenance management system [81], etc [117]. Another potential application benefiting O&M would be to exchange information that resides in the BAS model with FM tools like Metasys and

Niagara which directly connect to sensors and controllers as the control system using the MVD created in this paper. The MVD created in this paper is useful for exchange information at the BACnet object level which is necessary for FM tools that connect to BAS control systems. Although some FM tools like Archibus and EcoDomus have already integrated the BAS model with BIM data, these tools do not connect to sensor control systems using BACnet protocol and are specialized in higher-end management. Unlike the FM tools that connect to the BAS control system, these tools fail to encompass all FM requirements [118].

6.2.2 IFC/MVD/IDM Overview

The most commonly used data exchange format for open BIM is IFC that has been accepted as ISO 16739 standard. The IFC enables data exchange between different software applications across the entire building lifecycle [96]. It is the most suitable median to hold BACnet based BAS information both for technical and practice perspectives.

BIM models can be enormous if the information is fully integrated. A fully populated model is unnecessary for all stakeholders or a certain software at a project stage. To solve this issue, buildingSMART International created IDM and MVD approaches to define subsets of IFC schema for certain Exchange Requirements (ER) [119]. In 2012, buildingSMART released an integrated IDM/MVD approach named “An Integrated Process for Delivering IFC Based Data Exchange” which amalgamate the previous IDM approach and MVD approach into one [120]. An IDM firstly captures business processes and ERs at the user level [121]. Then, the MVD defines a subset of IFC schema based on ERs identified by the IDM. In this way, models can be filtered and size-reduced according to an MVD to satisfy specific business processes. The subset of IFC schema (the MVD) created in this paper can potentially encapsulate information to represent BAS and facilitate information sharing among different tools.

BuildingSMART has released official MVDs among which COBie (Construction Operation Building Information Exchange) is used to exchange specifications for life-cycle capture and delivery of information needed by facility managers [122]. Although COBie can capture life-cycle information needed by facility managers, the data entities in COBie MVD documentation particularly focused on IfcConstructionMgmtDomain (e.g. IfcConstructionProductResource, IfcContructionResrouce) [123]. COBie does not emphasize on exchange BAS information that follows the BACnet protocol. Some practitioners suggested that COBie contains universal facility management parameters and fails to selectively filter the most relevant data for bespoke operation and maintenance requirements [51]. There is a gap in understanding necessary semantic data to exchange regarding BAS design, construction, and operation following an international standard like BACnet.

Apart from official released MVDs on the buildingSMART website, some effort has been made to address the interoperability issue by creating several MVDs. For instance, Arayici et al. [124] utilized IDM/MVD methodology to create interoperability specifications for performance-based design. This research leveraged "An Integrated Process for Delivering IFC Based Data Exchange" method developed in 2012. However, an advanced MVD documentation tool named IFC Documentation Generator (IfcDoc) [125] enables a more convenient way to document an MVD compare to the conventional method. Some official MVD (e.g. Design Transfer View) do not necessarily require IDM to identify ERs and business processes. With the MVD documentation tool, Pinheiro et al. [126] and Andriamamonjy et al. [127] utilized the MVD methodology to facilitate information exchange between BIM tools and building energy performance simulation tools. Other efforts related to extending and improving IDM/MVD methodology can be seen in [128–131].

6.3 Methodology

This paper leveraged IDM/MVD methodologies to define an IFC subset schema (a BACnet MVD), so that BAS information complying with the BACnet protocol can be represented in the IFC data model for information exchange between BIM tools and FM tools throughout various project stages. Apart from official IDM/MVD process defined by buildingSMART [120], this study followed similar IDM/MVD methodology in previous research [124,126,127,132,133]. The methodologies taken in this paper can be divided into three parts as shown in **Figure 6.2**.

Firstly, the IDM method defined the information sharing process and a set of information to be exchanged at the user level for BAS design and operation. The IDM contained process model, ER, and Functional Parts (FPs). A process model using Business Process Modelling Notation (BPMN) was created. The process model identified the purpose and a set of data for exchange. Based on the process model, ANSI/ASHRAE Standard 135-2016 (the BACnet protocol) [115] was reviewed to initiate the identification of ERs needed for BIM assisted BAS design and operation information exchange. As the BACnet protocol was defined by BAS domain experts, information that needs to be exchanged (ERs) was directly extracted from the BACnet protocol. These ERs defined a group of information units to exchange based on the process model and ANSI/ASHRAE Standard 135-2016. The information units were further breakdown into FPs which described the information in terms of the required capabilities of IFC standard.

The second part involved developing an MVD using the IfcDoc tool. The BACnet MVD focused on the latest release of the IFC schema, IFC4 Addendum 2 (IFC4 Add2) [134]. The MVD enabled IDM outputs to translate into IFC entities, attributes, and properties to facilitate interoperability at the software level. The functions of IDM outputs including process model, ERs, FPs were represented by MVD concepts. A mapping spreadsheet was created to clarify the relationships between each information unit, FP, and MVD concept. The last part demonstrated a prototype test of the BACnet MVD using Autodesk Revit and Web browser as importing and exporting tools for BAS information exchange. Several Revit Families were constructed with custom-defined properties. Revit acted as an exporting tool following the BACnet MVD. BACnet Object Types and property identifiers expressed in IfcXML were imported into a Web browser to demonstrate the possibility to exchange BAS information using the IFC data model. Detail steps for each part will be described in this section.

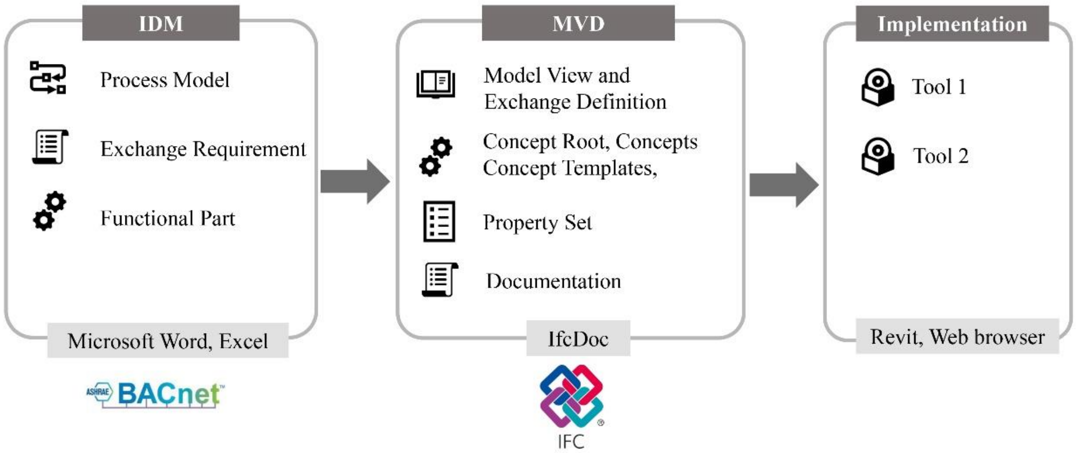


Figure 6.2 Overall Process of Methodology

6.3.1 IDM

IDM aims to collect domain knowledge and information needed regarding workflows from experts. IDM specifies process model, ERs, and FPs at the user level [121]. BuildingSMART provides official templates for IDM documentation including process map template, ER template, and FP template [135]. With modern documentation tools like IfcDoc, creating an MVD does not necessarily require an IDM. However, the conventional templates are useful to streamline the IDM/MVD process and demonstrate various terminologies. Considering the ease of use, templates were improved to cater to the modern tool in this study. Some redundant sections in the templates were removed and combined.

6.3.1.1 Process Model

The process model is the initial step to identify the purpose and a specified set of data for exchange. The formalization of the process model describes the activities, related information, logical sequence of activities, and roles involved for a particular goal. The Object Management Group (OMG) developed the BPMN for process modeling [136]. It provides a standard diagramming language for mapping flow-oriented representations of the business process, facilitating the identification of ERs [121,124].

The process model was created to illustrate the process of BAS design and operation information exchange across project stages. As shown in **Figure 6.3**, the process map represents the process model in BPMN created by Visio. The codes and the phases of this process are based on Omni Class Construction Classification System [137]. During the design phase (31-40 00 00), the process starts with design roles in the project team including architects, structural engineers, MEP engineers designing the preliminary BIM models using BIM tools like Autodesk Revit and Tekla. These preliminary BIM models, containing building contextual data, are submitted to BAS designers for BAS preliminary design. As there are no widely adopted BIM-based BAS design tools available in the market, the BAS design can be done using traditional BIM tools less efficiently. The process loops around preliminary project design and preliminary BAS design for several iterations in the coordination phase (31-50 00 00) before final models can be handed over to facility use roles. The final BAS model containing BAS metadata can be imported into FM tools for BAS in the operation phase (31-80 00 00). The data exchange swim lanes contain information flow between different parties.

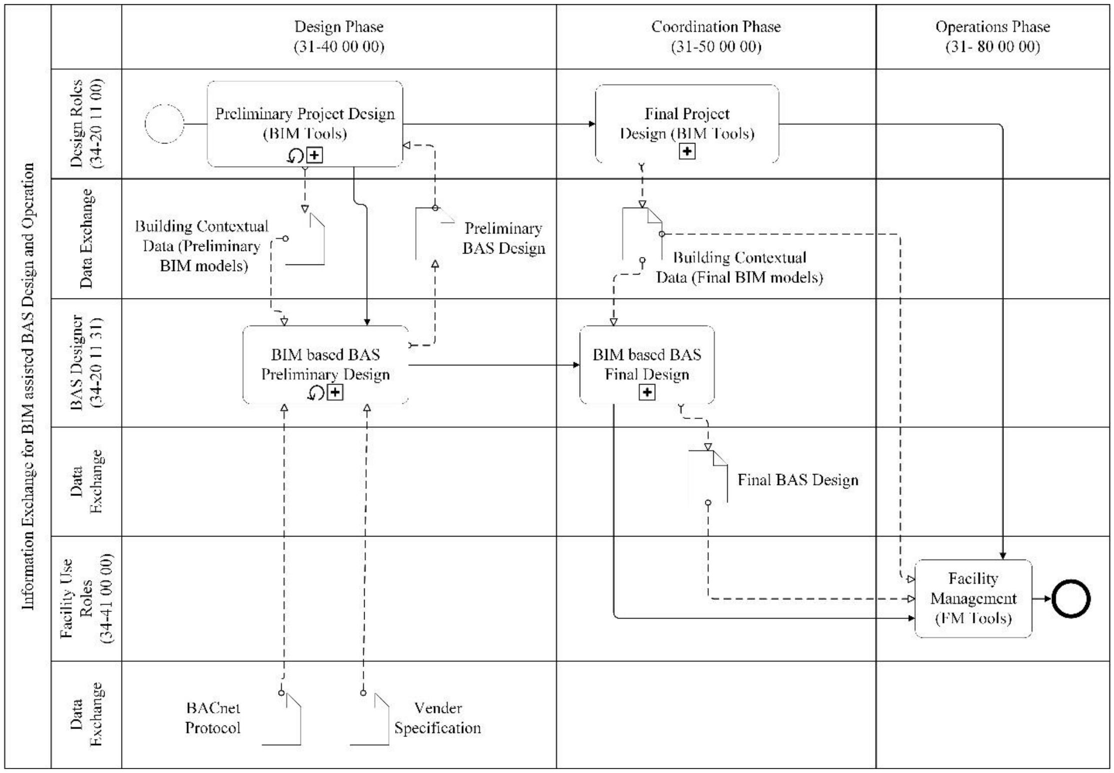


Figure 6.3 Process Map for BIM-based BAS Design and Operation Information Exchange

6.3.1.2 Information Exchange Requirements

Information or data flow between two or more parties is documented by ERs. An ER connects business processes with relevant information defined within a particular information model [120]. As shown in the process map (**Figure 6.3**), ERs which link two or more tasks, are the data items in data exchange swim lanes. The identification of ERs started with an in-depth analysis of ANSI/ASHRAE Standard 135-2016 [115] by understanding the BACnet protocol architecture, modeling control device as a collection of objects, object types, property identifiers, and property datatypes. The relationships and similarities between BACnet object types/property identifiers/property datatypes and IFC entities/attributes/datatypes were discovered during this analyzing process. As the objective is to declare BACnet data, which represents BAS metadata from the design phase to the operation phase, BACnet object types and properties identifiers are the keys to ERs.

The implementation starts with an official ER template that consists of a header & overview section and information requirements section. **Table 6-1** shows the header & overview section of the ER for a BAS model using the BACnet protocol. The codes and the phases of this process are based on Omni Class Construction Classification System[137]. It specifies general information, project stages, scope, and descriptions regarding the ER. The key to identify ERs is the list of information units in the information requirement section. The information requirement section describes a set of information units to satisfy user requirements. These information units were identified through BACnet object types and property identifiers in ANSI/ASHRAE Standard 135-2016 [115]. Apart from the BACnet information, 3D visualization, location, and relationships between BACnet objects for the BAS system would also benefit the design and operation process. One of the potential usages of the BACnet MVD is to assist design BAS systems using BIM authorizing tools, the BACnet objects can be modeled as virtual instances in BIM tools. The geometric representation, location, space relationships, and connectivity between these virtual instances are also important information for BAS design. Hence information units include Object Geometric Representation, Object Placement, Object Contained in Space, Object Connected From, and Decomposes were added to define ERs. **Figure 6.4** shows an example of mapping between information units and ERs. For example, the Properties of BACnet Analog Input Object Type, which were extracted from ANSI/ASHRAE Standard 135-2016, are illustrated in the left table in **Figure 6.4.** This object type and its property identifiers were listed as information units (first column) in the

ER_BACnet Analog Input Object Type Table (see the right table in **Figure 6.4**). The Header section, overview section, and the full list of information units can be found in ER_BACnet MVD.xlsx. as linked Mendeley data [107], sample data can be viewed in **APPENDIX G. SAMPLE ER_BACNET MVD.XLSX**.

Table 6-1 Exchange Requirements Header and Overview Section

	Exchange Requirement		
Header Section			
Name	ER_Exchange Final BAS Model		
Identifier	BACnet_ER_001		
Change Log			
<2018-03-20>	Created BACnet objects and Properties		stang93@gatech.edu
Project Stage	31-10 00 00	Inception Phase	
	31-20 00 00	Conceptualization Phase	
	31-30 00 00	Criteria Definition Phase	
	31-40 00 00	Design Phase	√
	31-50 00 00	Coordination Phase	√
	31-60 00 00	Implementation Phase	
	31-70 00 00	Handover Phase	
	31-80 00 00	Operations Phase	√
	31-90 00 00	Closure Phase	

Table 6-1 Continued

Overview **Scope** The scope of this ER is the exchange of BAS design information between BIM tools and FM tools in different project stages. The BAS components should be designed using BACnet protocol including object types and property identifiers. The ER also includes the need for shape, location, and connectivity of components. **General Description** This ER allows BAS component information represented in the IFC standard to be shared between BIM tools and FM tools throughout project phases. The BAS component information can also be shared throughout several design interactions from conceptual design to production information. **Information Description** Information provided through this ER includes: • BAS components objects, attributes • Shape representation of components • Location and orientation of occurrences of components • connectivity and composition of components

6.3.1.3 Functional parts

Information units can be broken down into FPs. Each FP describes the information in terms of the required capabilities of the IFC standard to provide technical support for information units. An FP is a reusable information model in its own right as well as being a subset of information model on which it is based on IFC [120]. Each information unit is mapped to an FP, which can be expressed as: i) IFC entity; ii) attribute of IFC entity with specified data type; iii) property in a property set with specified data type; iv) referring to another FP. Additional information regarding importing/exporting requirements for each

FP falls into one of these categories, namely mandatory, recommended, optional, and not recommended.

To start constructing FPs, the conventional template contains redundant information, which is unnecessary for the modern documentation tool, was modified into a spreadsheet. FPs can be expressed as a mapping table between information units and IFC entities, attributes, or properties in property sets. This step required a detailed exploration of IFC schema and BACnet protocol to discover target IFC entities, attributes, properties, and their data types to represent functions of information units. **Figure 6.4** shows an example of BACnet analog input object type, the FPs' data types as in the ER_BACnet Analog Input Object Type Table (right table, 2^{nd} column) conform to BACnet objects' property datatypes (left table, 2^{nd} column). The conformance code of BACnet object properties specifies the import/export requirement for ERs. The convention defined in IDM for expression of FPs are:

- Object.Attribute-> Datatype
- PropertySet.Property->Datatype
- PropertySet.Property->Property Type::Datatype

Properties of the Analog Input Object Type		
Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Present_Value	REAL	R
Description	CharacterString	O
Status Flags	BACnetStatusFlags	R
Event State	BACnetEventState	R
Out of Service	BOOLEAN	R
Units	BACnetEngineeringUnits	R
…	…	…
Property List	BACnetPropertyIdentifier	R

ER_BACnet Analog Input Object Type					Mapped Concept
Information Unit	Entity/Pset/Functional Part	Man	Rec	Opt	
Object Geometric Representation	IfcProduct.Representation ->IfcProductDefinitionShape		√		Box Geometry; FootPrint Annotation Geometry
Object Placement	IfcProduct.ObjectPlacement ->IfcObjectPlacement		√		Product local placement
Object Contained in Space	IfcElement.ContainedInStructure ->IfcRelContainedInSpatialStructure		√		Spatial Containment
Object Connected From	IfcElement.ConnectedFrom ->IfcRelConnectsElements		√		Path Connectivity
Decomposes	IfcObjectDefinition.Decomposes ->IfcRelAggregates		√		Object Definition Composition
Object Identifier	IfcControllort.GlobalId- >IfcGloballyUniqueId	√			Software Identity
Object Name	IfcControllor.Name->IfcLable	√			Object User Identity
Object Type	IfcController.PredefinedType ->IfcControllerTypeEnum	√			BACnet Controller Predefined Type; Object Predefined Type
Present Value	Pset_ControllerTypeMultiPosition.Value ->IfcInteger	√			PropertySets for Objects
Description	IfcRoot.Description->IfcText			√	Description
Status Flags	Pset_BACnetAnalogInputObject. StatusFlags ->IfcLabel:: PEnum_BACnetStatusFlag	√			PropertySets for Objects
Event State	Pset_BACnetAnalogInputObject. EventState ->IfcLabel:: PEnum_BACnetEventState	√			PropertySets for Objects
Out of Service	Pset_BACnetAnalogInputObject. OutOfService ->IfcBoolean	√	`		PropertySets for Objects
Units	Pset_BACnetAnalogInputObject.Units ->IfcLabel	√			PropertySets for Objects
Property List	IfcPropertySet.HasPropertites ->IfcProperty	√			PropertySets for Objects

Figure 6.4 Example of Information Unit, Functional Part and Concept Mapping

6.3.2 MVD

MVDs have been defined as subsets of IFC model specification to support IFC implementation. The IFC implementation should satisfy requirements coming from end-users as defined in IDM [120]. MVDs enable IDM outputs, including process map, ERs, and FPs to translate into IFC schema and to facilitate interoperability at the software level. An MVD consists of Model Views, Exchange Definition, Concept Root, Concepts, Concept templates, and Property Sets. Each of these elements will be explained in detail to generate the BACnet MVD in this section.

BuildingSMART developed an official tool IfcDoc to assist in generating MVDs. IfcDoc improves consistency and computer-interpretability of the MVDs' definition [126]. IfcDoc assists in generating diagrams, defining schemas, and specifying the scope and contents of custom-made specifications [125]. The basic steps to create the BACnet MVD started with the IfcDoc tool and ER_BACnet MVD.xlsx generated from the IDM process in **Section 6.3.2**.

6.3.2.1 Model View and Exchange Definition

Model View defines the scope of MVD by specifying Exchange Definition, entity usage, concepts usage, and importing/exporting requirements. Exchange Definition inherits information in ERs for a certain exchange scenario. Model View groups a set of IFC entities (entity usage) and Concepts (concept usage) to satisfy exchange scenarios. Model View allows development based on other existing Model Views. There are several existing official MVDs defined by buildingSMART, for example, IFC4 Reference View, IFC4 Design Transfer View, and IFC2x3 Coordination View [119]. These views are provided as default model views in the IfcDoc baseline file, which will be utilized in the following step.

6.3.2.2 Concept Root, Concept, Concept usage, Concept Template

Concepts are the technical solutions to exchange a commonly useful package of information identified in ERs. Concepts can be applied to IFC entities as Concept usage including attribute usage, property usage, quantity usage, and mapping usage. Concept Roots collect available Concepts in a hierarchical tree structure. Concept Roots divide Concepts by their context and objectives such as project content, object definition, and object attributes. Concepts can be represented using a Concept template as shown in **Figure 6.6.** Concept template specifies entity reference, attribute reference, and relationship constraints for a Concept. An instance diagram displays a graph of entities, relationships and constraints to clarify a Concept template.

The operation on IfcDoc started with importing the IFC4 Add 2 baseline file, which contains the full IFC schema, a reusable set of default Concepts, and default model views [138]. A new Model View name BACnet View together with its Exchange Definition, entity usage, concept usage, and importing/exporting requirements was created. **Figure 6.5** shows a matrix of entities usage and concepts usage in the BACnet View. Various colored boxes represent importing/exporting requirements for the concepts applied to each entity. The importing/exporting requirements followed the BACnet conformance code. For example, information units with conformance code "R" (required) and "O" (optional) were set to import/export mandatory (green) and import/export optional (yellow) respectively.

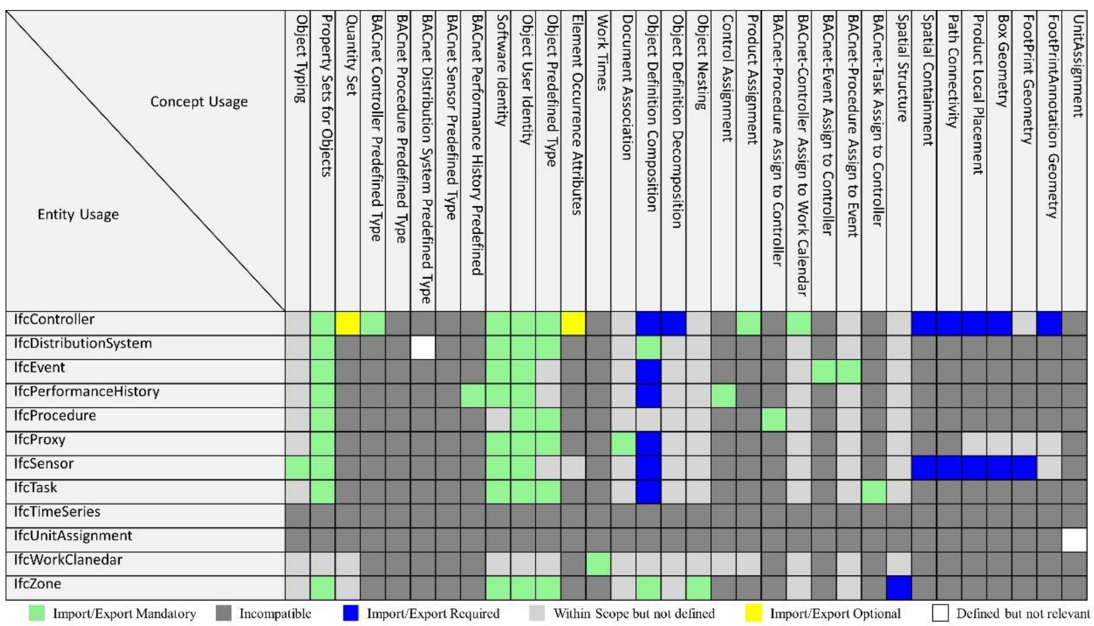


Figure 6.5 BACnet View Exchange Requirement View

The entities usage, concepts usage, and importing/exporting requirements for the BACnet View followed the rules below:

- BACnet object types → IfcEntities →entities usage

- BACnet object property identifiers → IfcEntity. Attributes →concepts usage

BACnet object types along with their property identifiers are information units with corresponding FPs. The FPs were further mapped to MVD elements as entity usage or concept usage. Concepts, when applied to IfcEntities, are concept usages. The concept usages contained both default concept templates from the baseline file and custom-made concept templates. For example, as shown in **Figure 6.4**, the BACnet analog input object type and its property identifiers were mapped to MVD entities usage and concept usage respectively (full list of the mapping table is in ER_BACnet MVD.xlsx). The BACnet Analog Input Object type was appointed to IfcController, which pointed out the IfcController entity usage in the BACnet View. The Object Type property identifier was mapped to IfcController.PredefinedType, which corresponded to BACnet Controller Predefined Type concept usage in the BACnet View. To follow the expression of the above rules, the BACnet Analog Input Object Type example can be described as below:

- BACnet Analog Input Object Type→IfcController→IfcController (entity usage)
- BACnet Analog Input Object Type. Object Type→IfcController.PredefinedType→ BACnet Controller Predefined Type (concept usage)

Although the baseline file contains a default set of concept templates, which can express certain relationships between some IfcEntities, limitation to fully express relationships between BACnet Object Types and their property identifiers still exist. The default concept templates such as software identity, spatial containment, and object user identity fulfill the need to represent relationships between some BACnet Object Types and their property identifiers, however, some relationships cannot be represented. For example, BACnet Calendar Object Type was mapped as IfcController, some of BACnet Calendar Object Type's property identifiers were mapped as attributes in IfcWorkCalender, so an assignment relationship between IfcController and IfcWorkCalendar is necessary. As shown in **Figure 6.6**, a custom-made concept template named *BACnet-Controller assign to Work Calendar* was added. The instance diagram in **Figure 6.6** shows the assignment relationship between IfcController and IfcWorkCalendar. A full list of custom-made concept templates is shown in **Table 6-2**.

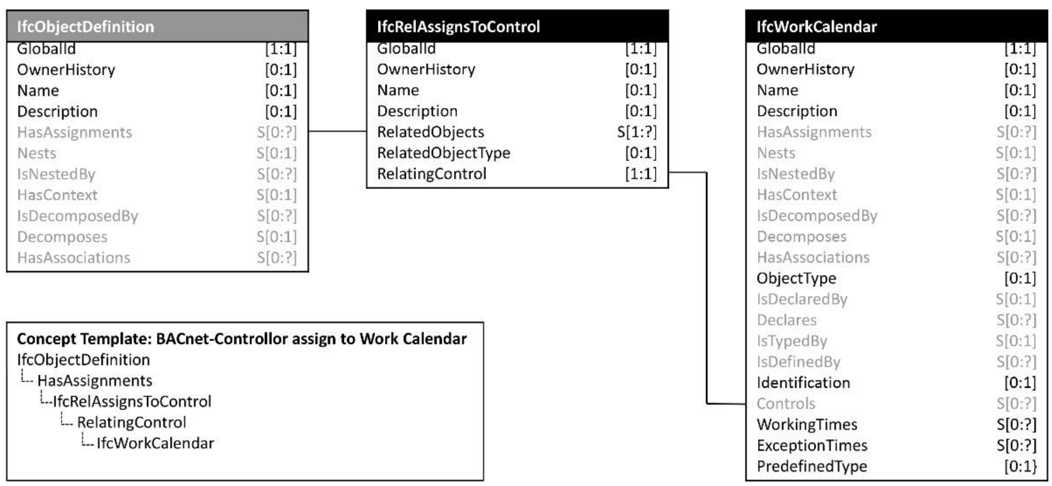


Figure 6.6 Concept template and Instance Diagram of *BACnet-Controller assign to Work Calendar*

Property sets group various properties, which contain the name, access state, property type, data type, and secondary data type. Property sets were applied to entities. The baseline file has default property sets and properties that can be assigned to entities using the property usage Concept. All BACnet Object Types and their property identifiers have been mapped as information units, however, part of these information units cannot be represented using default IFC schema and property sets. Hence, user-defined property sets and properties were created to emulate the BACnet protocol. The access state, property type, primary data type, and secondary data type observed the description for the property identifiers in the BACnet protocol.

Some of the information units in BACnet Analog Input Object Type example in **Figure 6.4** were mapped as properties in custom-made property sets named *Pset_BACnetAnalogInputObject* (**Figure 6.7**). The access state, property type, data type of custom-made properties were defined to correspond to the BACnet protocol. For example, in the BACnet protocol, the "Status Flag" BACnet property identifier, which represents Boolean flags to indicate object health, is of data type BACnetStatusFlags. The BACnetStatusFlag data type is an enumeration of IN_ALARM, FAULT, OVERRIDEN, and OUT_OF_SERVICE. In corresponding to the BACnet protocol, the custom-made Status Flag property in *Pset_BACnetAnalogInputObject* was set with property type as P_ENUMERATEDVALUE and primary data type as IfcLabel. The secondary data type was a custom-made enumeration value named *PEnum_BACnetStatusFlags* that inherited values from the BACnetStatusFlag data type. A full list of custom-made property sets and property enumeration is shown in **Table 6-3**.

Property Set Name: Pset_BACnetAnalogInputObject				
Property	**Access State**	**Property Type**	**Primary Data Type**	**Secondary Data Type**
StatusFlag	READWRITE	P_ENUMERATEDVALUE	IfcLabel	PEnum_BACnetStatusFlag
EventState	READWRITE	P_ENUMERATEDVALUE	IfcLabel	PEnum_BACnetEventState
OutOfService	READWRITE	P_SINGLEVALUE	IfcBoolean	-
Unit	READWRITE	P_SINGLEVALUE	IfcLabel	-

Property Enumeration			Property Constant	
PEnum_BACnetStatusFlag	IN_ALARM	FAULT	OVERRIDEN	OUT_OF_SERVICE
PEnum_BACnetEventState	NORMAL	OFFNORMAL	FAULT	

Figure 6.7 Example of Custom-made Property Sets *Pset_BACnetAnalogInputObject* and Property Enumeration *PEnum_BACnetStatusFlag & PEnum_BACnetEventState*

6.3.2.3 Documentation

An automatic documentation process enables additional descriptions and constraints to be encoded into MVD using HyperText Markup Language (HTML). IfcDoc generated an HTML documentation containing entity usage, concepts usage, and properties for BAS design and operation information exchange (**Figure 6.8**). This documentation acts as an indication of information that is necessary to import/export between different BIM tools and FM tools. It expedites the process to adopt IFC to targeted uses. The IfcDoc tool also automates the generation of the mvdXML file that can be used for the buildingSMART certification process or consumed by software for data transformation [139]. The sample MVD documentation (BACnet-MVD Documentation folder) in the HTML version is attached as Mendeley data [107]. The MVD can be accessed by open the index HTML document in the folder.

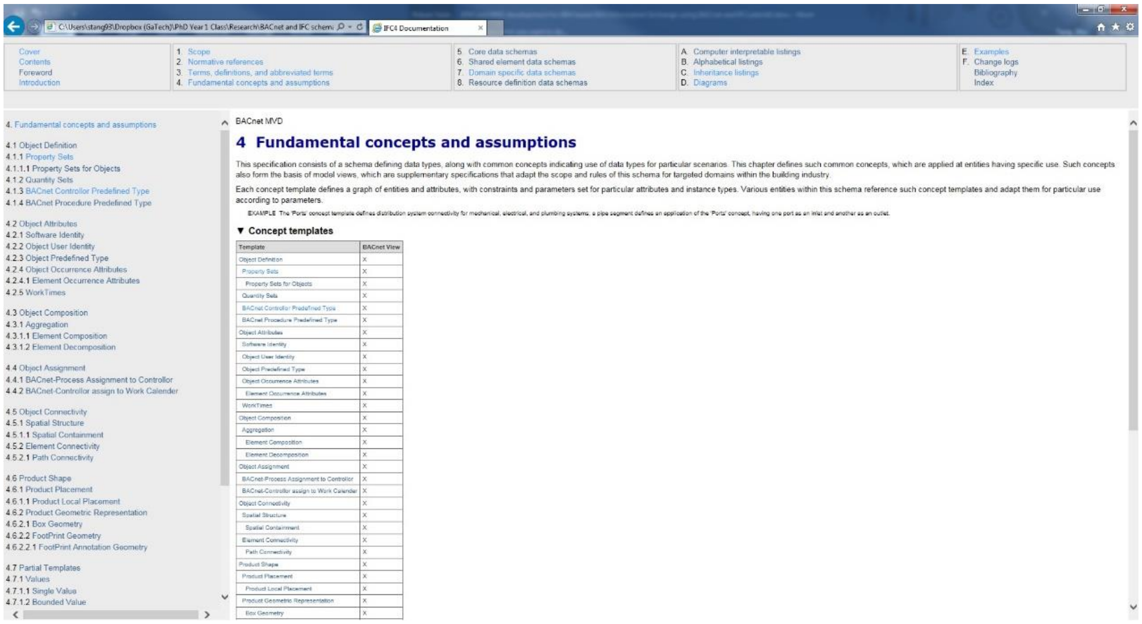


Figure 6.8 The BACnet MVD documentation in HTML

6.3.3 Implementation of the prototype test

A prototype test following the BACnet MVD was carried out with both the exporting tool and importing tool. **Figure 6.9** shows the basic steps of the BACnet MVD prototype test. These steps are described below:

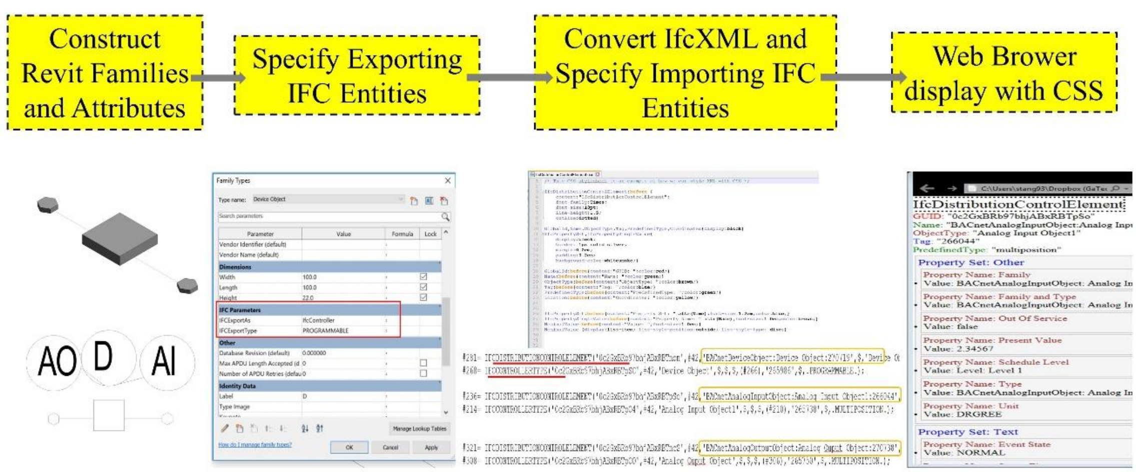


Figure 6.9 The BACnet MVD Implementation Steps

1. As the most commonly used 3D BIM modeling tool, Revit was used to create BACnet objects as families. In the prototype test scenario, the list of created Revit families representing the BACnet object types is shown in ER_BACnet MVD.xlsx. as linked Mendeley data [107], sample data can be viewed in **APPENDIX G. SAMPLE ER_BACNET MVD.XLSX**. These Revit families representing the BACnet Object types are also shown in **Figure 6.10** as *.rfa files. All these BACnet Object type families contain their BACnet property identifiers as user-defined attributes in Revit. The complete list of BACnet property identifiers is shown in ER_BACnet MVD.xlsx. as linked Mendeley data [107], sample data can be viewed in **APPENDIX G. SAMPLE ER_BACNET MVD.XLSX**. Several Revit families include BACnet Analog Input Object Type, BACnet Analog Output Object Type, and BACnet Device Object Type were modeled. The constructed BACnet objects were annotated using abbreviations like "AO", "AI" and "D" in the 2D view. One of the modeled BACnet Object types (BACnet Analog Input Object Type) is shown in **Figure 6.10** as an explanation for part of the sample testing scenario. As shown on the right-hand side of **Figure 6.10**, both type properties and instance properties were added as user-defined attributes that match the BACnet Analog Input Object Type's property identifiers.

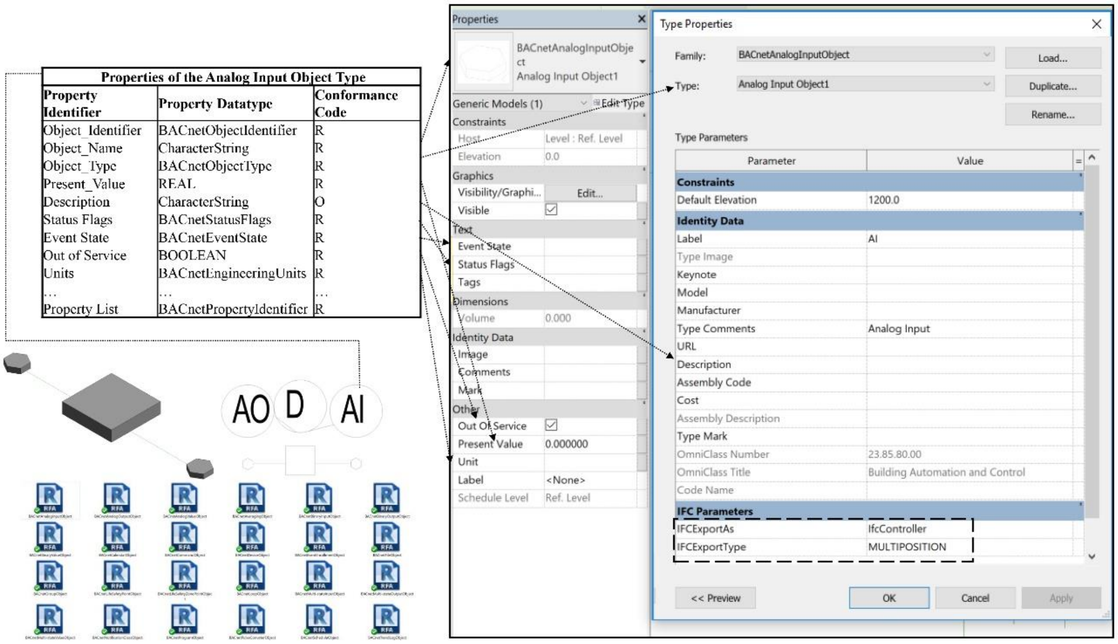


Figure 6.10 Sample Testing Scenario （BACnet Analog Input Object Type）

2. Apart from basic geometry representation, placement, and connection, BACnet properties identifiers were added as user-defined attributes for all families. For each of these BACnet object families, Revit share parameters named "IFCExportAs" and "IFCExportType" (in **Figure 6.10**) were defined to specify target IFC export entities following the BACnet MVD. E.g. in **Figure 6.12**, the BACnet Device Object has "IFCExportAS" share parameter value = *IfcControllerType* and "IFCExportType" share parameter value = *IfcDistributionControllerElement*. In this way, the target exporting IFC entities for BACnet object families and BACnet property identifiers were assigned. When loading these BACnet object families into Revit projects based on various test cases, an "In-session" exporting setup was required to follow the previously assigned share parameters. User-defined properties were also exported with the "In-session" export settings in Revit as

shown in the top dialog in **Figure 6.11**. Guidance for specifying exporting IFC entities for families using share parameters and “In-session” exporting setup can be found in the Revit user guide.

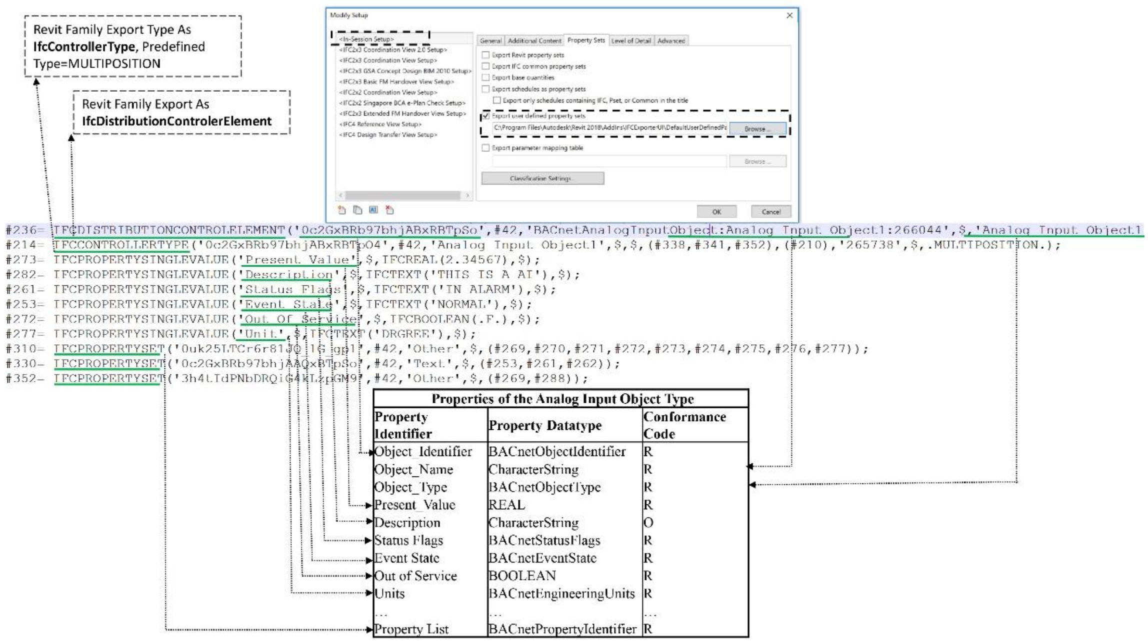


Figure 6.11 Sample exported IFC instance file with corresponding BACnet data (BACnet Analog Input Object Type)

3. The exported *.ifc file followed the exporting instruction specified by share parameters in step 2. For example, the values of “IFC Export As” and “IFCExportType” of BACnet Analog Input Object Type were “IfcCotroller” and “MULTIPOSITION” respectively. The exported IFC instance file was screened to check whether the mandatory BACnet data was included. For example, in **Figure 6.11**, the sample exported IFC instance file matches with the BACnet Analog Input Object Type’s property identifiers. IfcDoc converted *.ifc file into *.ifcXML file before specifying the importing IFC entities.
4. To demonstrate the possibility of importing BAS information in the IFC data model

into FM tools, a web browser was used as a replacement since current BIM assisted FM tools are not open source. The web browser is a demonstration of possible importing tools that can utilize the BACnet MVD. IfcDoc or other tools can be alternative options. However, IfcDoc requires special expertise while the web browser is more widely adopted. Besides, the web browser provides other potential applications such as integration with data in other domains, connecting with online resources, and integration with other data models. It is necessary to convert the *.ifc file into *ifcXML file since the web browser was chosen to be the importing tool. ifcXML show better integration capability with other software and data model than STEP Physical File (SPF) [88][109]. It also provides enhanced readability and benefits from other tools [140]. To display useful information in the importing tool, the authors specified importing IFC entities and displayed an XML file using Cascade Styling Sheet (CSS) file. The useful IFC entities and their attributes were automatically parsed and extracted into the web browser. In this way, BACnet object types and properties identifiers can be visualized through a web browser.

6.4 Results and Discussion

This study has successfully represented BACnet Object Types and property identifiers with the defined subset of IFC schema. As a result, it demonstrated the possibility to exchange BAS information conforming to the BACnet protocol with the IFC data model in various project stages. The result of this study showed that IFC is suitable for representing BAS metadata, whether representing BAS control, communication, constraints, and other data sources like real-time data using the IFC data model is appropriate or not remains to be a concern. Limitations in data mapping, prototype test implementation, tools, and representing other data sources are discussed in this section. The result of this study includes IDM, MVD, and implementation of a prototype test.

6.4.1 IDM

A process model (**Figure 6.3**) was created to capture the BAS information exchange from the design phase to the operation phase. This process model together with the BACnet protocol was explored to identify ERs. Altogether, the authors identified 395 information units including 25 BACnet Object Types and 370 property identifiers as listed in the ER_BACnet MVD.xlsx. as linked Mendeley data [107], sample data can be viewed in **APPENDIX G. SAMPLE ER_BACNET MVD.XLSX**. Corresponding FPs were mapped to these information units with importing/exporting requirements. The IDM process facilitated information exchange from the user level to the technical schema level. Consequentially, the Object Types, Property Identifiers, Property Datatype, and Conformance Code in BACnet protocol transformed into IFC entities, attributes, IFC data types, and importing/exporting requirements.

6.4.2 MVD

A BACnet View was created using IfcDoc to document all collected information in the IDM process. Twenty-five BACnet Object Types indicated 12 entity usages in the BACnet View. Concept usages were applied to entity usages with importing/exporting requirements as shown in **Figure 6.5** Concept usages hosted information about BACnet Object Types and property identifiers. However, default Concept Templates had limitations to fully express the relationship between targeting IFC entities. To tackle these limitations, several custom-made Concept Templates were defined in **Table 6-2**. Although there are some similar default Concept Templates in the baseline file, these default concept templates expressed relationships for IFC entities that fall in various inheritance branches with mapped BACnet entities. The complete custom-made concept templates were documented in BACnet_MVD_Final.ifcdoc file as linked Mendeley data [107]. As some BACnet property identifiers cannot be represented with default IFC properties, custom-made property sets, properties, and property enumeration were made to solve this issue as shown in **Table 6-3**. Altogether 22 property sets and 362 properties were defined, a complete list of property sets, properties, and property enumeration was shown in BACnet_MVD_Final.ifcdoc file. Also, the sample MVD documentation (BACnet-MVD Documentation folder) in the HTML version is attached as Mendeley data [107]. The MVD can be accessed by open the index HTML document in the folder.

Table 6-2 Limitations and Custom-made Concept Templates

Limitation	Custom-made Concept Templates	Concept Templates
Decomposition of BACnet devices and objects	BACnet Object Definition Composition	IfcObjectDefinition.IsDecomposedBy—IfcRelAggregates.RelatingObject—IfcController.Decomposes
Composition of BACnet devices and objects	BACnet Object Definition Decomposition	IfcObjectDefinition.Decomposes—IfcRelAggregates—IfcController.IsDecomposedBy
Property identifiers in BACnet Calendar Object Type	BACnet Controller assigns to Work Calendar	IfcObjectDefinition.HasAssignments—IfcRelAssignsToControl.RelatingControl—IfcWorkCalendar

Table 6-2 Continued

Property identifiers in BACnet Command Object Type	BACnet Procedure Assign to Controller	IfcProcedure.HasAssignments—IfcRelAssignsToControl.RelatedObjects—IfcObject
Property identifiers in BACnet Event Enrollment Object Type	BACnet Event Assign to Controller	IfcEvent.HasAssignments—IfcRelAssignsToControl.RelatedObjects—IfcController

Table 6-3 Summary of Custom-made Property Sets and Property Enumeration

Property Set	Property Enumeration
Pset_BACnetAnalogInput Object	PEnum_BACnetAction
Pset_BACnetAnalogOutput Object	PEnum_BACnetDestination
Pset_BACnetAnalogValueObject	PEnum_BACnetDeviceObjectPropertyReference
Pset_BACnetAveragingObject	PEnum_BACnetDeviceStatus
Pset_BACnetBinaryInputObject	PEnum_BACnetEventParameter
Pset_BACnetBinaryOutputObject	PEnum_BACnetEventState
Pset_BACnetBinaryValueObject	PEnum_BACnetEventTimeStamp
Pset_BACnetCommandObject	PEnum_BACnetEventTrasitionStamp
Pset_BACnetDeviceObject	PEnum_BACnetEventTrasitionBits
Pset_BACnetEventEnrollmentObject	PEnum_BACnetEventType
Pset_BACnetFileObject	PEnum_BACnetFileAccessMethod
Pset_BACnetLifeSafetyPointObject	PEnum_BACnetLifeSafetyMode

Table 6-3 Continued

Pset_BACnetLifeSafetyZoneObject	PEnum_BACnetLifeSafetyOperation
Pset_BACnetLoopObject	PEnum_BACnetLifeSafetyState
Pset_BACnetMulti-stateInputObject	PEnum_BACnetLoggingType
Pset_BACnetMulti-stateOutputObject	PEnum_BACnetLogRecord
Pset_BACnetMulti-stateValueObject	PEnum_BACnetNotifyType
Pset_BACnetNotificationClassObject	PEnum_BACnetPolarity
Pset_BACnetProgramObject	PEnum_BACnetPolarityArray
Pset_BACnetPulseConverterObject	PEnum_BACnetProgramState
Pset_BACnetScheduleObject	PEnum_BACnetReliability
Pset_BACnetTrendLogObject	PEnum_BACnetSegmentation
	PEnum_BACnetServiceSupported
	PEnum_BACnetSilencedState
	PEnum_BACnetStatusFlag

6.4.3 Implementation of Prototype Test

BACnet object types were modeled as Revit families. Several Revit families include BACnet Device, Analog Input Object, and Analog Output Object were constructed with user-defined attributes and share parameters indicating exporting settings as a test scenario. The prototype test has successfully exported IFC entities, predefined types, and other attributes following the BACnet MVD. As shown in **Figure 6.12** the extracted IFC physical file showed exported IFC entities' names and predefined types obeying the exporting settings. In terms of importing tool, based on the BACnet MVD, the IfcXML file was automatically parsed with CSS file into the web browser. In this way, the importing tool has achieved the goal to display desired BAS information including IFC entities, global unique identification, name, predefined type, and other required properties in BACnet protocol. Also, the exported ifcXML or IFC models can facilitate BAS information exchange in: i) integrate with BAS tools like Metasys and Niagara; ii) connect with data in other domain such as real-time sensor readings; iii) integrate with different data models such as Linked Data [141] and JSON [142]; iv) sharing information between various project stages including design, construction, operation, etc. for BAS systems.

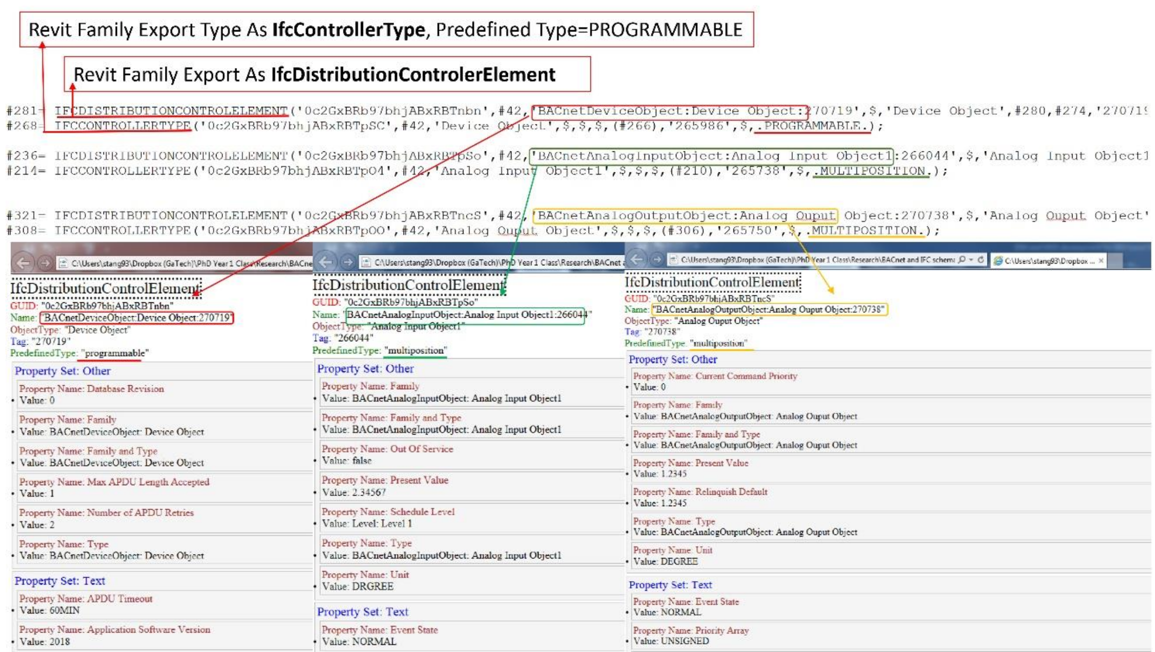


Figure 6.12 Exporting and Importing BAS Information in Prototype Test

6.4.4 Limitations

IFC enables information exchange throughout the entire project lifecycle. A single software or a procedure cannot produce all useful data. Thus, data integrated from multi-software and project stages is a necessity. IFC plays a critical role in information exchange and interoperability for the construction industry. This study has leveraged IDM/MVD methodologies to represent BAS information conforming to the BACnet protocol in the IFC data model so that BAS information can be exchanged among BIM tools and FM tools throughout various project stages. However, there are several limitations which may bring insights to future studies.

Data mapping: Although this study has mapped 25 BACnet Object Types and 370 property identifiers to IFC entities, attributes, and properties, only part of the BACnet Object Types and required property identifiers were involved to demonstrate the possibility of representing BACnet data in IFC data model. Limited information regarding official mapping between BACnet protocol and IFC was only found in IFC 2x4 [143] without updating in recent years. There were only 25 Object Types been mapped without providing any information about BACnet property identifiers and services on the IFC website. ASHRAE keeps adding addendums every 2-4 years to update BACnet Object Types and property identifiers to keep up with the evolving BAS world. There are more than 60 BACnet Object Types in the latest BACnet protocol. Since this study has partially mapped BACnet Object Types and property identifiers due to the limited official mapping information provided in IFC 2x4, an additional effort for all Object Types, property identifiers is necessary. This study demonstrated the possibility to represent BACnet data in IFC data model, most data mapping between BACnet Object Types/property identifiers and IFC entities/attributes was manually achieved, automatic ways for data mapping are worth exploring. Another limitation in terms of data mapping is the lack of representation for BACnet services in the IFC data model. BACnet services enable BACnet objects and devices to issue commands for accessing and manipulating information as well as providing additional functions for applications [144]. The BACnet services, which hold information about control and communication between BACnet objects, were not represented using the IFC data model. Both BACnet and IFC represent BAS information using object-oriented modeling. The result of this study showed that IFC is suitable for representing BAS metadata, whether representing BAS control and communication using the IFC data model is appropriate or not remains to be a concern. Moreover, this study has a limitation in fully representing constraints of relationships between BACnet Object Types and property identifiers. Some constraints between BACnet properties were not represented using the IFC data model. For a constraint instance, when property identifier A has a value "true", property identifier B needs to be value "b", cannot be represented. Furthermore, existing IFC entities did not satisfy representing all BACnet Object Types, either creating a new IFC entity or using IfcBuildingElementProxy is the current solution. Nevertheless, the certification process for adding new IFC entities is complex and the IfcBuildingElementProxy entity may cause exporting issues for similar objects, new solutions are needed for further exploration.

Implementation of prototype test: This research has utilized a BIM tool and a web browser to demonstrate the implementation of BACnet MVD for BAS information exchange. During the prototype test, some limitations were identified: i) the prototype test was based on limited test scenarios and BACnet Object Types. Only a few BACnet Objects were constructed and were modeled based on non-practical cases. Extending the test scenario with a more complex project can guarantee a more robust research result. Besides, for a more complex testing scenario, a quantified measurement, and systematic evaluation are necessary to assess the efficiency of this method. In addition, more complex testing cases are necessary to evaluate the completeness and robustness of this IDM and MVD; ii) Export settings in Revit contains official buildingSMART MVD like IFC4 Design Transfer View, IFC 2x3 Coordination View, and IFC4 Reference View. Since the BACnet view was newly proposed, the exporting process required manual set up for each object. This process is time-consuming and cannot create a one-fits-all solution. Besides, Revit has a limitation in exporting family properties to desired IFC entities. An export plug-in following the BACnet MVD may be a potential solution for automatic data transformation; iii) to check whether the exported IFC instance file contains necessary BACnet data, the data validation process was done manually. Although the mandatory BACnet data was included in the exported instance file, this process was inefficient and might cause errors. An automatic MVD-based data validation process using tools like IfcDoc should be explored in the future [118][145]; vi) Most of the current BIM tools including Revit have the limitation in creating object connectivity so that BACnet objects/devices decomposition and composition cannot be explicitly indicated.

Tool limitation: The implementation of the prototype test for this study was based on Revit and a Web Brower, as BIM-based BAS tool is rare. The current design of the BAS system is either using 2D drawings based on AutoCAD or vendor customized tools. Design BAS using Revit required custom-made families and the manual connection between objects. This process is time-consuming and may not satisfy industry needs. Most available FM tools like Metasys and Niagara are not BIM-based so that IFC data format is not compatible which may cause interoperability issues. Some BIM-based FM tools like EcoDomus and Archibus are not open-source to test the BACnet MVD, further collaboration with software vendors may be the next step.

Other Data Source: BAS information exchange is more complex than just metadata. Time-series data, which record continuous readings from BAS sensors and meters, is another important source of data. IFC was designed to store building contextual data such as building geometry, material properties, as-built construction detail, and HVAC specifications. As this study shows, the IFC data model is suitable to represent BAS metadata. However, representing real-time data in the IFC data model may not be an appropriate approach. Real-time data recorded from BAS sensors are time-series data, which can be effectively stored and manipulated in a relational data model [5]. Hence, integration between the time-series data model and IFC needs further studies. Besides, this study only explored information exchange between design and operation at a macro level. There are other data sources relating to BAS require consideration. For example, BAS construction phase data and data produced from interaction with other disciplines. Besides, there are other communication protocols such as LonWorks, EIB/KNX, and MODBUS for BAS. This study only considered the BACnet protocol. Future studies regarding other BAS protocols and the integration of data generated from different domains are worth exploring [146].

6.5 Conclusion

This research demonstrates the possibility to exchange BAS information conforming to the BACnet protocol with the IFC data model for BAS design and operation. This study has successfully leveraged IDM/MVD methodologies to define a subset of IFC schema, which represents BACnet Object Types and property identifiers. The IDM method was utilized to identify the BAS information sharing process and ERs at the user level. ANSI/ASHRAE Standard 135-2016 was used to initiate the identification of ERs needed for BIM assisted BAS design and operation information exchange. The BACnet MVD created in this study has utilized the IfcDoc tool for documentation of exchange definition, entity usage, concept usage, importing/exporting requirement, and property sets. The IfcDoc also enables the defined subset schema to be displayed in HTML and exported as mvdXML for software vendors when transforming data. A prototype test was carried out with a BIM tool and web browser to demonstrate the implementation of BACnet MVD for BAS information exchange between BIM tools and FM tools. In this way, the BAS information represented in IFC standard can be shared among different stakeholders and BAS software through various project stages, connect with data from other domains, and integrate with different data models.

Several limitations involving data mapping, available tools, various data sources integration, and data validation were identified during the implementation of the prototype test. These limitations bring insight for future studies. As the BACnet protocol keeps evolving, data mapping to the IFC data model needs to keep up with the pace. Since this study has partially mapped BACnet Object Types and property identifiers, an additional effort for all Object Types, property identifiers is necessary depending on a use case. This research provides insight into the methodology for enabling such an exchange. BACnet service and constraints between property identifiers are the other aspects to consider. A potential study can be extending the IFC data model and integration with other data models to represent BACnet services and constraints. In terms of prototype test implementation, complex real projects or more test scenarios should be tested. An automatic data validation process for exporting BIM-based BAS data based on BACnet protocol should be explored. Besides, there is a great opportunity to create BIM-based BAS design tools or a plug-in for both importing and exporting requirements following the BACnet MVD. An API or plugin in BIM tools that allows automatic data mapping when exporting and importing BAS information following the BACnet MVD is necessary to guarantee a reliable data validation process [145,147,148]. Further collaboration with current BIM-based design and FM tools vendors may be the next step to enable the seamless exchange of BAS information using the IFC data model between BIM tools and FM tools. Moreover, different data sources such as real-time data, data generated during various project stages, and data produced from interaction with other disciplines need the most suitable data models for representation. Integration between building contextual data, BAS information, and the Internet of Things using extended BACnet/WS and other open-source data models like Project Haystack deserve future investigation [149,150].

The research work presented in this paper brings BAS information that conformed to the BACnet protocol into the IFC realm. The contributions of this MVD are listed below:

1. The significance of designing BAS using BIM tools is addressed, brining insights for the potential of using BIM tools for BAS information exchange.
2. Information units for the BAS design and operation that conformed to the BACnet protocol were extracted from ANSI/ASHRAE Standard 135-2016 to identify ERs. It brings the buildingSMART effort in mapping data between BACnet and IFC a step forward. This study made detailed IFC representation of BACnet protocol including BACnet objects and associated property identifiers using IFC entities, properties, and relationships. Custom-made property sets and concept templates were made in addition to the IFC baseline file to fully represent BAS information.

3. The BACnet MVD created allows BAS information to be represented in IFC data models which enable information sharing among various BIM tools and project stages.
4. The prototype test demonstrates the possibility to design BAS using BIM tools by selecting BACnet objects with BACnet property identifiers, and exchange BAS information using the IFC data model.
5. The prototype test sets a foundation for software vendors to develop automatic data importing and exporting in BIM tools for BAS information exchange. It also can be the starting point of the software certification process and the data validation process.
6. Limitation for data mapping, tool implementation, and process of prototype test was identified to shed light on future work to bring the BAS design and operation into the BIM cycle. It facilitates data exchange in other domains like Electrical Computer Engineering with the AEC industry, and for the development of IoT and BIM integration.

This research laid a solid foundation for exchanging BAS information conforming to the BACnet protocol with the IFC data model for BAS design and operation. In this way, BAS information represented in the open BIM standard can unlock the potential of future smart building information exchange between various tools throughout multi-project stages and information integration with other domains.

CHAPTER 7 USE CASE FOR FRAMEWORK VALIDATION

In this chapter, the use case in Chapter 6 (BIM assisted BAS design and operation information exchange using BACnet and IFC) is utilized to validate the proposed framework. The proposed framework is validated through a proof of concept procedure. Detail description of the design BAS using BIM tools is in **CHAPTER 6.**

7.1 About the Use Case

Smart buildings are the trend of the next generation's commercial buildings that link different building systems together with the BAS. It is rarely seen to design BAS or exchange BAS information in different project stages using BIM tools. The current design of the BAS system is either using 2D drawings based on AutoCAD or vendor customized tools. Unlike the other building systems, BAS seldom participates in the design-build BIM cycle but blends into facility management in the later stage. To tackle this issue, this use case uses the BACnet MVD and the proposed framework to facilitate information exchange for BIM assisted BAS design and operation. In this way, a BAS system can be modeled based on BIM tools without specifying actual devices from the design stage. The BAS information represented in open BIM standard (IFC) can be serialized as a linked data format for information exchange between various AEC data silos. In doing so, the newly designed BAS information can be linked with building contextual data, existing BAS metadata, and time-series data using the proposed framework. The BAS information can also be exchanged among different stakeholders, between BIM/FM tools, as a knowledge base for other domains using the proposed framework.

7.2 Data Acquisition

The proof of concept validation process uses raw data from Georgia Tech John and Joyce Caddell Building, BAS metadata from campus facility management BAS, and time-series data generated by sensors and announced through Georgia Tech facility management server.

7.2.1 Building Contextual Data

The university has a building model database. Buildings are modeled in Revit. Building contextual information represented in IFC can be exported from Revit for further manipulation with the RDF convertor. **Figure 7.1** shows the building model in Revit. In addition to the building model, data points are modeled as virtual objects in the Revit model as IoT device contextual data.

Figure 7.1 John and Joyce Caddell Building Model in Revit

7.2.2 BAS Metadata

The BAS metadata includes various building subsystems like lighting, electricity, HVAC, etc. These subsystems are characterized by data points, which report values from/to sensors like temperature points. The university data points are stored in a BAS system named MetaSys and categorized by buildings, rooms, and subsystems. **Figure 7.2** shows an example of the data points list for the Caddell building. The raw data points naming conversion follows the university customized way. Hence, understand the naming convention for all data points requires collaboration with the university's facility managers.

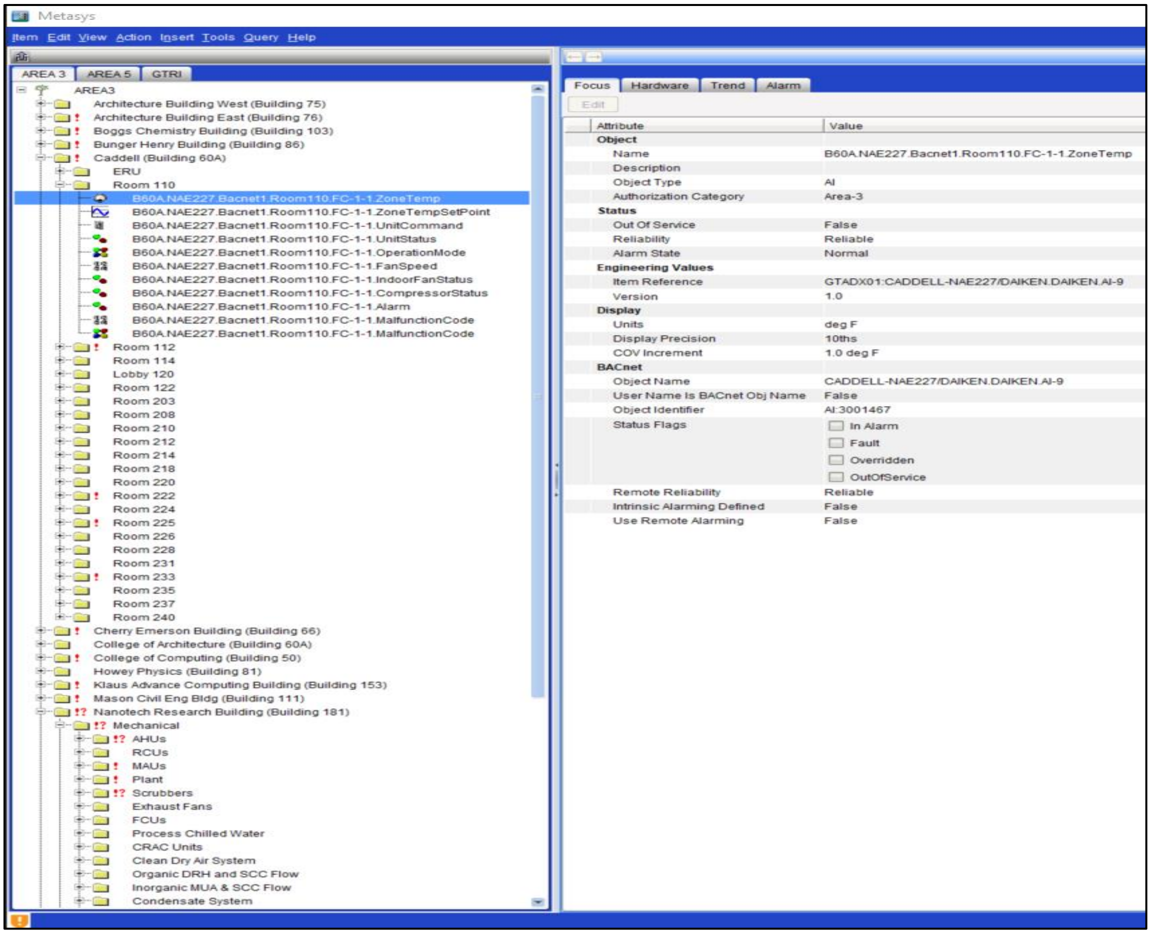


Figure 7.2 MetaSys Screenshot of Data Points in Caddell Building

These data points are analyzed from the naming convention to figure out: i) which subsystem/equipment it belongs to; ii) location: Floor/Space/Zone; iii) control sequence/ dependencies; iv) other relationships; v) types of data points; vi) other information: ID, description, etc. A list of data points with this information is extracted from GT BAS software and represented in the CSV file.

7.2.3 Time-series Data

The data points have associated values from sensors. Time-series data contains both real-time reading and historical data. There are several data sources for time-series data.

The University uses Johnson Control BAS. Real-time reading is sent to Niagara, while the historical data is stored in SkySpark. Real-time reading can be pulled from Niagara using a pyHaystack library in Python script. SkySpark enables direct download of historical data for each data point. **Figure 6.3** shows a screenshot of the Niagara system and **Figure 6.4** shows the SkySpark interface of historical data.

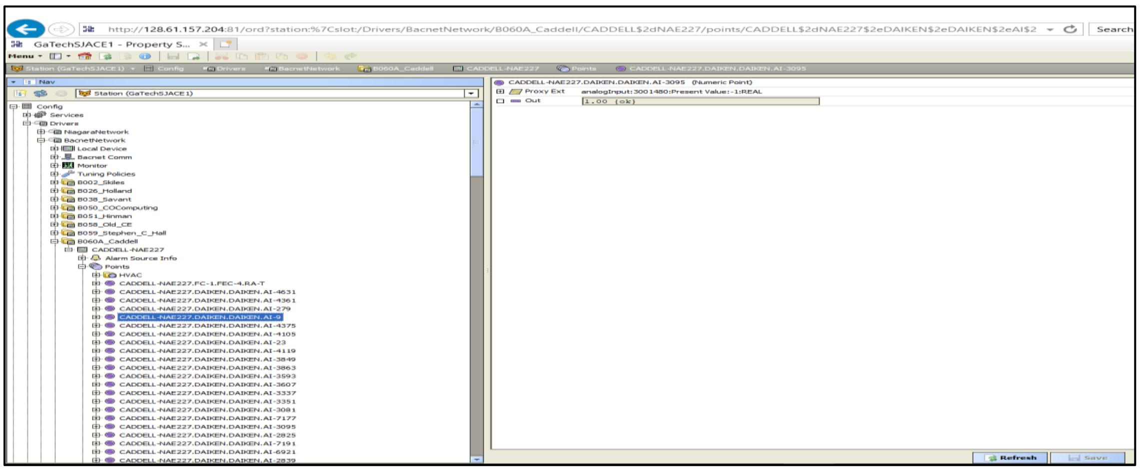


Figure 7.3 Niagara Interface with Data Points Real-time Reading

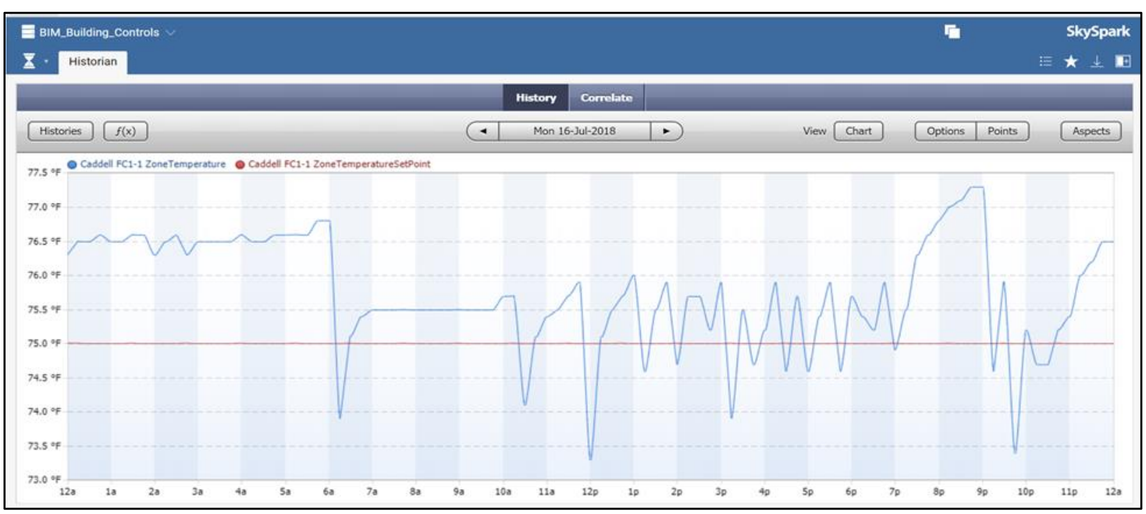


Figure 7.4 SkySpark Interface with Historical Data

The university also announces data points' readings through a relational database server (GT Facility ION Server, **Figure 7.5**). **Figure 7.6** shows an example of time-series data for one of the electricity meters from **Figure 7.5.** Each electricity meter has a unique URL for both real-time and historical time-series data.

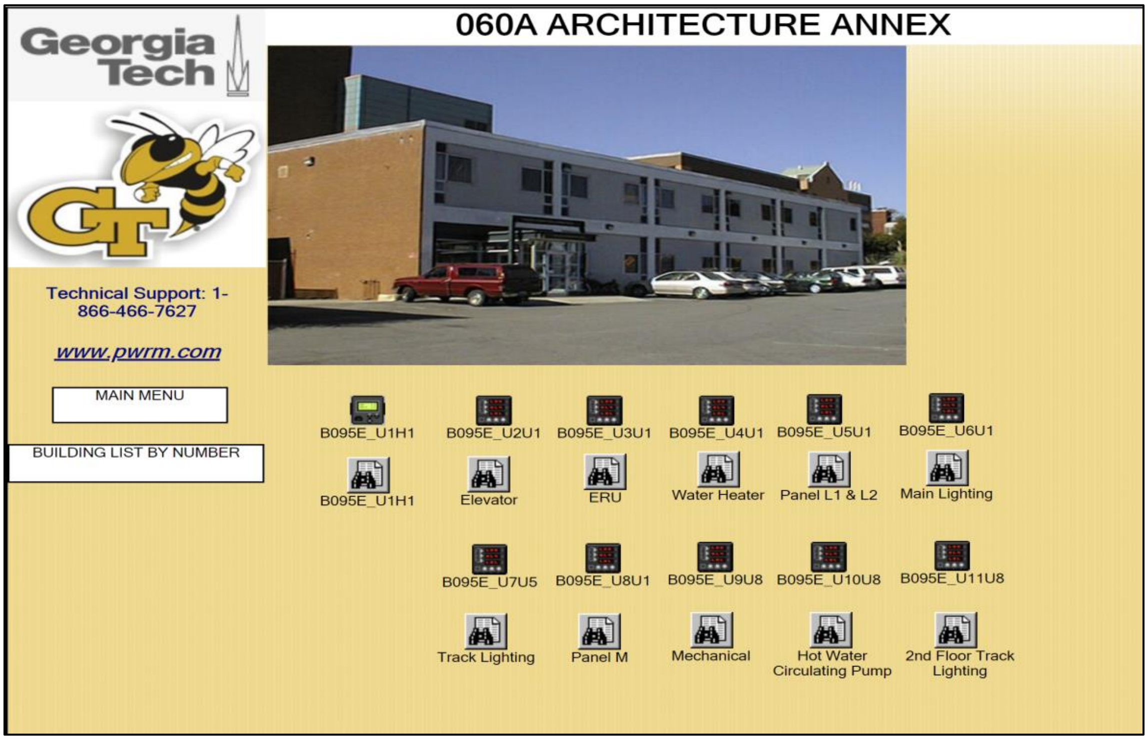


Figure 7.5 GT Facility ION Server-Example of Cadell Building Electricity Meters

Considering the login credential and data accessibility, this use case is using the GT Facility ION Server as the source of time-series data. Both real-time and historical data can be obtained from HTTP links: http://ionsvr2.fac.gatech.edu/ion/default.aspx?dgm=x-pml:/DIAGRAMS/UD/network.dgm&node=GTECH.B051E_MH1&logServerName=QUERYSERVER.IONSVR2&logServerHandle=327952

Device Diagram | Change Date Range | Show Graph

Timestamp	Real Energy Into the Load	Voltage A-B	Voltage B-C	Voltage C-A	Current A	Current B	Current C	Real Power A	Real Power B	Real Power C	Real Power	Apparent Energy Total	Reactive Energy Into the Load	Reactive Energy Out of the Load	Peak Block Demand Real Power
2/13/2020 1:00:00.000 PM	3,553.000	489.800	490.600	485.500	0.000	0.500	0.400	0.000	0.000	0.000	0.000	6,509.000	5,420.000	0.000	1.000
2/13/2020 12:45:00.000 PM	3,553.000	489.600	490.700	485.600	0.000	0.500	0.400	0.000	0.000	0.000	0.000	6,509.000	5,420.000	0.000	1.000
2/13/2020 12:30:00.000 PM	3,553.000	488.900	489.800	485.200	0.000	0.400	0.400	0.000	0.000	0.000	0.000	6,509.000	5,420.000	0.000	1.000
2/13/2020 12:15:00.000 PM	3,553.000	489.100	490.000	485.200	0.000	0.400	0.400	0.000	0.000	0.000	0.000	6,509.000	5,420.000	0.000	1.000
2/13/2020 12:00:00.000 PM	3,553.000	489.300	490.000	485.100	0.000	0.500	0.400	0.000	0.000	0.000	0.000	6,509.000	5,420.000	0.000	1.000
2/13/2020 11:45:00.000 AM	3,553.000	489.500	490.100	485.300	0.000	0.500	0.400	0.000	0.000	0.000	0.000	6,509.000	5,420.000	0.000	1.000
2/13/2020 11:30:00.000 AM	3,553.000	488.000	488.700	483.700	0.000	0.400	0.400	0.000	0.000	0.000	0.000	6,509.000	5,420.000	0.000	1.000
2/13/2020 11:15:00.000 AM	3,553.000	487.900	488.600	483.400	0.000	0.400	0.400	0.000	0.000	0.000	0.000	6,509.000	5,420.000	0.000	1.000
2/13/2020 11:00:00.000 AM	3,553.000	487.400	488.100	483.100	0.000	0.400	0.400	0.000	0.000	0.000	0.000	6,509.000	5,420.000	0.000	1.000
2/13/2020 10:45:00.000 AM	3,553.000	487.300	488.100	483.400	0.000	0.400	0.400	0.000	0.000	0.000	0.000	6,509.000	5,420.000	0.000	1.000
2/13/2020 10:30:00.000 AM	3,553.000	487.200	488.200	483.600	0.000	0.400	0.400	0.000	0.000	0.000	0.000	6,509.000	5,420.000	0.000	1.000
2/13/2020 10:15:00.000 AM	3,553.000	487.600	488.600	483.900	0.000	0.400	0.400	0.000	0.000	0.000	0.000	6,509.000	5,420.000	0.000	1.000
2/13/2020 10:00:00.000 AM	3,553.000	487.700	488.400	483.700	0.000	0.400	0.400	0.000	0.000	0.000	0.000	6,508.000	5,420.000	0.000	1.000
2/13/2020 9:45:00.000 AM	3,553.000	488.100	488.600	483.900	0.000	0.400	0.400	0.000	0.000	0.000	0.000	6,508.000	5,420.000	0.000	1.000
2/13/2020 9:30:00.000 AM	3,553.000	488.100	488.800	483.800	0.000	0.400	0.400	0.000	0.000	0.000	0.000	6,508.000	5,420.000	0.000	1.000
2/13/2020 9:15:00.000 AM	3,553.000	488.600	489.100	484.300	0.000	0.400	0.400	0.000	0.000	0.000	0.000	6,508.000	5,419.000	0.000	1.000
2/13/2020 9:00:00.000 AM	3,553.000	488.500	489.100	484.300	0.000	0.400	0.400	0.000	0.000	0.000	0.000	6,508.000	5,419.000	0.000	1.000
2/13/2020 8:45:00.000 AM	3,553.000	488.400	489.100	484.200	0.000	0.400	0.400	0.000	0.000	0.000	0.000	6,508.000	5,419.000	0.000	1.000
2/13/2020 8:30:00.000 AM	3,553.000	489.000	489.400	484.600	0.000	0.400	0.400	0.000	0.000	0.000	0.000	6,508.000	5,419.000	0.000	1.000
2/13/2020 8:15:00.000 AM	3,553.000	488.900	489.600	485.000	0.000	0.400	0.400	0.000	0.000	0.000	0.000	6,508.000	5,419.000	0.000	1.000
2/13/2020 8:00:00.000 AM	3,553.000	489.100	489.700	485.000	0.000	0.400	0.400	0.000	0.000	0.000	0.000	6,508.000	5,419.000	0.000	1.000
2/13/2020 7:45:00.000 AM	3,553.000	489.200	489.400	484.600	0.000	0.400	0.400	0.000	0.000	0.000	0.000	6,508.000	5,419.000	0.000	1.000
2/13/2020 7:30:00.000 AM	3,553.000	488.400	489.100	484.500	0.000	0.400	0.400	0.000	0.000	0.000	0.000	6,508.000	5,419.000	0.000	1.000
2/13/2020 7:15:00.000 AM	3,553.000	488.400	488.700	484.200	0.000	0.400	0.400	0.000	0.000	0.000	0.000	6,508.000	5,419.000	0.000	1.000
2/13/2020 7:00:00.000 AM	3,553.000	488.500	489.000	484.200	0.000	0.400	0.400	0.000	0.000	0.000	0.000	6,508.000	5,419.000	0.000	1.000
2/13/2020 6:45:00.000 AM	3,553.000	487.900	488.600	483.700	0.000	0.400	0.400	0.000	0.000	0.000	0.000	6,508.000	5,419.000	0.000	1.000
2/13/2020 6:30:00.000 AM	3,553.000	488.500	488.900	483.900	0.000	0.400	0.400	0.000	0.000	0.000	0.000	6,508.000	5,419.000	0.000	1.000
2/13/2020 6:15:00.000 AM	3,553.000	488.700	489.300	484.600	0.000	0.400	0.400	0.000	0.000	0.000	0.000	6,508.000	5,419.000	0.000	1.000
2/13/2020 6:00:00.000 AM	3,553.000	489.200	489.700	484.900	0.000	0.500	0.400	0.000	0.000	0.000	0.000	6,508.000	5,419.000	0.000	1.000
2/13/2020 5:45:00.000 AM	3,553.000	485.100	486.100	481.200	0.000	0.400	0.400	0.000	0.000	0.000	0.000	6,508.000	5,419.000	0.000	1.000
2/13/2020 5:30:00.000 AM	3,553.000	484.800	485.700	480.900	0.000	0.400	0.400	0.000	0.000	0.000	0.000	6,508.000	5,419.000	0.000	1.000
2/13/2020 5:15:00.000 AM	3,553.000	486.100	486.900	482.000	0.000	0.400	0.400	0.000	0.000	0.000	0.000	6,508.000	5,419.000	0.000	1.000
2/13/2020 5:00:00.000 AM	3,553.000	486.900	487.600	482.900	0.000	0.400	0.400	0.000	0.000	0.000	0.000	6,508.000	5,419.000	0.000	1.000

Figure 7.6 Example of Electricity Meters Time-series Data Announcement Through GT Facility ION Server

7.3 Implementation Procedures

The overall implementation procedures are shown in **Figure 7.7**. Nine steps are described in this section in detail. The process starts with construction BACnet objects to represent a BAS system together with other building geometry in Revit. Both the building contextual data and BACnet object data are exported according to the BACnet MVD. This data is then serialized according to the BOT ontology into turtle triples. The fourth step is to collect raw BAS metadata as a CSV file which then is serialized according to BRICK ontology into turtle file. The two turtle files are validated through online RDF validators before loading into the graph database. The connection between time-series data and BAS metadata uses the node's property in the graph database. Finally, building contextual data containing BACnet objects, BAS metadata, and time-series data are connected. Query on BACnet objects and their associated BAS metadata and time-series data is achieved through the graph database.

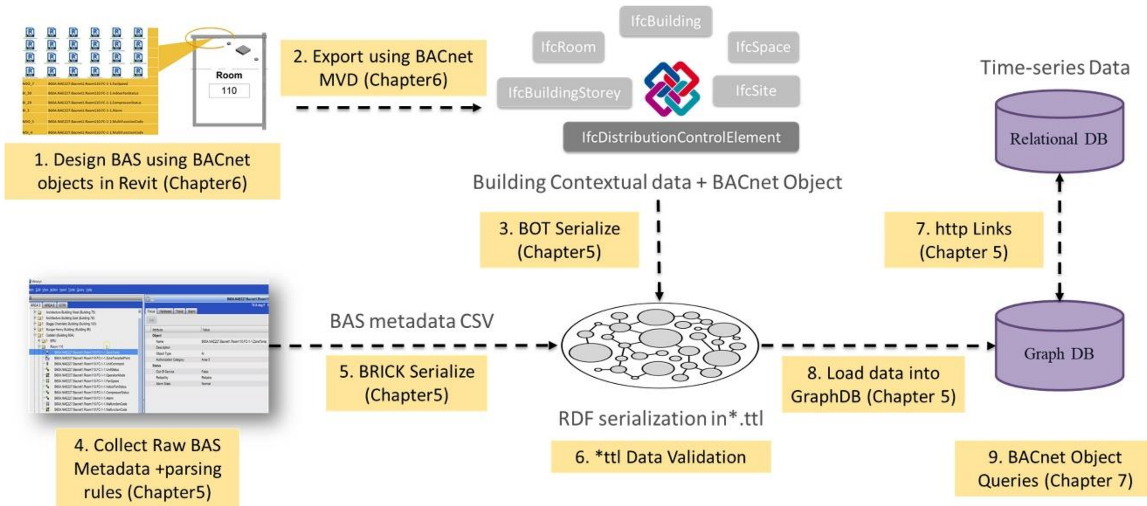


Figure 7.7 Overall Implementation Procedures for the Validation Use case

7.3.1 Design BAS using BACnet Objects in Revit

This step starts with constructing Revit families to represent BACnet objects. These groups of BACnet objects can be arranged and organized to represents the BAS system. For example, in room 110 as shown in Error! Reference source not found., 11 BACnet objects are representing Fan Coil Units (FCU). This FCU does not need to specify vendors at the beginning of the design stage, instead, it's functioning can be represented as a series of BACnet objects. In Revit, the FCU is represented as a collection of BACnet objects families and their properties. Error! Reference source not found. shows one of the 11 BACnet object families which named 'BACnet Analog Input Object' with type 'AI_9'.

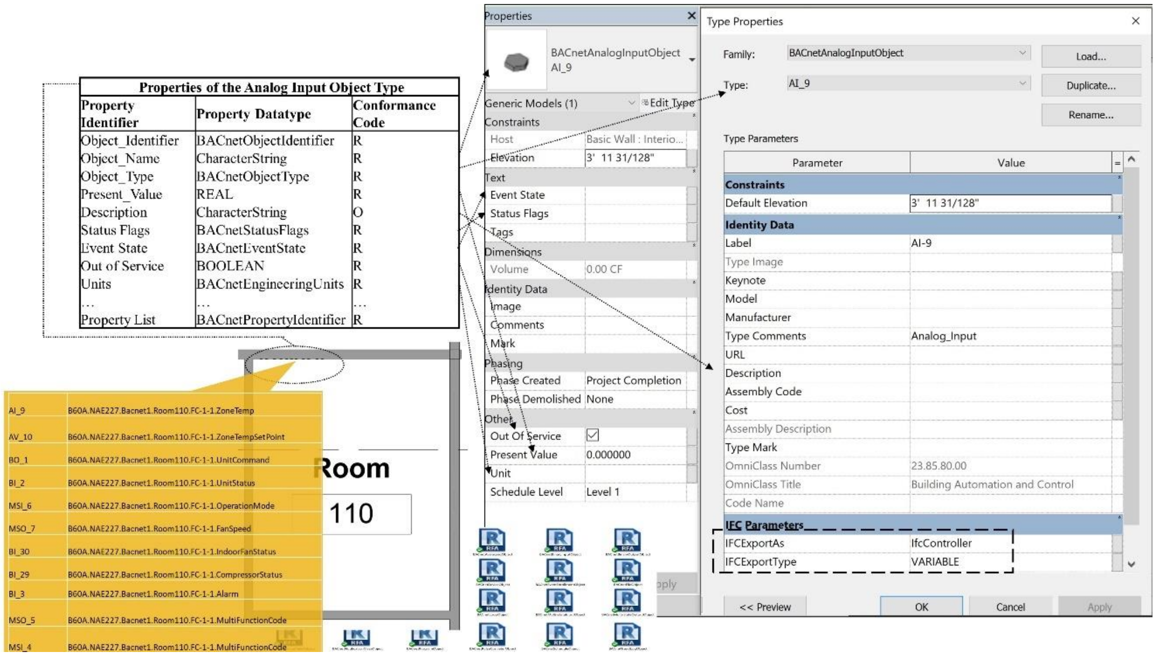


Figure 7.8 Design BAS using BACnet Objects in Revit

7.3.2 Export Building Contextual Data According to BACnet MVD

This step focuses on exporting building contextual data which includes all BACnet objects and their associated properties according to the BACnet MVD. The export process uses 'shared parameters' and 'in-session' to guarantee the mapping between BACnet objects and target IFC entities. Detail description of the export setting is described in **Section 6.3.3**. **Figure 7.9** shows an example of an export IFC file describing the BACnet object 'AI_9' from Error! Reference source not found., the output IFC file shows both the target mapped IFC entities, types, and properties.

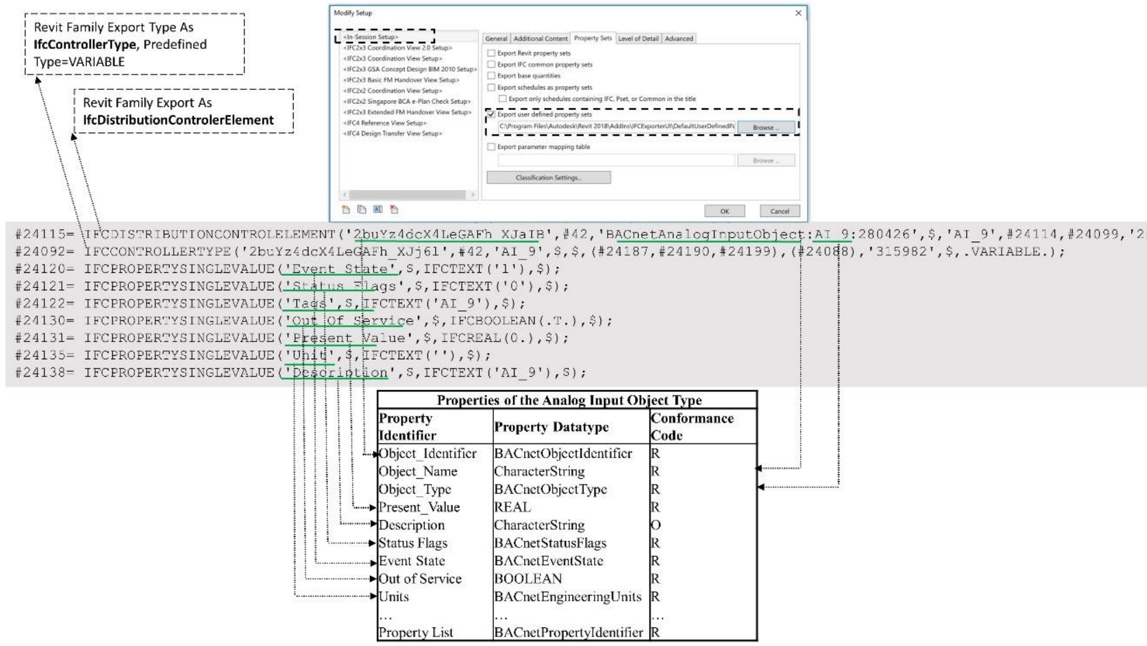


Figure 7.9 Export Building Contextual Data using BACnet MVD

7.3.3 Serialize Building Contextual Data According to BOT

The exported IFC file contains both building geometry and BACnet object information. This IFC file is then serialized into a turtle file according to the BOT ontology using the IFCtoLDB java component. The namespace for building contextual data is: inst: <https://dbl.gatech.edu/project/BIM_BAS_IoT_Contextual#>. For example, in **Figure 7.10**, IFC entities related to building topology are serialized as BOT concepts like *bot:Site, bot:Building, bot:Storey, bot:Space*, and *bot:Element*, while the BACnet objects are serialized as *bot:Elements* relate to *bot:Space* using *bot:containsElement* relationship. The output contains building contextual data and BACnet object data as a *ttl file.

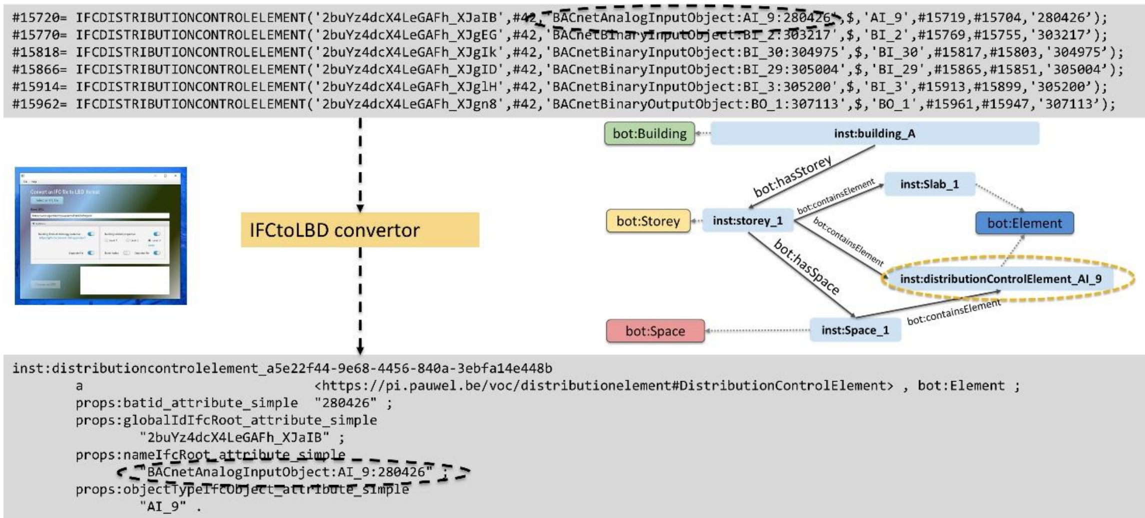


Figure 7.10 Building Contextual Data Serialization using BOT

7.3.4 Collect Raw BAS Metadata and Develop Parsing Rules

This step aims to collect raw BAS metadata from the BAS software to produce a CSV file and parsing rules. The CSV file contains all data points' UUID and vendor names. In this project, the UUID is the BACnet object instance type (e.g. AI_9, BI_10, MSV_31) and the vendor names are directly extracted from the BAS software.

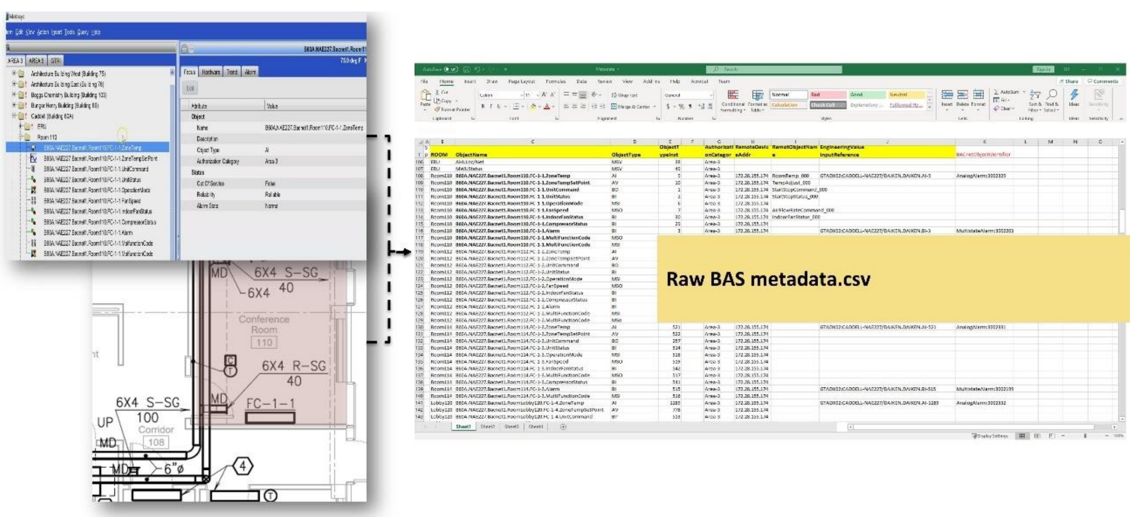


Figure 7.11 Collect Raw BAS Metadata and Develop Parsing Rules

In addition to the CSV file, the step generates parsing rules in JSON files. The full parsing rules are shown in **APPENDIX D. BAS Metadata Parsing Rules**. Altogether, there are three parsing files in JSON:

- BAS raw metadata mapping: the name_point_map.json shows the alignment between raw data points naming characters and BRICK tagsets. As shown in **Figure 7.12,** the key and value in name_point_map.json represent raw BAS data point names and BRICK tagsets respectively. For example, a data points named "B60A.NAE227.Bacnet1.Room110.FC-1-1.ZoneTemp", the character "FC" can be mapped to BRICK schema tagset "Fan_Coil_Unit". To fully understand the naming convention and implied relationships, information from project documentation and facility managers is necessary. For example, several "Points" feeds one "Fan_Coil_Unit" which isConrolledBy a certain "Energy_Revovery_Unit".

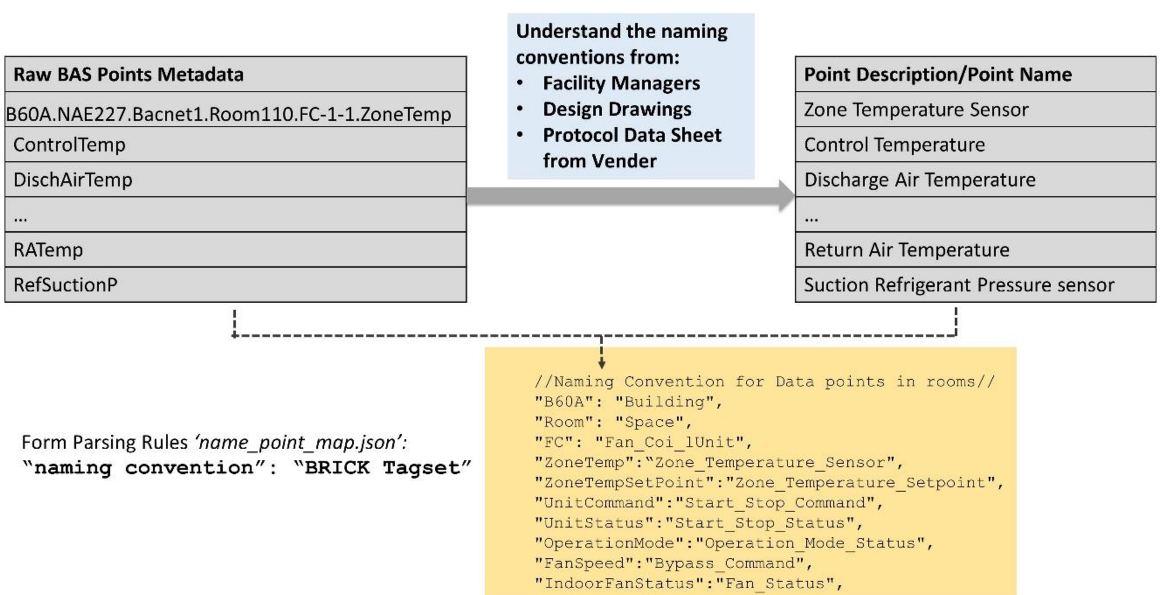


Figure 7.12 Parsing Rules in JSON for BAS Raw Data Points and BRICK Tagsets Mapping

- BRICK room and BOT space map: the space_room_map.json maps the BRICK room instances and BOT space instances using isPartOf relationship in the BRICK schema. As shown in **Figure 7.13**, the keys and values in this JSON represent BRICK room numbers and BOT space ID respectively.

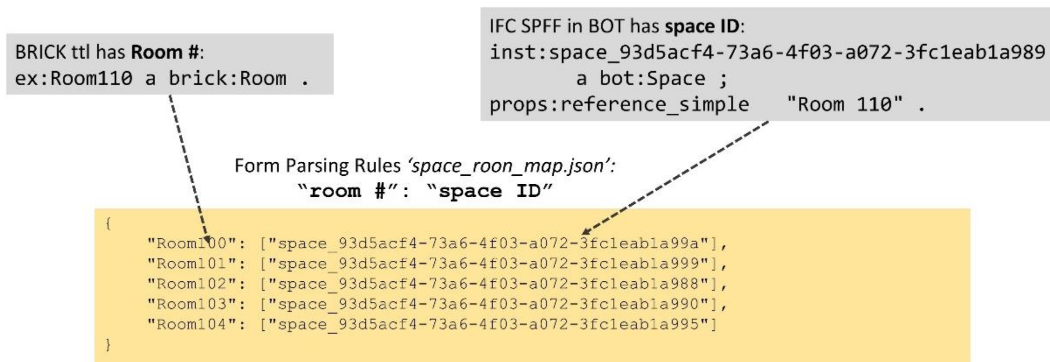


Figure 7.13 Parsing Rules in JSON for BRICK Room and BOT Space Mapping

- Virtual BACnet object UUID in Revit and data point from BAS map: the file is for demonstration of the Use case in **CHAPTER 6**. The botbacnet_uuid_map.json maps the data points name from the BAS system and the virtual BACnet object's UUID from Revit. As shown in **Figure 7.14**, the keys and value in this JSON file represent the BAS data points' names and BACnet object UUIDs in Revit respectively.

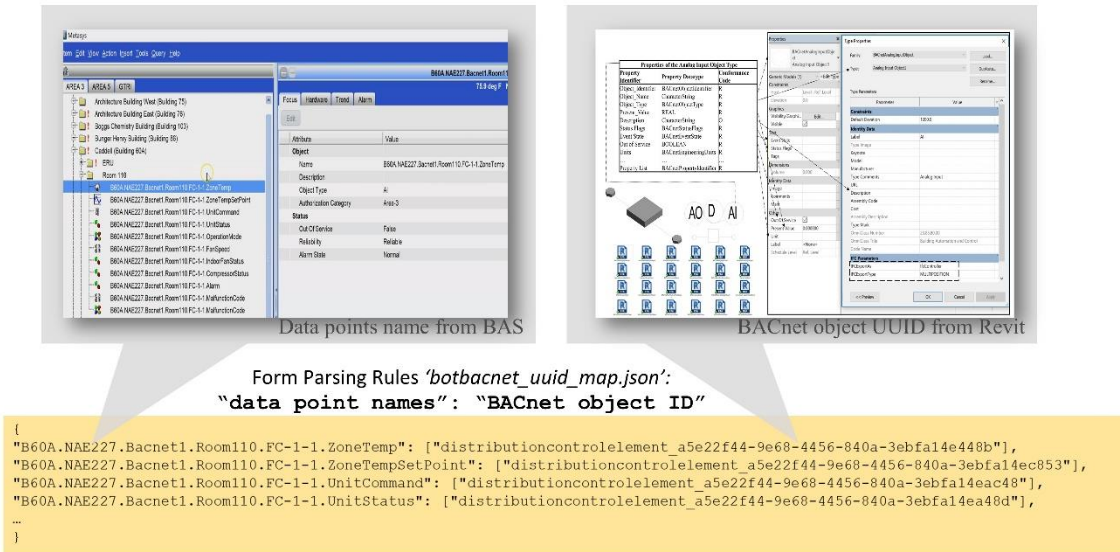


Figure 7.14 Parsing Rules in JSON for Data Point Names and BACnet Object ID Mapping

The outputs of this step are parsing rules in three JSON files and a CSV file containing raw BAS metadata and BACnet object information.

7.3.5 Serialized BAS Metadata

With the output files from the previous step, this step utilized a Python script (**APPENDIX E. Python Script for BAS Metadata Parsing**) to automatically serialized raw BAS metadata into a *ttl file. The namespace for BAS metadata is ex: <https://dbl.gatech.edu/project/BIM_BAS_IoT_Metadata#>. This script creates links between building contextual data and BAS metadata by:

- Linking the BACnet object data from Building Contextual data with the BAS metadata through BACnet object UUID (Parsing Rule *'botbacnet_uuid_map.json'*).
- Linking the space from building contextual data with the room in BAS metadata through *brick:isPartOf* relationship (Parsing Rule *'space_room_map.json'*).

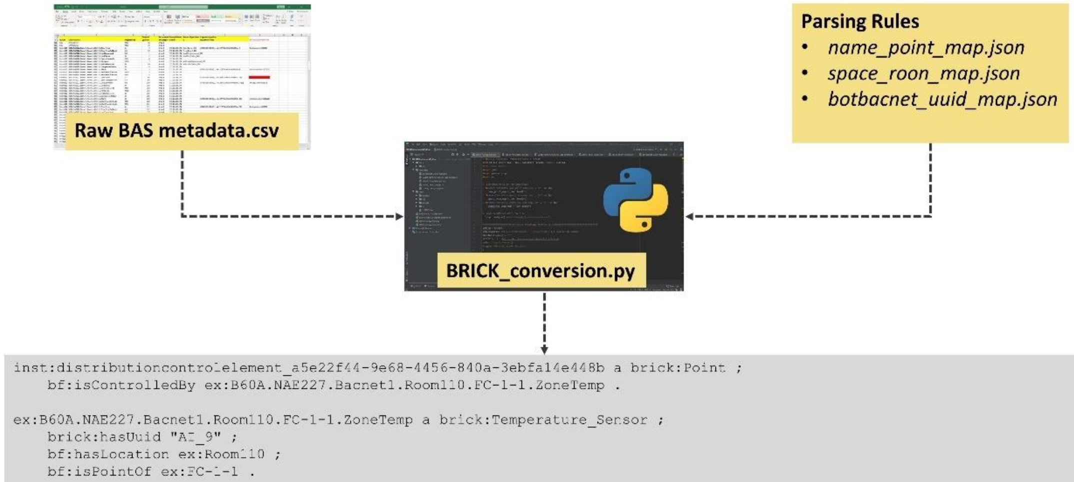


Figure 7.15 BAS Metadata Serialization

As shown in **Figure 7.15**, the BACnet object *'inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14e448b'* from the building contextual data is linked with BAS data point *'ex:B60A.NAE227.Bacnet1.Room110.FC-1-1.ZoneTemp'* which has UUID as *'AI_9'* using *bf:isControlledBy* relationship. An example of BAS metadata serialized in *ttl is shown in **APPENDIX F. An Example of BAS Metadata serialized in *ttl**.

7.3.6 RDF data Validation

The data validation process guarantees the serialized *ttl file is valid as an RDF graph. For building contextual data, the turtle file is validated through an online RDF validator [106]. For BAS metadata, it is validated through the BRICK TTL viewer [151]. The two turtle files are validated as shown in **Figure 7.16**. the left side shows the BRICK tagsets and relationships used in the BAS metadata turtle file, the right side shows that the building contextual data*ttl is validated without any error.

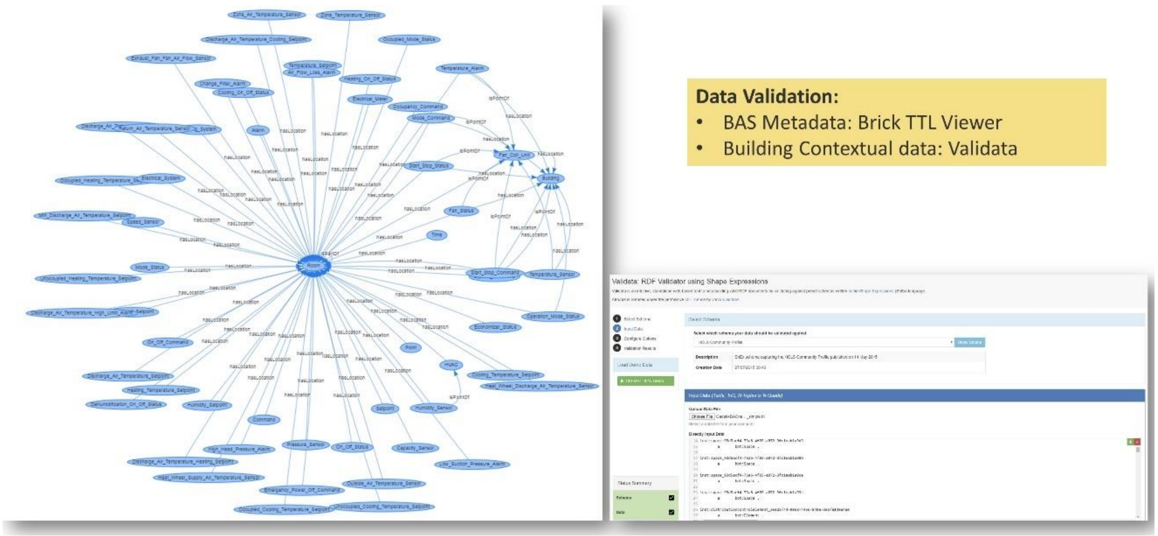


Figure 7.16 RDF Data Validation

7.3.7 Create Time-series data link

The GT Facility ION server provides unique HTTP links for each of the data points. Those links announce the data points reading at fifteen minutes interval. Those links also connect with historical time-series data. In this way, these unique links are also serialized into BAS metadata using *bf:hasInput* relationship as shown in **Figure 7.17**. for example, in the following text, the data point "*ex:B60A.NAE227.Bacnet1.Room110.FC-1-1.ZoneTemp*" links to an HTTP URL using *bf:hasInput* relationship.

```
     ex:B60A.NAE227.Bacnet1.Room110.FC-1-1.ZoneTemp        a
brick:Zone_Temperature_Sensor ;

         brick:hasUuid "AI_9" ;

         bf:hasLocation ex:Room110 ;

         bf:isPointOf ex:FC-1-1 .

         bf:hasInput
"http://ionsvr2.fac.gatech.edu/ion/default.aspx?dgm=OPEN_TE
MPLATE_DIAGRAM&node=GTECH.B095E_U10U&logServerName=QUERYSER
VER.IONSVR2&logServerHandle=327952" ;
```

```
bf:hasLocation ex:Room110 .
```

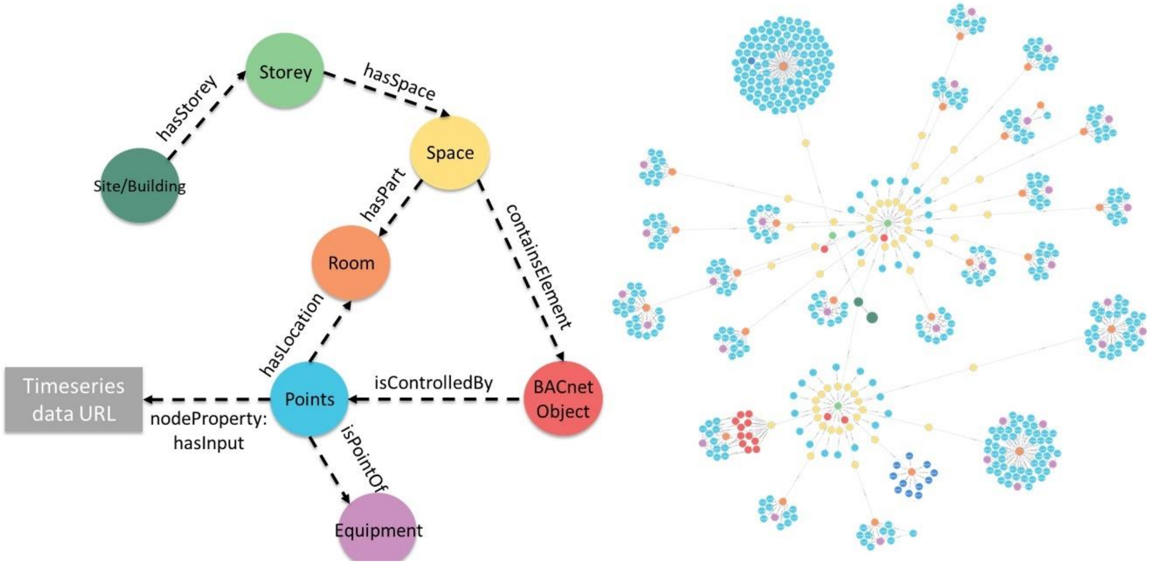


Figure 7.17 Time-series and BAS metadata link

7.3.8 Load RDF data into Graph Database

This step aims to load the two *ttl files into a graph database system named Neo4j. Neo4j represents data using nodes and edges. There is a java library/plugin that enables the use of RDF in Neo4j name NSMNTX-Neo4j RDF & Semantic toolkit [152]. The following queries enable importing turtles files into the Neo4j server.

```
Neo4j: import RDF into graph

CALL semantics.importRDF("file:///C:/Users/Shu
Tang/Desktop/Cadell+BACnetObject_LBD.ttl","Turtle")

CALL semantics.importRDF("file:///C:/Users/Shu
Tang/Desktop/BAS_Metadata_Export.ttl","Turtle")
```

The result shows one large RDF graph (**Figure 7.18**) containing building contextual data, BAS metadata, and the links to time-series data in a relational database.

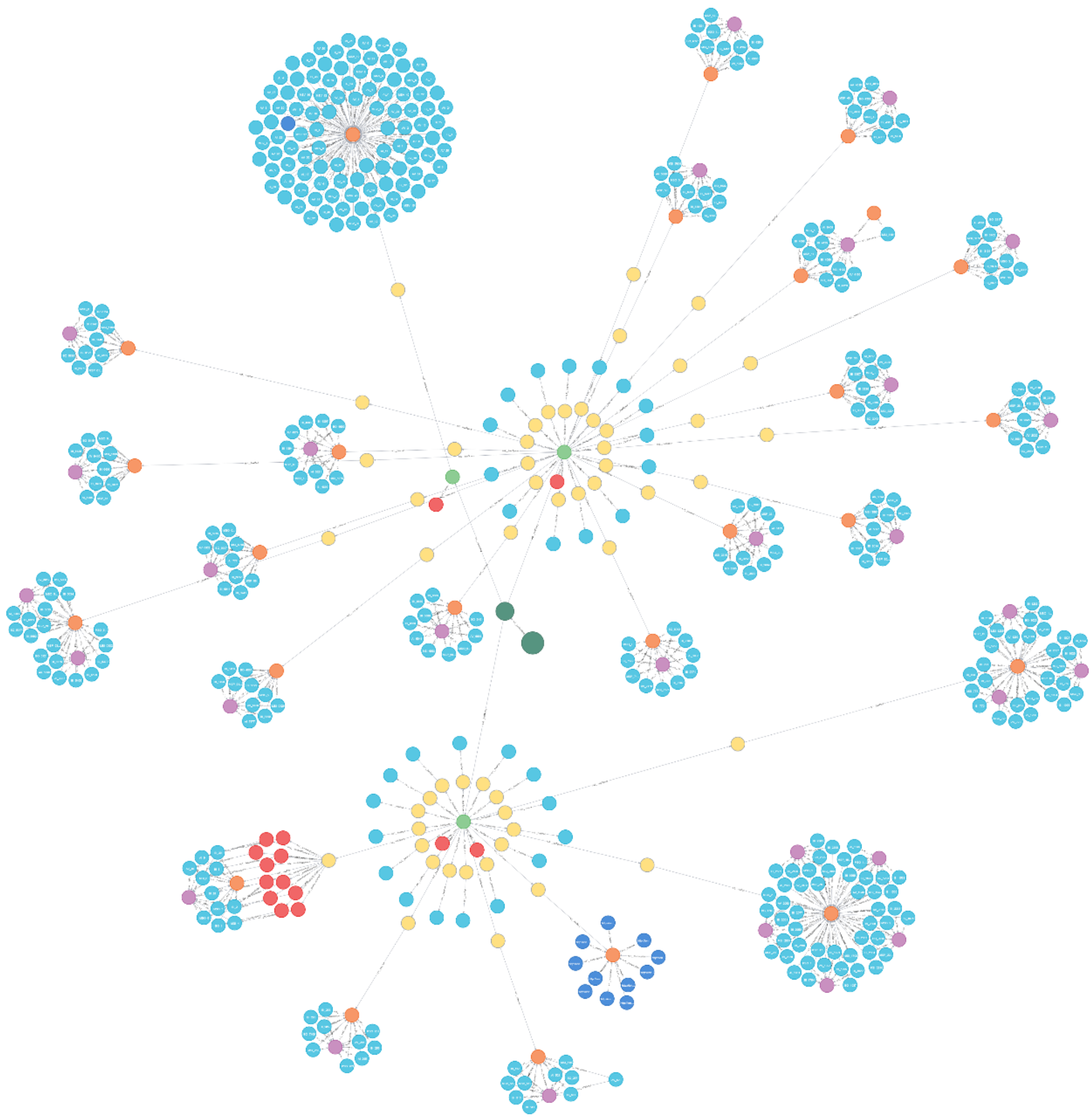


Figure 7.18 Integrated RDF Graph Containing Building Contextual Sata, BAS Metadata and Time-series Data Links.

The coloring conventions are:

- Building contextual data:
    - Green: Site, Building, Storey

    - Yellow: Space
    - Red: BACnet Objects
- Building contextual data:
    - Blue: BAS data points
    - Orange: Rooms
    - Purple: Equipment

7.3.9 Query BACnet Object in Graph Database

After the *ttl file loaded into the database system, Neo4j enables various queries using a query language called Cypher. With the NSMNTX plug-in, Neo4j also enables query RDF using SPARQL. This section demonstrates several sample queries on the BACnet objects.

For example in **Figure 7.19**, to visualize all the BACnet objects in Room 110 and their associated data points and time-series data readings. For building contextual data, the yellow node represents room 110 is connected with eleven BACnet objects represented in red nodes. The blue dotted oval contains the BAS metadata, blue node are the data points which are associated with fan coil unit FC-1-1. The time-series data is linked through HTTP links as a data point's node property (hasInput).

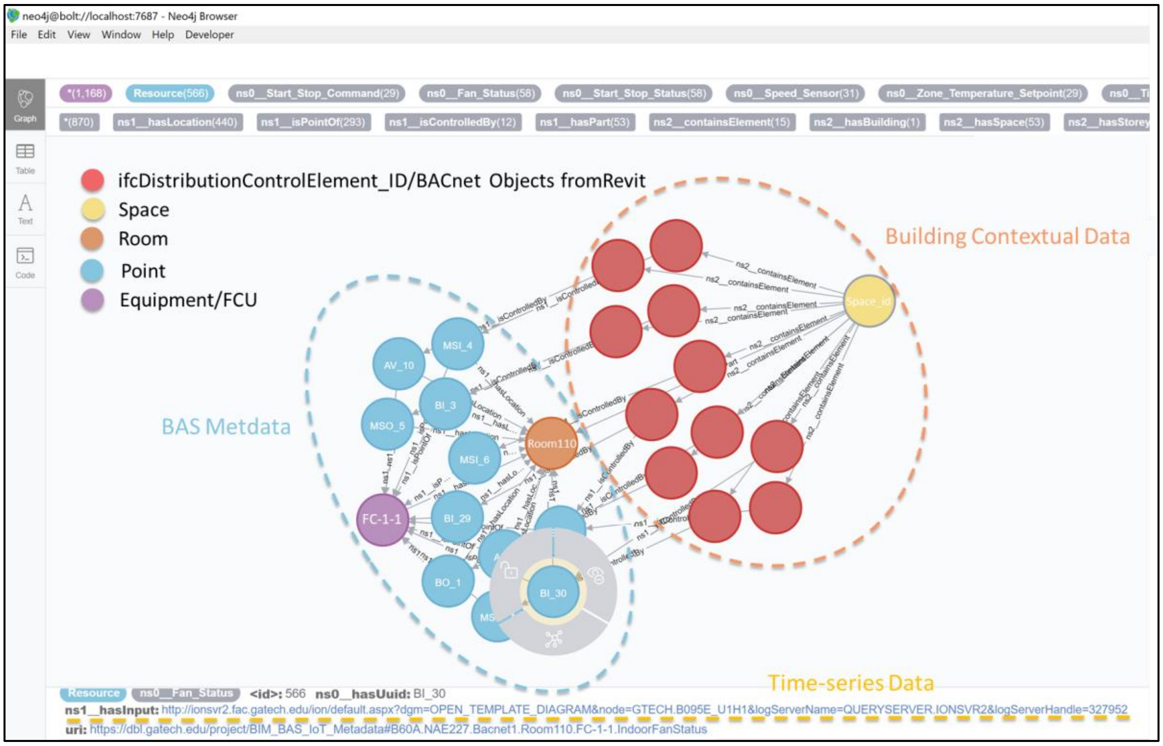


Figure 7.19 Visualizing BACnet Object and Associated BAS Data Points and Time-series data in Neo4j

Some other sample queries are:

- Example 1, find all the HTTP links for all BACnet objects: **Figure 7.20**

```
MATCH (n) WHERE EXISTS(n.ns1__hasInput) RETURN DISTINCT
"BACnetObject" as entity, n.ns1__hasInput AS ns1__hasInput
LIMIT 25 UNION ALL MATCH ()-[r]-() WHERE
EXISTS(r.ns1__hasInput) RETURN DISTINCT "relationship" AS
entity, r.ns1__hasInput AS ns1__hasInput LIMIT 25
```

Figure 7.20 Result for Example Query 1

- Example 2, find all the UUID for all BACnet objects: **Figure 7.21**

```
MATCH (n) WHERE EXISTS(n.ns0__hasUuid) RETURN DISTINCT
"BACnetObject" as entity, n.ns0__hasUuid AS ns0__hasUuid
LIMIT 25 UNION ALL MATCH ()-[r]-() WHERE
EXISTS(r.ns0__hasUuid) RETURN DISTINCT "relationship" AS
entity, r.ns0__hasUuid AS ns0__hasUuid LIMIT 25
```

```
$ MATCH (n) WHERE EXISTS(n.ns0__hasUuid) RETURN DISTINCT "BACnetObject" as entity, n.n…
```

entity	ns0__hasUuid
"BACnetObject"	"BO_5121"
"BACnetObject"	"MSI_774"
"BACnetObject"	"BO_5377"
"BACnetObject"	"BO_1"
"BACnetObject"	"MSI_1798"
"BACnetObject"	"BI_4893"
"BACnetObject"	"MSI_3846"
"BACnetObject"	"MSP_5639"
"BACnetObject"	"AV_10"

Started streaming 25 records after 24 ms and completed after 24 ms.

Figure 7.21 Result for Example Query 2

In this way, the BACnet object information from building contextual data, BAS metadata, and time-series data are connected. This use case successfully demonstrates information exchange between BIM, BAS, and IoT for BIM assisted BAS design using the proposed framework.

7.4 Other Potential Use cases

This section describes several potential use cases that can be implemented using the proposed framework. These use cases are not intended to serve as an exhaustive exploration of the data silos as indicators for professional assessment like building performance assessment or energy simulation but as a demonstration for the potential use cases for the proposed framework and how different data silos in AEC can be linked via semantic web technologies for various purposes.

7.4.1.1 Actual energy consumption vs simulated energy consumption for future facility maintenance and remodel at room level or equipment level

This use case intends to link contextual data from BIM, time-series data from BAS readings of energy consumption for mechanical systems in buildings, and the output from the EnergyPlus model for a particular zone (room) or a piece of equipment in a time (winter/summer). The objective is to compare traditional analysis based on data from the actual reading and the analysis based on output from the E+ model. In this way, meaningful additional engineering information can be extracted for facility managers to make informed decisions for future remodel or maintenance to improve energy efficiency.

This use case utilizes a rule-based performance assessment mechanism called the scenario-modeling method [5], which presents different aspects of building performance in parallel to discover holistic perspectives on global and local performance. A scenario model can reflect some key concerns in the area of simulated energy vs. actual energy consumption. A window form UI can present a comparison between the simulated energy consumption of selected room or equipment and actual time-series reading from BAS as shown in **Figure 7.22**.

By coupling the outputs of a calibrated energy simulation model with available data from the BAS, facility managers can examine a certain energy consumer at equipment or room level without additional workload or cost, it can be utilized as the first step for building service system optimization.

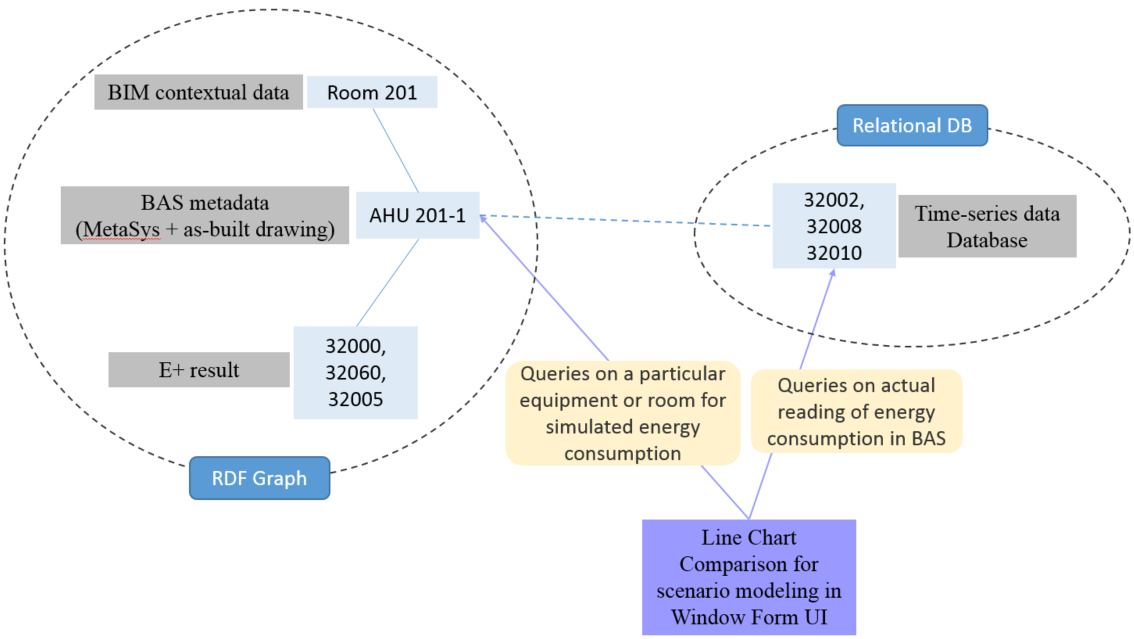


Figure 7.22 Simulated Energy vs Actual Energy Consumption

7.4.1.2 <u>Preparing Asset information Model as CDE</u>

This use case aims to use the data serialized based on the proposed framework to prepare Asset Information Model (AIM) according to BSI_PAS_1192_2014 standard [153]. The PAS specification indicates the data transfer process to prepare the AIM concerning the operation and maintenance of assets. PAS standard specifies the implementation processes to provide the Common Data Environment (CDE) as shown in **Figure 7.23**. The

CDE provides a collaborative environment where all parties can share works to form the foundation for the information management processes.

Data related to BIM contextual information and BAS metadata can be extracted from the RDF graph to prepare as a source for an AIM related to BAS for facility operation and maintenance. This AIM can then go through the PAS process including approval, authorization, and verification to ensuring data and information governance and assurance for the CDE [154].

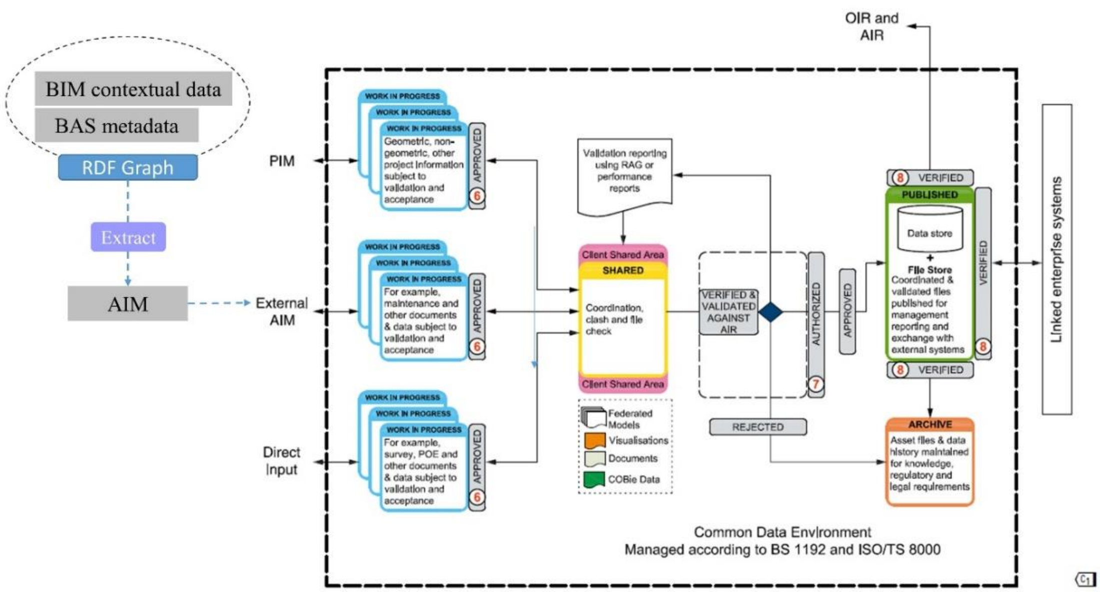


Figure 7.23 Processes to prepare AIM from the RDF graph [153]

CHAPTER 8 CONCLUSION

This research proposes a framework for BIM building contextual data, BAS metadata, and IoT devices' time-series data exchange through semantic web technology. Part of this chapter is published in [149].

8.1 Contribution and Impacts

This research proposes a framework for BIM building contextual data, BAS metadata, and IoT devices' time-series data exchange through semantic web technology. The major contributions and impacts of this research include:

- The literature review in this research contributes to the body of knowledge by presenting an in-depth review of BIM and IoT devices integration in the AEC industry from domain application perspective and integration methodologies. Apart from the summarized application domains, the authors summarized five integration methods with description, examples, discussion, and suggested the current optimal approach [149].
- The proposed framework facilitates information exchange between building contextual information in BIM, BAS metadata, and time-series data generated from IoT devices. This framework lays a solid foundation for future studies utilizing BIM, BAS, and IoT data such as facility management, energy bench marking, and fault detection. The overall methodology can also inspire other data silos such as construction management data, logistic data, climate data, simulation data, etc integration, thus stimulate new applications. It sets a foundation for data exchange

between isolated information islands in the AEC industry, and brings insights for cross-domain data interlinking.

- The proposed framework demonstrates how to reuse existing ontologies for new applications. The data serialization process can be a guideline for future researchers to implement semantic web technologies for various purposes.
- The developed BACnet MVD enables information exchange for BIM assisted BAS design and operation using BACnet and IFC. This MVD lays a solid foundation for exchanging BAS information conforming to the BACnet protocol with the IFC data model for BAS design and operation [107].
- The framework validation process demonstrates the data silos interlinking for BIM assisted BAS design and operation. In this way, BAS information represented in the open BIM standard or linked data formats can unlock the potential of future smart building information exchange between various tools throughout multi-project stages and information integration with other domains.

8.2 Limitation and Future Research

The proposed framework aims to achieve information exchange between building contextual data, BAS metadata, and time-series data. The validation for the proposed framework is based on a 'BIM assisted BAS design and operation information exchange using BACnet and IFC' use case. The proof-of-concept use case implementation with the proposed framework has several limitations:

- Limitation in building contextual data: the building contextual data contains building topology information related to entities like site, building, storey, space, part of the elements, and spatial relationships. Other building contextual data such as building geometry, all building elements are not with on the project scope. Although the full set of building contextual data is unnecessary, further test on more complex building contextual data with other ontologies like ifcOWL, PRODUCT, PROPS, GEOM can be tested. Besides, the java component IFCtoLDB convertor has limitations itself. When serializing building contextual data, some adjacency and intersection spatial relationships are not successfully converted. Further exploration in converting the spatial relationship automatically can be considered.

- Limitation in BAS metadata:
    - Georgia Tech stores BAS data points in Metasys in well-organized ways. So, the building project information and BAS data point naming convention can be obtained from facility managers. This thesis only considers the naming convention and project information obtained from facility managers. However, there is other implied information. To form comprehensive parsing rules, more in-depth BAS background knowledge is required in the future to complete full relationships between data points, equipment, and other facilities. For example, the control sequence of various equipment.
    - Also, this thesis used a python script to automatically convert raw BAS metadata into *ttl format based on parsing rules. However, the parsing rules require manual encoding. The fully automatic conversion process is currently an active research topic.
    - BAS metadata is serialized according to the BRICK schema. Although all data points have been assigned with BRICK tagsets, some of these data

points are tagged high-level classes like `brick:Point`. The BRICK schema lacks vocabularies in tagsets to annotate BAS metadata more precisely.

- Limitation in time-series data: the times-series data is store in various BAS system and GT facility ION server. This thesis only uses GT Facility ION server as the source for time-series data. Data accessibility to other BAS software requires a special credential. As a result, not all the BAS metadata data points are connected with time-series data. In the future, the connection between other BAS software time-series data and BAS metadata will be explored.
- Limited testing scenario: the implementation scenario is based on one GT campus building with more than 500 data points. In the future, the scalability of this framework can be tested with more complex scenarios or urban scale projects.
- Limited testing application use cases: this thesis validates the proposed framework based on one use case. In the future, other use cases can be applied for information exchange. Moreover, connect other data silos such as energy simulation, weather, construction progress, cost data, etc using this framework will be an interesting topic. Adding a web-based API between serialized data and databases to scale up the variety of data silos is worth exploring. In this case, user interfaces can be built to send an HTTP request to a neutral REST-ful web service to query different databases [155]. The improvement may facilitate more innovative applications.
- MVD Limitation: A detailed description of the MVD limitation is in **Section 6.4.4**.

This research develops a framework to facilitate information exchange between BIM-based building contextual data, IoT devices' time-series data, and BAS metadata using Semantic Web technology. The research achievements are: **i)** conduct a comprehensive literature review on BIM and IoT integration based on domain of application and integration methods to summarized an optimal current approach; **ii)** propose a framework which enables information exchange among semantically described building contextual data, BAS metadata, and time-series data; **iii)** the proposed framework uses BOT and BRICK schema to describe building contextual data and BAS metadata; **iv)** create an MVD for BIM assisted BAS design and information exchange using BACnet and IFC use case; **v)** validate the framework with the use case and data from Georgia Tech campus building data.

Emerging technologies provide more advanced methods for data silos integration, thus facilitating and laying foundations for potential applications. This framework lays a solid foundation for future studies utilizing BIM, BAS, and IoT data such as facility management, energy bench marking, and fault detection. The overall methodology can also inspire other data silos such as construction management data, logistic data, climate data, simulation data, etc integration, thus stimulate new applications. It sets a foundation for data exchange between isolated information islands in the AEC industry, and brings insights for cross-domain data interlinking.

8.3 The Bigger Picture for BIM and IoT Integration Research Future Directions

The potential of information exchange between various data silos in the AEC industry and other domains is enormous. The proposed framework demonstrates the possibility for information exchange between BIM, BAS, and IoT. The innovations and opportunities related to BIM, IoT, BAS, smart built environment, and other emerging technologies can be foreseen. From the perspective of the broader picture, some potential research directions related to BIM and IoT devices integration are proposed as follows:

8.3.1 SOA and Web Services for BIM and IoT Integration

SOA acts as the premier integration and architecture framework for the complex and heterogeneous computing environment. It uses the concept of software design where various services can be combined to provide functionalities of a large application through a communication protocol over networks. The idea starts with combining various services such as visualizing BIM models, querying real-time sensor data, analyzing sensor readings, and other IoT applications. The SOA unique features such as – service composition, service discovery, asset wrapping, model-driven implementation, loosely coupled and platform-independent – enable information flow, organizational flexibility, and scalability while maintaining internal functionality for each service. The reusability of services can extend and combine other applications with existing services so that software development costs and management time can be decreased. These features can largely benefit the integration of BIM with IoT devices and further extend future applications. Although there is existing research on proposing system architectures for BIM and IoT devices integration [71,78,156], these system architectures are not supposed to be a one-size-fits-all solution. There are still abundant applications of BIM and IoT integration that need new designs of SOA, web services, and integration methods. Web services are building blocks for SOA's service layer. Loosely coupled web services amalgamate semantic information in BIM and feed from sensor networks with various SOA design patterns. The Representational State Transfer (REST) architecture style is often used to utilize and interact with IoT nodes [157]. The design patterns of SOA and Web services offer opportunities for making BIM stateful (e.g. real-time, information update). The potential applications are:

Real-time Model Update based on IoT Device Readings: BIMs provide rich semantic information about building elements but fail to display element states and indoor conditions. Static models become real-time information models if BIM entities' states can be updated by real-time IoT devices' readings. A new design of SOA patterns using RESTful Web Services named RESTful endpoint would be a potential solution to enable BIM entities' status update based on IoT devices' readings. A RESTful endpoint, on one hand, receives readings from IoT nodes, on the other hand, conducts create/read/update/delete (CRUD) operations in the BIM data layer [157]. How to utilize different SOA design patterns to enable BIM entities' status update based on sensor readings will be research to be investigated.

Information Acquisition and Control- A Two-way-Interaction: most of the existing research have already achieved reaching information residing in BIMs and visualize IoT devices readings from models. This multi-source information acquisition and fusion can be done with SOA pattern which utilizes a RESTful service façade (web services) [157]. However, current research only realized one-way interaction (e.g. energy monitoring, IEQ monitoring, building performance monitoring). Only a few studies have explored human-building interaction for cognitive buildings[46,158]. No two-way interaction that involves control of actuators through BIMs has been discovered in the reviewed articles. When combining with control on actuators through Web services, information acquired from BIM and IoT devices offers opportunities for smart cognitive buildings and human-building interaction like emergency response and disaster evacuation. Potential research problems regarding information acquisition and control will be: How to create SOA patterns for information acquisition and control interactions? How these SOA patterns can benefit advanced building technologies like cognitive building and human-building interaction?

Ubiquitous Monitoring and Crowd Sourcing Monitoring: for future smart cities, information residing in BIMs are valuable. City modeling and management application like smart city platforms and city portals can absorb BIMs and IoT. Thousands of APIs, BIMs, IoT devices will involve in SOA design patterns for smart cities. Potential solutions utilizing a RESTful service named Callback Responder can blend with traditional SOA patterns to realize ubiquitous monitoring and crowdsourcing monitoring. Ubiquitous monitoring continuously providing information about building elements and IoT devices regardless of the situation in a 24/7 manner. When an event happens, crowd-sourcing monitoring can produce information about this event and physical condition by IoT devices near the event site [157]. Rich spatial and temporal information gathered from the integrated BIM, GIS, and IoT devices is efficacious for city modeling and management applications. How to blend potential solutions like RESTful service with traditional SOA patterns to realize ubiquitous monitoring and crowdsourcing monitoring that can be applied to applications like smart city energy management [67], urban-scaled facility management and emergency response[56], flood analysis, transportation monitoring and indoor/outdoor positioning[28] is worth exploring.

Integration with other Cutting-Edge Technologies: New technologies such as VR, AR, mixed reality (MR) are leading to greater integration across BIM and IoT. Adding VR/AR/MR web-based application framework like the mobile agent to the SOA design patterns, digital models can be superimposed into the real world for web-based AR applications [159]. A large number of applications using the combination of VR and AR can support effective information flow and display [45]. For example, using AR, BIM and sensors can be beneficial to automatic schedule update [160], safety inspection [161], smart building design [162,163], facility management, construction lifecycle management, smart learning environment for educational institutions. Hence, the study on adding VR/AR/MR web-based application frameworks to SOA patterns to achieve integration between IoT devices and BIM for various applications will continue to be a future direction of interest.

8.3.2 Standards for information integration and management in the AEC industry

With the emergence of new technologies like the web of data and IoT, information diversity and overload will happen [44]. Heterogeneous data sources among different stakeholders, data across different domains, data throughout all phases of lifecycle need to be well handled for different purposes [164]. Furthermore, as the data amount drastically increases over time, it is important to ensure information consistency, traceability, and long-term archiving [165]. A standardization way to integrate and manage data for BIM and IoT integration arises to be a problem [64]. Some effort has been made by the U.S. Department of Commerce's National Institute of Standards and Technology (NIST) by releasing IoT-Enabled Smart (IES) Cities Framework and Framework for Cyber-Physical System (CPS). The continuous developing frameworks aim to develop a shared understanding of CPS and smart cities including their foundational concepts and unique dimensions such as common language, taxonomy, architectural principles. These frameworks are useful for exchanging ideas, integrating research across domains, and to develop new IoT applications with BIM. However, these frameworks do not fully tackle the above-mentioned issues [166–168]. Yet, hardly any approach is available for this industry that i) provides a comprehensive overview of data sets that need to be handled in the AEC industry; ii) evaluates the effectiveness of different methods that query and represent cross-domain data sets. Semantic web technology is claimed to be a solution, which integrates concepts from knowledge representation and reasoning (KRR). KRR aims to represent data in a form that a computer system can utilize and solve problems through finding logic, rules, and relationships. However, the performance of the query, knowledge representation, and reasoning in dealing with these data cannot be evaluated [169]; iii) globally manages collected data, processes information, and accumulates knowledge; iv) assures seamless information flow across different domains throughout lifecycle [165].

Without standards for information integration and management, it is costly and time consuming to sort large and heterogeneous data sets into usable order [57]. As a result, poorly designed and implemented information integration and management system can hinder the future development of IoT and BIM-enabled smart environments. As a fundamental step for future data integration related research, a potential question will be how information integration and management process can be standardized to facilitate effective data flow for various purposes, industry, time phase, IoT applications, and future technologies?

8.3.3 Interoperability: IoT Devices-BIM-Smart Cities

Although there is plenty of on-going research on solving interoperability issues among IoT devices and BIM, the interoperability issues among the IoT paradigm and the AEC industry still remain[45].

Firstly, there are diverse data schemas for devices, buildings, and cities. On the IoT device level, there are many data communication protocols such as BACnet, OPC, LonWorks, EIB/KNX, and MODBUS that play key roles for information exchange between different sensors and subsystems in BAS. While IFC is the most commonly known data exchange schema for BIM, CityGML dominates the city-level data interoperability. Some effort has been made in data schemas mapping. BuildingSmart has partially mapped BACnet and OPC objects with IFC Entities in IFC 2x4 RC1. In terms of interoperability between CityGML and IFC, research work like [170,171] has mapped some CityGML objects with IFC entities. However, i) only part of the communication protocol's objects has been mapped to IFC entities; ii) neither object's attributes or services have been mapped; iii) communication protocols are updating, continuous data mapping is necessary; iv) no device-level data has been mapped to city level; v) data mapping between all these schemas and protocols are heavy, various application requires distinguished data, standard data models views for different applications need to be generated to achieve efficient data exchange.; vi) Current CityGML and IFC integration is not sufficient to represent the entire built environment lifecycle [172], so that some IoT applications cannot be realized. Hence, a potential research question can be how different data models, schema, standards and protocols like IoT device protocols, open BIM standards, and city-scale data models can be integrated to solve interoperability issues for smart devices, smart buildings, and smart cities.

Furthermore, the AEC industry is part of the IoT-enabled smart city system. Information is more valuable when exchanged across systems in different domains inside the complete smart city ecosystem on the Internet [173]. The possibility to get access to the information from all these schemas and protocols through web-service (mentioned in Section 5.1) will arouse great interest. The Open Geospatial Consortium (OCG) and the

World Wide Web Consortium (W3C) developed some interoperability interfaces and metadata encodings to integrate heterogeneous sensor webs into the Internet. For example, the OGC's Sensor Web Enablement (SWE) including Sensor Model Language (SensorML), Observations and Measurements (O&M), Sensor Observation Service (SOS), Transducer Model Language (TML) and the W3C's Semantic Sensor Network ontology [46]. How to map between different data schemas and enable cloud-based smart environment is a key step for future integration of BIM and IoT. Potential research directions can focus on current limitations, including: i) differences in vocabulary, context, and semantic meaning in various domains; ii) differences in the data structure like data attributes and data formats among these schemas, IoT devices' communication protocols, and web-services' protocols.

8.3.4 Cloud Computing

The concept of IoT is not just related to IoT devices like sensors and actuators, the key concern is the interconnection of sensing and actuating devices providing information sharing through the internet. Cloud computing involves hosting computing services over the Internet and enables connecting different IoT devices to existing Internet infrastructure [156]. Cloud computing has been widely adopted in the AEC industry as it supports some BIM tools applications and storage. Most of the current sensors and BIM integration research are not yet connected with the cloud, hence further exploration in cloud computing for IoT and BIM integration is essential. With IoT devices integration with BIM, some potential problems are worth exploring.

<u>Enable Real-time Big Data Analytics:</u> Recently, people start focusing on big data techniques in the construction industry. These techniques such as statistics, data mining,

and warehousing, machine learning infused into the context of the construction industry [174]. Based on BIM and IoT devices collected data, big data techniques can be applied to automated decision making that enables intelligent monitoring and actuation. Current research is focusing on how the massive AEC data can leverage artificial intelligence algorithms and big data techniques in potential areas such as generating the optimal solution based on sensor data [64,175,176], assisting real-time operation[16,160] and problem identification [32,177]. Future research should also focus on real-time big data analytics and cloud-based big data management solutions for extensive real-time data from IoT devices and AEC data resides in BIM.

To enable real-time big data analytics, information acquired needs to be stored in online storage which can be accessed from multiple IoT devices. Design patterns such as Message-Based Cloud Update and On-Demand Cloud Update for SOA can effectively solve BIM and multiple IoT device information cloud storage and information querying issues [157]. Web services and cloud services need to be combined to perform effective managing and processing of data.

In the IoT cloud paradigm, there is no perfect big data management solution for the cloud [167]. One important factor which hinders the quality of service, security, and privacy is that data integrity is not guaranteed. Combining BIM data with sensor big data will exacerbate this issue. Some research work listed in [178] tried to propose solutions to collect and managing sensors data in the smart building in the IoT environment, but the solutions are still in infancy.

<u>Create Standards for the BIM-IoT cloud:</u> the lack of standards is considered to be

an open issue for IoT and Cloud paradigm. Although some research tried to standardize IoT and Cloud paradigms, there is no clear standard protocols, architecture, and APIs that interconnect various IoT devices and services in the Cloud [167]. With BIM data infused into IoT and Cloud paradigm, the problem further extends to the AEC industry. A general standard must be established to connect hardware, BIM data, communication protocols, ontologies, semantic rules, middleware, and applications [168] as a future research direction.

BIM and IoT data Storage: Cloud-based storage solutions became increasingly popular since 2012 [1]. The massive data amount coming from IoT deceives and BIMs will arouse problems in cloud-based data storage. Future research should be conducted to tackle existing issues, as follows: i) current commercial BIM clouds like A360 and proposed cloud-based BIM systems [166] are focused on data sharing, collaboration, analyzation, and visualization of BIMs. However, these solutions do not involve sensor data storage, future research is needed to explore how to link sensor data with BIMs in the cloud; ii) what are the optimal solutions to store both sensor data and BIM data in an appropriate format (without heavy data conversion) with current cloud-based data storage like NoSQL database? The open storage solution should preserve integrated information which can be shared with other industries [164]; iii) research is needed to investigate how to transfer data from IoT devices to the server-side with the timestamp to enable reconstruction and processing that does not arouse the problem of transferring timing [167]; iv) how to update continuous scene of IoT enabled BIM models stored in the cloud? The game industry solves scenes updating through patches and downloadable content (DLC). A similar patching procedure is necessary to continue updating BIM-IoT models when new buildings or devices are connecting [179].

Using Cloud-Based IoT Integration Portal for BIM: there were a few studies that used IoT clouds/platforms/services like Eclipse IoT, Xively, ThingSpeak and other technology listed in [157] to integrate BIMs. These integration portals enable machine-to-machine integration, visualization of sensor data, user interaction with actuators, service development, and web resource update based on sensor data. These cloud-based integration portals together with cloud storage will facilitate BIM application integration in more convenient ways. Future studies can explore how to utilize these cloud-based IoT integration portal with BIM for different applications.

Create General Integration Methodology: even though several applications were built around IoT, BIM, and Cloud [26,52,62,177], little effort has been made to produce a universal methodology that integrates IoT, BIM, and Cloud systems [167]. A common workflow, generic architecture, or platform will be beneficial for building applications that share common requirements and characteristics for the future.

Other Issues: some other issues, which have been mentioned in the computing domain, can be extended to BIM and IoT integration in the cloud and can be the subject of future studies: i). security and privacy [24]; ii) pricing and billing; iii) network and communication; iv) scalability and flexibility; v) IoT devices performance management [167].

APPENDIX A. AN EXAMPLE OF BUILDING CONTEXTUAL RAW DATA


```
ISO-10303-21;
HEADER;

/*******************************************************************
******************
* STEP Physical File produced by: The EXPRESS Data Manager Version 5.02.0100.07 :
28 Aug 2013
* Module:                    EDMstepFileFactory/EDMstandAlone
* Creation date:             Mon Mar 02 12:08:24 2020
* Host:                      COA-HIN-228-01
* Database:                  C:\Users\stang93\AppData\Local\Temp\{4AE5EA17-0D2E-
4E02-AA14-B46C4D77CC13}\ifc
* Database version:          5507
* Database creation date:    Mon Mar 02 12:08:13 2020
* Schema:                    IFC4
* Model:                     DataRepository.ifc
* Model creation date:       Mon Mar 02 12:08:14 2020
* Header model:              DataRepository.ifc_HeaderModel
* Header model creation date:    Mon Mar 02 12:08:14 2020
* EDMuser:                   sdai-user
* EDMgroup:                  sdai-group
* License ID and type:       5605 : Permanent license. Expiry date:
* EDMstepFileFactory options:    020000
********************************************************************
*****************/
FILE_DESCRIPTION(('ViewDefinition [DesignTransferView_V1.0]'),'2;1');
FILE_NAME('Project Number','2020-03-02T12:08:24',(''),(''),'The EXPRESS Data
Manager Version 5.02.0100.07 : 28 Aug 2013','20170630_0700(x64) - Exporter 18.1.0.92
- Alternate UI 18.1.0.92','');
FILE_SCHEMA(('IFC4'));
ENDSEC;

DATA;
#1= IFCORGANIZATION($,'Autodesk Revit 2018 (ENU)',$,$,$);
#5= IFCAPPLICATION(#1,'2018','Autodesk Revit 2018 (ENU)','Revit');
#6= IFCCARTESIANPOINT((0.,0.,0.));
#10= IFCCARTESIANPOINT((0.,0.));
#12= IFCDIRECTION((1.,0.,0.));
#14= IFCDIRECTION((-1.,0.,0.));
#16= IFCDIRECTION((0.,1.,0.));
```

#18= IFCDIRECTION((0.,-1.,0.));
#20= IFCDIRECTION((0.,0.,1.));
#22= IFCDIRECTION((0.,0.,-1.));
#24= IFCDIRECTION((1.,0.));
#26= IFCDIRECTION((-1.,0.));
#28= IFCDIRECTION((0.,1.));
#30= IFCDIRECTION((0.,-1.));
#32= IFCAXIS2PLACEMENT3D(#6,$,$);
#33= IFCLOCALPLACEMENT(#21540,#32);
#36= IFCPERSON($,",'stang93',$,$,$,$,$);
#38= IFCORGANIZATION($,",",$,$);
#39= IFCPERSONANDORGANIZATION(#36,#38,$);
#42= IFCOWNERHISTORY(#39,#5,$,.NOCHANGE.,$,$,$,1583166494);
#43= IFCSIUNIT(*,.LENGTHUNIT.,$,.METRE.);
#44= IFCDIMENSIONALEXPONENTS(1,0,0,0,0,0,0);
#45= IFCMEASUREWITHUNIT(IFCRATIOMEASURE(0.3048),#43);
#46= IFCCONVERSIONBASEDUNIT(#44,.LENGTHUNIT.,'FOOT',#45);
#48= IFCSIUNIT(*,.AREAUNIT.,$,.SQUARE_METRE.);
#49= IFCDIMENSIONALEXPONENTS(2,0,0,0,0,0,0);
#50= IFCMEASUREWITHUNIT(IFCRATIOMEASURE(0.09290304),#48);
#51= IFCCONVERSIONBASEDUNIT(#49,.AREAUNIT.,'SQUARE FOOT',#50);
#52= IFCSIUNIT(*,.VOLUMEUNIT.,$,.CUBIC_METRE.);
#53= IFCDIMENSIONALEXPONENTS(3,0,0,0,0,0,0);
#54= IFCMEASUREWITHUNIT(IFCRATIOMEASURE(0.028316846592),#52);
#55= IFCCONVERSIONBASEDUNIT(#53,.VOLUMEUNIT.,'CUBIC FOOT',#54);
#56= IFCSIUNIT(*,.PLANEANGLEUNIT.,$,.RADIAN.);
#57= IFCDIMENSIONALEXPONENTS(0,0,0,0,0,0,0);
#58= IFCMEASUREWITHUNIT(IFCRATIOMEASURE(0.0174532925199433),#56);
#59= IFCCONVERSIONBASEDUNIT(#57,.PLANEANGLEUNIT.,'DEGREE',#58);
#60= IFCSIUNIT(*,.MASSUNIT.,.KILO.,.GRAM.);
#61= IFCDERIVEDUNITELEMENT(#60,1);
#62= IFCDERIVEDUNITELEMENT(#43,-3);
#63= IFCDERIVEDUNIT((#61,#62),.MASSDENSITYUNIT.,$);
#65= IFCSIUNIT(*,.TIMEUNIT.,$,.SECOND.);
#66= IFCSIUNIT(*,.FREQUENCYUNIT.,$,.HERTZ.);
#67= IFCSIUNIT(*,.THERMODYNAMICTEMPERATUREUNIT.,$,.KELVIN.);
#68=
IFCSIUNIT(*,.THERMODYNAMICTEMPERATUREUNIT.,$,.DEGREE_CELSIUS.);
#69= IFCDERIVEDUNITELEMENT(#60,1);
#70= IFCDERIVEDUNITELEMENT(#67,-1);
#71= IFCDERIVEDUNITELEMENT(#65,-3);
#72= IFCDERIVEDUNIT((#69,#70,#71),.THERMALTRANSMITTANCEUNIT.,$);
#74= IFCDERIVEDUNITELEMENT(#43,3);
#75= IFCDERIVEDUNITELEMENT(#65,-1);
#76= IFCDERIVEDUNIT((#74,#75),.VOLUMETRICFLOWRATEUNIT.,$);
#78= IFCSIUNIT(*,.ELECTRICCURRENTUNIT.,$,.AMPERE.);

```
#79= IFCSIUNIT(*,.ELECTRICVOLTAGEUNIT.,$,.VOLT.);
#80= IFCSIUNIT(*,.POWERUNIT.,$,.WATT.);
#81= IFCSIUNIT(*,.FORCEUNIT.,$,.NEWTON.);
#82= IFCSIUNIT(*,.ILLUMINANCEUNIT.,$,.LUX.);
#83= IFCSIUNIT(*,.LUMINOUSFLUXUNIT.,$,.LUMEN.);
#84= IFCSIUNIT(*,.LUMINOUSINTENSITYUNIT.,$,.CANDELA.);
#85= IFCDERIVEDUNITELEMENT(#60,-1);
#86= IFCDERIVEDUNITELEMENT(#43,-2);
#87= IFCDERIVEDUNITELEMENT(#65,3);
#88= IFCDERIVEDUNITELEMENT(#83,1);
#89= IFCDERIVEDUNIT((#85,#86,#87,#88),.USERDEFINED.,'Luminous Efficacy');
#91= IFCDERIVEDUNITELEMENT(#43,1);
#92= IFCDERIVEDUNITELEMENT(#65,-1);
#93= IFCDERIVEDUNIT((#91,#92),.LINEARVELOCITYUNIT.,$);
#95= IFCSIUNIT(*,.PRESSUREUNIT.,$,.PASCAL.);
#96= IFCDERIVEDUNITELEMENT(#43,-2);
#97= IFCDERIVEDUNITELEMENT(#60,1);
#98= IFCDERIVEDUNITELEMENT(#65,-2);
#99= IFCDERIVEDUNIT((#96,#97,#98),.USERDEFINED.,'Friction Loss');
#101=
IFCUNITASSIGNMENT((#46,#51,#55,#59,#60,#63,#65,#66,#68,#72,#76,#78,#79,#80,#
81,#82,#83,#84,#89,#93,#95,#99));
#103= IFCAXIS2PLACEMENT3D(#6,$,$);
```

APPENDIX B. CONTEXTUAL DATA SERIALIZED IN *TTL


```
@prefix geo:   <http://www.opengis.net/ont/geosparql#> .
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix bot:   <https://w3id.org/bot#> .
@prefix inst:  <https://dbl.gatech.edu/project/BIM_BAS_IoT_Contextual#> .
@prefix xsd:   <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9c2
        a       bot:Space .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a992
        a       bot:Space .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d5
        a       bot:Space .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a98b
        a       bot:Space .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9ce
        a       bot:Space .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a99e
        a       bot:Space .

inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14ea4ae
        a       bot:Element .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9db
        a       bot:Space .

inst:slab_75e7fcf3-a2d1-49e7-b294-83172df02e01
        a       bot:Element .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a997
        a       bot:Space .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9ba
        a       bot:Space .

inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14eabd1
        a       bot:Element .
```

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a991
a bot:Space .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d4
a bot:Space .

inst:slab_c67de89d-75f7-4c4d-8ce2-f71834149ba6
a bot:Element .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9cd
a bot:Space .

inst:building_66ee715c-53f7-4aba-8a63-84d490acf36d
a bot:Building ;
bot:hasStorey inst:storey_66ee715c-53f7-4aba-8a63-84d46f530dff , inst:storey_66ee715c-53f7-4aba-8a63-84d46f530676 , inst:storey_66ee715c-53f7-4aba-8a63-84d46f532ef4 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a99d
a bot:Space .

inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14ebd85
a bot:Element .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a989
a bot:Space ;
bot:containsElement inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14ea4ae , inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14e448b , inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14eb532 , inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14ebd85 , inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14ea48d , inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14ebda5 , inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14eb558 , inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14ea390 , inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14eabd1 , inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14eac48 , inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14ec853 .

inst:slab_c397d501-a86e-4e62-9b28-6c3b0a6dd871
a bot:Element .

inst:storey_66ee715c-53f7-4aba-8a63-84d46f530dff
a bot:Storey ;
bot:containsElement inst:slab_c397d501-a86e-4e62-9b28-6c3b0a6dd871 , inst:stair_c397d501-a86e-4e62-9b28-6c3b0a6dd490 ;

bot:hasSpace inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a990 , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a994 , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a98d , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a993 , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a997 , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a99c , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9e5 , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a992 , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a98b , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a99f , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a999 , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a989 , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9ba , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a991 , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a995 , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a99a , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a99e , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a98e , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a998 , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a988 , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a99d .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9da
a bot:Space .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d9
a bot:Space .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9c0
a bot:Space .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a990
a bot:Space .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d3
a bot:Space .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9cc
a bot:Space .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9c1
a bot:Space .

inst:storey_66ee715c-53f7-4aba-8a63-84d46f530676
a bot:Storey ;
bot:containsElement inst:slab_c67de89d-75f7-4c4d-8ce2-f71834149ba6 ;
bot:hasSpace inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a926 , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9de , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d7 , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d1 , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d4 , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a925 , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9ca , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9cd , inst:space_93d5acf4-73a6-4f03-

a072-3fc1eab1a9c0 , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9dd , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9c3 , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d3 , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d8 , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d0 , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d6 , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9c9 , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a924 , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9c2 , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9dc , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9df , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9cc , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9c5 , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d2 , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d9 , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d5 , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9cb , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9c8 , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9db , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9da , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9ce , inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9c1 .

```
inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a99c
        a       bot:Space .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9df
        a       bot:Space .

inst:site_66ee715c-53f7-4aba-8a63-84d490acf36e
        a               bot:Site ;
        bot:hasBuilding  inst:building_66ee715c-53f7-4aba-8a63-84d490acf36d .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a988
        a       bot:Space .

inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14ec853
        a       bot:Element .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9c5
        a       bot:Space .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a926
        a       bot:Space .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a995
        a       bot:Space .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d8
        a       bot:Space .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a98e
        a       bot:Space .
```

```
inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14ea390
        a       bot:Element .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d2
        a       bot:Space .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9cb
        a       bot:Space .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9e5
        a       bot:Space .

inst:stair_c397d501-a86e-4e62-9b28-6c3b0a6dd490
        a       bot:Element .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9de
        a       bot:Space .

inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14e448b
        a       bot:Element .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a925
        a       bot:Space .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a994
        a       bot:Space .

inst:space_bccb3261-733c-482c-8b55-56175a058e2b
        a       bot:Space .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d7
        a       bot:Space .

inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14eac48
        a       bot:Element .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a98d
        a       bot:Space .

inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14ebda5
        a       bot:Element .

inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14eb532
        a       bot:Element .
```

```
inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d1
        a       bot:Space .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9ca
        a       bot:Space .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a99a
        a       bot:Space .

inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14eb558
        a       bot:Element .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9dd
        a       bot:Space .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9c9
        a       bot:Space .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a999
        a       bot:Space .

inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14ea48d
        a       bot:Element .

inst:storey_66ee715c-53f7-4aba-8a63-84d46f532ef4
        a                    bot:Storey ;
        bot:containsElement  inst:slab_75e7fcf3-a2d1-49e7-b294-83172df02e01 ;
        bot:hasSpace         inst:space_bccb3261-733c-482c-8b55-56175a058e2b .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9c3
        a       bot:Space .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a924
        a       bot:Space .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a993
        a       bot:Space .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d6
        a       bot:Space .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a99f
        a       bot:Space .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d0
        a       bot:Space .
```

```
inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9dc
        a       bot:Space .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9c8
        a       bot:Space .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a998
        a       bot:Space .
```

APPENDIX C. AN EXAMPLE OF BAS RAW METADATA

ROOM	Object Name	Object Type	Object Type Instance Number	Authorization Category
ERU	ControlTemp	AI	14	Area-3
ERU	SpaceRH	AI	11	Area-3
ERU	RATemp	AI	2	Area-3
ERU	OutdoorTemp	AI	4	Area-3
ERU	DischAirTemp	AI	1	Area-3
ERU	FanSpd	AI	8	Area-3
ERU	ExhFanValue	AI	10	Area-3
ERU	ERWheelSpd	AI	15	Area-3
ERU	EFT_LCT	AI	7	Area-3
ERU	EREAT	AI	17	Area-3
ERU	ERLAT	AI	16	Area-3
ERU	RefSuctionP	AI	25	Area-3
ERU	SucnRefTemp	AI	24	Area-3
ERU	RefDischP	AI	26	Area-3
ERU	DischLn3Temp	AI	21	Area-3
ERU	DischLn1Temp	AI	22	Area-3
ERU	OFReqSBEvnt	BI	87	Area-3
ERU	INVCmpTSBEvnt	BI	86	Area-3
ERU	ClgDPSBEvnt	BI	83	Area-3
ERU	ClgLPSBEvnt	BI	80	Area-3
ERU	DirtyFilterSw	BI	3	Area-3
ERU	HiPress1Sw	BI	4	Area-3

ERU	AirFlwSw	BI	1	Area-3
ERU	EmergencyOffSw	BI	22	Area-3
ERU	Ret/ExhFanWrn	BI	42	Area-3
ERU	LoPress1Prb	BI	33	Area-3
ERU	LoPressDiffPrb	BI	39	Area-3
ERU	HiDLTempPrb	BI	35	Area-3
ERU	LoSuperHtWrn	BI	46	Area-3
ERU	LoChargePrb	BI	37	Area-3
ERU	ChargeLossPrb	BI	41	Area-3
ERU	HiPress1Prb	BI	32	Area-3
ERU	ExpValvePrb	BI	40	Area-3
ERU	OAFanPrb	BI	36	Area-3
ERU	INVCompPrb	BI	34	Area-3
ERU	HiDLTUnldEvnt	BI	55	Area-3
ERU	ClgHPUnldEvnt	BI	52	Area-3
ERU	CmpDsblSBEvnt	BI	73	Area-3
ERU	INVReqSBEvnt	BI	65	Area-3
ERU	ClgHPSBEvnt	BI	78	Area-3
ERU	INVUnldReqEvnt	BI	72	Area-3
ERU	INVAmpUnldEvnt	BI	75	Area-3
ERU	INVFinTUnldEvnt	BI	71	Area-3
ERU	Comp3DLTULEvnt	BI	74	Area-3
ERU	CmpRatioULEvnt	BI	70	Area-3
ERU	ClgDPUnldEvnt	BI	76	Area-3
ERU	ClgLPUnldEvnt	BI	67	Area-3
ERU	EconCapacity	AV	15	Area-3
ERU	HtgCapacity	AV	2	Area-3
ERU	ClgCapacity	AV	1	Area-3

ERU	DAClgSetpt	AV	13	Area-3
ERU	DefaultDATClgSetpt	AV	14	Area-3
ERU	OccHeatSP	AV	11	Area-3
ERU	OccCoolSP	AV	9	Area-3
ERU	UnoccHeatSetpt	AV	12	Area-3
ERU	UnoccCoolSetpt	AV	10	Area-3
ERU	LocalSpaceTmp	AV	4	Area-3
ERU	LocalOATemp	AV	5	Area-3
ERU	TimeToNextState	AV	3	Area-3
ERU	SpaceDewPt	AV	20	Area-3
ERU	DAHtgSetpt	AV	17	Area-3
ERU	DefaultDATHtgSetpt	AV	18	Area-3
ERU	SpaceTempInput	AV	28	Area-3
ERU	SpaceRHNetln	AV	19	Area-3
ERU	OutdoorTempInput	AV	29	Area-3
ERU	AlarmValue	AV	27	Area-3
ERU	ActiveWarning	AV	24	Area-3
ERU	ActiveProblem	AV	25	Area-3
ERU	ActiveFault	AV	26	Area-3
ERU	ReheatCapacity	AV	44	Area-3
ERU	DewpointSP	AV	41	Area-3
ERU	HumiditySP	AV	40	Area-3
ERU	EffDATempSP	AV	39	Area-3
ERU	ReceiveHeartbeat	AV	43	Area-3
ERU	HeatEnablePct	AV	37	Area-3
ERU	HeatEnable	AV	32	Area-3
ERU	CoolEnablePct	AV	35	Area-3
ERU	CoolEnable	AV	34	Area-3

ERU	DACMWUSpt	AV	55	Area-3
ERU	MinDehumLCTSpt	AV	56	Area-3
ERU	Comp3Hrs	AV	107	Area-3
ERU	SupplyFanHrs	AV	100	Area-3
ERU	RF_EFHrs	AV	101	Area-3
ERU	CmpClgHrs	AV	113	Area-3
ERU	TenantORHrs	AV	118	Area-3
ERU	DehumHrs	AV	119	Area-3
ERU	HeatingHrs	AV	116	Area-3
ERU	ERWhlHrs	AV	120	Area-3
ERU	EffectOccup	MSV	6	Area-3
ERU	HtgStatus	MSV	4	Area-3
ERU	EconoStatus	MSV	3	Area-3
ERU	ClgStatus	MSV	2	Area-3
ERU	UnitState	MSV	15	Area-3
ERU	ClearAlarms	MSV	13	Area-3
ERU	NextState	MSV	9	Area-3
ERU	EmergOverride	MSV	10	Area-3
ERU	OccManCmd	MSV	7	Area-3
ERU	ApplicCmd	MSV	5	Area-3
ERU	CurrentState	MSV	8	Area-3
ERU	ExhRetFanCtrl	MSV	12	Area-3
ERU	UnitSupport	MSV	16	Area-3
ERU	ERWhlOnOff	MSV	37	Area-3
ERU	DehumStatus	MSV	46	Area-3
ERU	CtlrTempSrc	MSV	39	Area-3
ERU	AHULoc/Net	MSV	38	Area-3
ERU	MWUStatus	MSV	49	Area-3

Room110	**B60A.NAE227.Bacnet1.Room110.FC-1-1.ZoneTemp**	AI	9	Area-3
Room110	**B60A.NAE227.Bacnet1.Room110.FC-1-1.ZoneTempSetPoint**	AV	10	Area-3
Room110	**B60A.NAE227.Bacnet1.Room110.FC-1-1.UnitCommand**	BO	1	Area-3
Room110	**B60A.NAE227.Bacnet1.Room110.FC-1-1.UnitStatus**	BI	2	Area-3
Room110	**B60A.NAE227.Bacnet1.Room110.FC-1-1.OperationMode**	MSI	6	Area-3
Room110	**B60A.NAE227.Bacnet1.Room110.FC-1-1.FanSpeed**	MSO	7	Area-3
Room110	**B60A.NAE227.Bacnet1.Room110.FC-1-1.IndoorFanStatus**	BI	30	Area-3
Room110	**B60A.NAE227.Bacnet1.Room110.FC-1-1.CompressorStatus**	BI	29	Area-3
Room110	**B60A.NAE227.Bacnet1.Room110.FC-1-1.Alarm**	BI	3	Area-3
Room110	**B60A.NAE227.Bacnet1.Room110.FC-1-1.MultiFunctionCode**	MSO	5	Area-3
Room110	**B60A.NAE227.Bacnet1.Room110.FC-1-1.MultiFunctionCode**	MSI	4	Area-3
Room112	B60A.NAE227.Bacnet1.Room112.FC-1-2.ZoneTemp	AI	265	Area-3
Room112	B60A.NAE227.Bacnet1.Room112.FC-1-2.ZoneTempSetPoint	AV	266	Area-3
Room112	B60A.NAE227.Bacnet1.Room112.FC-1-2.UnitCommand	BO	7169	Area-3
Room112	B60A.NAE227.Bacnet1.Room112.FC-1-2.UnitStatus	BI	258	Area-3
Room112	B60A.NAE227.Bacnet1.Room112.FC-1-2.OperationMode	MSI	262	Area-3
Room112	B60A.NAE227.Bacnet1.Room112.FC-1-2.FanSpeed	MSO	263	Area-3
Room112	B60A.NAE227.Bacnet1.Room112.FC-1-2.IndoorFanStatus	BI	286	Area-3
Room112	B60A.NAE227.Bacnet1.Room112.FC-1-2.CompressorStatus	BI	285	Area-3
Room112	B60A.NAE227.Bacnet1.Room112.FC-1-2.Alarm	BI	259	Area-3
Room112	B60A.NAE227.Bacnet1.Room112.FC-1-2.MultiFunctionCode	MSI	260	Area-3
Room112	B60A.NAE227.Bacnet1.Room112.FC-1-2.MultiFunctionCode	MSo	261	Area-3
Room114	B60A.NAE227.Bacnet1.Room114.FC-1-3.ZoneTemp	AI	521	Area-3
Room114	B60A.NAE227.Bacnet1.Room114.FC-1-3.ZoneTempSetPoint	AV	522	Area-3
Room114	B60A.NAE227.Bacnet1.Room114.FC-1-3.UnitCommand	BO	257	Area-3
Room114	B60A.NAE227.Bacnet1.Room114.FC-1-3.UnitStatus	BI	514	Area-3
Room114	B60A.NAE227.Bacnet1.Room114.FC-1-3.OperationMode	MSI	518	Area-3
Room114	B60A.NAE227.Bacnet1.Room114.FC-1-3.FanSpeed	MSO	519	Area-3

Room114	B60A.NAE227.Bacnet1.Room114.FC-1-3.IndoorFanStatus	BI	542	Area-3
Room114	B60A.NAE227.Bacnet1.Room114.FC-1-3.MultiFunctionCode	MSO	517	Area-3
Room114	B60A.NAE227.Bacnet1.Room114.FC-1-3.CompressorStatus	BI	541	Area-3
Room114	B60A.NAE227.Bacnet1.Room114.FC-1-3.Alarm	BI	515	Area-3
Room114	B60A.NAE227.Bacnet1.Room114.FC-1-3.MultiFunctionCode	MSI	516	Area-3
Lobby120	B60A.NAE227.Bacnet1.RoomLobby120.FC-1-4.ZoneTemp	AI	1289	Area-3
Lobby120	B60A.NAE227.Bacnet1.RoomLobby120.FC-1-4.ZoneTempSetPoint	AV	778	Area-3
Lobby120	B60A.NAE227.Bacnet1.RoomLobby120.FC-1-4.UnitCommand	BP	513	Area-3
Lobby120	B60A.NAE227.Bacnet1.RoomLobby120.FC-1-4.UnitStatus	BI	770	Area-3
Lobby120	B60A.NAE227.Bacnet1.RoomLobby120.FC-1-4.OperationMode	MSI	774	Area-3
Lobby120	B60A.NAE227.Bacnet1.RoomLobby120.FC-1-4.FanSpeed	MSO	775	Area-3
Lobby120	B60A.NAE227.Bacnet1.RoomLobby120.FC-1-4.IndoorFanStatus	BI	798	Area-3
Lobby120	B60A.NAE227.Bacnet1.RoomLobby120.FC-1-4.CompressorStatus	BI	797	Area-3
Lobby120	B60A.NAE227.Bacnet1.RoomLobby120.FC-1-4.Alarm	BI	771	Area-3
Lobby120	B60A.NAE227.Bacnet1.RoomLobby120.FC-1-4.MultiFunctionCode	MSI	772	Area-3
Lobby120	B60A.NAE227.Bacnet1.RoomLobby120.FC-1-4.MultiFunctionCode	MSO	773	Area-3
Lobby120	B60A.NAE227.Bacnet1.RoomLobby120.FC-1-5.ZoneTemp	AI	2057	Area-3
Lobby120	B60A.NAE227.Bacnet1.RoomLobby120.FC-1-5.ZoneTempSetPoint	AV	1034	Area-3
Lobby120	B60A.NAE227.Bacnet1.RoomLobby120.FC-1-5.UnitCommand	BO	769	Area-3
Lobby120	B60A.NAE227.Bacnet1.RoomLobby120.FC-1-5.UnitStatus	BI	1026	Area-3
Lobby120	B60A.NAE227.Bacnet1.RoomLobby120.FC-1-5.OperationMode	MSI	1030	Area-3
Lobby120	B60A.NAE227.Bacnet1.RoomLobby120.FC-1-5.FanSpeed	MSP	1031	Area-3
Lobby120	B60A.NAE227.Bacnet1.RoomLobby120.FC-1-5.IndoorFanStatus	BI	1054	Area-3
Lobby120	B60A.NAE227.Bacnet1.RoomLobby120.FC-1-5.CompressorStatus	BI	1053	Area-3
Lobby120	B60A.NAE227.Bacnet1.RoomLobby120.FC-1-5.Alarm	BI	1027	Area-3
Lobby120	B60A.NAE227.Bacnet1.RoomLobby120.FC-1-5.MultiFunctionCode	MSI	1028	Area-3
Lobby120	B60A.NAE227.Bacnet1.RoomLobby120.FC-1-5.MultiFunctionCode	MSO	1029	Area-3
Lobby120	B60A.NAE227.Bacnet1.RoomLobby120.FC-1-6.ZoneTemp	AI	2313	Area-3

Lobby120	B60A.NAE227.Bacnet1.RoomLobby120.FC-1-6.ZoneTempSetPoint	AV	1290	Area-3
Lobby120	B60A.NAE227.Bacnet1.RoomLobby120.FC-1-6.UnitCommand	BO	1025	Area-3
Lobby120	B60A.NAE227.Bacnet1.RoomLobby120.FC-1-6.UnitStatus	BI	1282	Area-3
Lobby120	B60A.NAE227.Bacnet1.RoomLobby120.FC-1-6.OperationMode	MSI	1286	Area-3
Lobby120	B60A.NAE227.Bacnet1.RoomLobby120.FC-1-6.FanSpeed	MSP	1287	Area-3
Lobby120	B60A.NAE227.Bacnet1.RoomLobby120.FC-1-6.IndoorFanStatus	BI	1310	Area-3
Lobby120	B60A.NAE227.Bacnet1.RoomLobby120.FC-1-6.CompressorStatus	BI	1309	Area-3
Lobby120	B60A.NAE227.Bacnet1.RoomLobby120.FC-1-6.Alarm	BI	1283	Area-3
Lobby120	B60A.NAE227.Bacnet1.RoomLobby120.FC-1-6.MultiFunctionCode	MSI	1284	Area-3
Lobby120	B60A.NAE227.Bacnet1.RoomLobby120.FC-1-6.MultiFunctionCode	MSO	1285	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-1.ZoneTemp	AI	2569	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-1.ZoneTempSetPoint	AV	2058	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-1.UnitCommand	BO	1793	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-1.UnitStatus	BI	2050	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-1.OperationMode	MSI	2054	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-1.FanSpeed	MSP	2055	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-1.IndoorFanStatus	BI	2078	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-1.CompressorStatus	BI	2077	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-1.Alarm	BI	2051	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-1.MultiFunctionCode	MSO	2053	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-1.MultiFunctionCode	MSI	2052	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-2.ZoneTemp	AI	4876	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-2.ZoneTempSetPoint	AV	2314	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-2.UnitCommand	BO	2049	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-2.UnitStatus	BI	2306	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-2.OperationMode	MSI	2310	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-2.FanSpeed	MSP	2311	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-2.IndoorFanStatus	BI	2334	Area-3

Room122	B60A.NAE227.Bacnet1.Room122.FC-2-2.CompressorStatus	BI	2333	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-2.Alarm	BI	2307	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-2.MultiFunctionCode	MSO	2309	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-2.MultiFunctionCode	MSI	2308	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-3.ZoneTemp	AI	1545	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-3.ZoneTempSetPoint	AV	1546	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-3.UnitCommand	BO	1281	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-3.UnitStatus	BI	1538	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-3.OperationMode	MSI	1542	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-3.FanSpeed	MSP	1543	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-3.IndoorFanStatus	BI	1566	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-3.CompressorStatus	BI	1565	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-3.Alarm	BI	1539	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-3.MultiFunctionCode	MSO	1541	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-3.MultiFunctionCode	MSI	1540	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-4.ZoneTemp	AI	1801	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-4.ZoneTempSetPoint	AV	1802	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-4.UnitCommand	BO	1537	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-4.UnitStatus	BI	1794	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-4.OperationMode	MSI	1798	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-4.FanSpeed	MSP	1799	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-4.IndoorFanStatus	BI	1822	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-4.CompressorStatus	BI	1821	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-4.Alarm	BI	1795	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-4.MultiFunctionCode	MSO	1797	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-4.MultiFunctionCode	MSI	1796	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-5.ZoneTemp	AI	2825	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-5.ZoneTempSetPoint	AV	2570	Area-3

Room122	B60A.NAE227.Bacnet1.Room122.FC-2-5.UnitCommand	BO	2305	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-5.UnitStatus	BI	2562	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-5.OperationMode	MSI	2566	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-5.FanSpeed	MSP	2567	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-5.IndoorFanStatus	BI	2590	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-5.CompressorStatus	BI	2589	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-5.Alarm	BI	2563	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-5.MultiFunctionCode	MSO	2565	Area-3
Room122	B60A.NAE227.Bacnet1.Room122.FC-2-5.MultiFunctionCode	MSI	2564	Area-3
Room203	B60A.NAE227.Bacnet1.Room203.FC-4-3.ZoneTemp	AI	6921	Area-3
Room203	B60A.NAE227.Bacnet1.Room203.FC-4-3.ZoneTempSetPoint	AV	4874	Area-3
Room203	B60A.NAE227.Bacnet1.Room203.FC-4-3.UnitCommand	BO	4609	Area-3
Room203	B60A.NAE227.Bacnet1.Room203.FC-4-3.UnitStatus	BI	4866	Area-3
Room203	B60A.NAE227.Bacnet1.Room203.FC-4-3.OperationMode	MSI	4870	Area-3
Room203	B60A.NAE227.Bacnet1.Room203.FC-4-3.FanSpeed	MSP	4871	Area-3
Room203	B60A.NAE227.Bacnet1.Room203.FC-4-3.IndoorFanStatus	BI	4894	Area-3
Room203	B60A.NAE227.Bacnet1.Room203.FC-4-3.CompressorStatus	BI	4893	Area-3
Room203	B60A.NAE227.Bacnet1.Room203.FC-4-3.Alarm	BI	4867	Area-3
Room203	B60A.NAE227.Bacnet1.Room203.FC-4-3.MultiFunctionCode	MSO	4869	Area-3
Room203	B60A.NAE227.Bacnet1.Room203.FC-4-3.MultiFunctionCode	MSI	4868	Area-3
Room208	B60A.NAE227.Bacnet1.Room208.FC-3-1.ZoneTemp	AI	3081	Area-3
Room208	B60A.NAE227.Bacnet1.Room208.FC-3-1.ZoneTempSetPoint	AV	2826	Area-3
Room208	B60A.NAE227.Bacnet1.Room208.FC-3-1.UnitCommand	BO	2561	Area-3
Room208	B60A.NAE227.Bacnet1.Room208.FC-3-1.UnitStatus	BI	2818	Area-3
Room208	B60A.NAE227.Bacnet1.Room208.FC-3-1.OperationMode	MSI	2822	Area-3
Room208	B60A.NAE227.Bacnet1.Room208.FC-3-1.FanSpeed	MSP	2823	Area-3
Room208	B60A.NAE227.Bacnet1.Room208.FC-3-1.IndoorFanStatus	BI	2846	Area-3
Room208	B60A.NAE227.Bacnet1.Room208.FC-3-1.CompressorStatus	BI	2845	Area-3

Room208	B60A.NAE227.Bacnet1.Room208.FC-3-1.Alarm	BI	2819	Area-3
Room208	B60A.NAE227.Bacnet1.Room208.FC-3-1.MultiFunctionCode	MSO	2821	Area-3
Room208	B60A.NAE227.Bacnet1.Room208.FC-3-1.MultiFunctionCode	MSI	2820	Area-3
Room210	B60A.NAE227.Bacnet1.Room210.FC-3-2.ZoneTemp	AI	4361	Area-3
Room210	B60A.NAE227.Bacnet1.Room210.FC-3-2.ZoneTempSetPoint	AV	3082	Area-3
Room210	B60A.NAE227.Bacnet1.Room210.FC-3-2.UnitCommand	BO	2817	Area-3
Room210	B60A.NAE227.Bacnet1.Room210.FC-3-2.UnitStatus	BI	3074	Area-3
Room210	B60A.NAE227.Bacnet1.Room210.FC-3-2.OperationMode	MSI	3078	Area-3
Room210	B60A.NAE227.Bacnet1.Room210.FC-3-2.FanSpeed	MSP	3079	Area-3
Room210	B60A.NAE227.Bacnet1.Room210.FC-3-2.IndoorFanStatus	BI	3102	Area-3
Room210	B60A.NAE227.Bacnet1.Room210.FC-3-2.CompressorStatus	BI	3101	Area-3
Room210	B60A.NAE227.Bacnet1.Room210.FC-3-2.Alarm	BI	3075	Area-3
Room210	B60A.NAE227.Bacnet1.Room210.FC-3-2.MultiFunctionCode	MSO	3077	Area-3
Room210	B60A.NAE227.Bacnet1.Room210.FC-3-2.MultiFunctionCode	MSI	3076	Area-3
Room212	B60A.NAE227.Bacnet1.Room212.FC-4-1.ZoneTemp	AI	6665	Area-3
Room212	B60A.NAE227.Bacnet1.Room212.FC-4-1.ZoneTempSetPoint	AV	4362	Area-3
Room212	B60A.NAE227.Bacnet1.Room212.FC-4-1.UnitCommand	BO	4097	Area-3
Room212	B60A.NAE227.Bacnet1.Room212.FC-4-1.UnitStatus	BI	4354	Area-3
Room212	B60A.NAE227.Bacnet1.Room212.FC-4-1.OperationMode	MSI	4358	Area-3
Room212	B60A.NAE227.Bacnet1.Room212.FC-4-1.FanSpeed	MSP	4359	Area-3
Room212	B60A.NAE227.Bacnet1.Room212.FC-4-1.IndoorFanStatus	BI	4382	Area-3
Room212	B60A.NAE227.Bacnet1.Room212.FC-4-1.CompressorStatus	BI	4381	Area-3
Room212	B60A.NAE227.Bacnet1.Room212.FC-4-1.Alarm	BI	4355	Area-3
Room212	B60A.NAE227.Bacnet1.Room212.FC-4-1.MultiFunctionCode	MSO	4357	Area-3
Room212	B60A.NAE227.Bacnet1.Room212.FC-4-1.MultiFunctionCode	MSI	4356	Area-3
Room214	B60A.NAE227.Bacnet1.Room214.FC-4-2.ZoneTemp	AI	5129	Area-3
Room214	B60A.NAE227.Bacnet1.Room214.FC-4-2.ZoneTempSetPoint	AV	4618	Area-3
Room214	B60A.NAE227.Bacnet1.Room214.FC-4-2.UnitCommand	BO	4353	Area-3

Room214	B60A.NAE227.Bacnet1.Room214.FC-4-2.UnitStatus	BI	4610	Area-3
Room214	B60A.NAE227.Bacnet1.Room214.FC-4-2.OperationMode	MSI	4614	Area-3
Room214	B60A.NAE227.Bacnet1.Room214.FC-4-2.FanSpeed	MSP	4615	Area-3
Room214	B60A.NAE227.Bacnet1.Room214.FC-4-2.IndoorFanStatus	BI	4638	Area-3
Room214	B60A.NAE227.Bacnet1.Room214.FC-4-2.CompressorStatus	BI	4637	Area-3
Room214	B60A.NAE227.Bacnet1.Room214.FC-4-2.Alarm	BI	4611	Area-3
Room214	B60A.NAE227.Bacnet1.Room214.FC-4-2.MultiFunctionCode	MSO	4612	Area-3
Room214	B60A.NAE227.Bacnet1.Room214.FC-4-2.MultiFunctionCode	MSI	4613	Area-3
Room218	B60A.NAE227.Bacnet1.Room218.FC-5-1.ZoneTemp	AI	3849	Area-3
Room218	B60A.NAE227.Bacnet1.Room218.FC-5-1.ZoneTempSetPoint	AV	5898	Area-3
Room218	B60A.NAE227.Bacnet1.Room218.FC-5-1.UnitCommand	BO	5633	Area-3
Room218	B60A.NAE227.Bacnet1.Room218.FC-5-1.UnitStatus	BI	5890	Area-3
Room218	B60A.NAE227.Bacnet1.Room218.FC-5-1.OperationMode	MSI	5894	Area-3
Room218	B60A.NAE227.Bacnet1.Room218.FC-5-1.FanSpeed	MSP	5895	Area-3
Room218	B60A.NAE227.Bacnet1.Room218.FC-5-1.IndoorFanStatus	BI	5918	Area-3
Room218	B60A.NAE227.Bacnet1.Room218.FC-5-1.CompressorStatus	BI	5917	Area-3
Room218	B60A.NAE227.Bacnet1.Room218.FC-5-1.Alarm	BI	5891	Area-3
Room218	B60A.NAE227.Bacnet1.Room218.FC-5-1.MultiFunctionCode	MSI	5892	Area-3
Room218	B60A.NAE227.Bacnet1.Room218.FC-5-1.MultiFunctionCode	MSO	5893	Area-3
Room220	B60A.NAE227.Bacnet1.Room220.FC-5-2.ZoneTemp	AI	4105	Area-3
Room220	B60A.NAE227.Bacnet1.Room220.FC-5-2.ZoneTempSetPoint	AV	6154	Area-3
Room220	B60A.NAE227.Bacnet1.Room220.FC-5-2.UnitCommand	BO	5889	Area-3
Room220	B60A.NAE227.Bacnet1.Room220.FC-5-2.UnitStatus	BI	6146	Area-3
Room220	B60A.NAE227.Bacnet1.Room220.FC-5-2.OperationMode	MSI	6150	Area-3
Room220	B60A.NAE227.Bacnet1.Room220.FC-5-2.FanSpeed	MSP	6151	Area-3

APPENDIX D. BAS METADATA PARSING RULES

D.1 name_point_map.json

{

"Room": "Room",

"Lobby": "Room",

"FC": "Fan_Coil_Unit",

"ZoneTemp": "Temperature_Sensor",

"ZoneTempSetPoint": "Zone_Temperature_Setpoint",

"UnitCommand": "Start_Stop_Command",

"UnitStatus": "Start_Stop_Status",

"OperationMode": "Fan_Status",

"FanSpeed": "Speed_Sensor",

"IndoorFanStatus": "Fan_Status",

"CompressorStatus": "Start_Stop_Status",

"Alarm": "Temperature_Alarm",

"MultiFunctionCode": "Mode_Command",

"B095E": "Electrical_Meter",

"B095E_U2U1": "Elevator",

"BO95E_U3U1": "HVAC",

"B095E_U4U1": "Boiler",

"B095E_U5U1": "Electrical_System",

"B095E_U6U1": "Lighting_System",

"B095E_U7U5": "Lighting_System",

"B095E_U8U1": "Electrical_System",

"B095E_U9U8": "HVAC",

"B095E_U10U8": "Hot_Water_Pump",

"B095E_U11U8": "Lighting_System",

"ControlTemp": "Temperature_Setpoint",

"SpaceRH": "Humidity_Sensor",

"RATemp": "Return_Air_Temperature_Sensor",

"OutdoorTemp": "Outside_Air_Temperature_Sensor",

"DischAirTemp": "Discharge_Air_Temperature_Sensor",

"FanSpd": "Speed_Sensor",

"ExhFanValue": "Exhaust_Fan_Fan_Air_Flow_Sensor",

"ERWheelSpd": "Speed_Sensor",

"EFT_LCT": "Temperature_Sensor",

"EREAT": "Heat_Wheel_Supply_Air_Temperature_Sensor",

"ERLAT": "Heat_Wheel_Discharge_Air_Temperature_Sensor",

"RefSuctionP": "Low_Suction_Pressure_Alarm",

"SucnRefTemp": "Temperature_Sensor",

"RefDischP": "Pressure_Sensor",

"DischLn3Temp": "Temperature_Sensor",

"DischLn1Temp": "Temperature_Sensor",

"OFReqSBEvnt": "Point",

"INVCmpTSBEvnt": "Point",

"ClgDPSBEvnt": "Point",

"ClgLPSBEvnt": "Point",

"DirtyFilterSw": "Change_Filter_Alarm",

"HiPress1Sw": "Alarm",
"AirFlwSw": "Air_Flow_Loss_Alarm",
"EmergencyOffSw": "Alarm",
"Ret/ExhFanWrn": "Alarm",
"LoPress1Prb": "Low_Suction_Pressure_Alarm",
"LoPressDiffPrb": "Alarm",
"HiDLTempPrb": "Discharge_Air_Temperature_High_Limit_Alarm",
"LoSuperHtWrn": "Alarm",
"LoChargePrb": "Alarm",
"ChargeLossPrb": "Alarm",
"HiPress1Prb": "High_Head_Pressure_Alarm",
"ExpValvePrb": "Alarm",
"OAFanPrb": "Alarm",
"INVCompPrb": "Alarm",
"HiDLTUnldEvnt": "Point",
"ClgHPUnldEvnt": "Point",
"CmpDsblSBEvnt": "Point",
"INVReqSBEvnt": "Point",
"ClgHPSBEvnt": "Point",
"INVUnldReqEvnt": "Point",
"INVAmpUnldEvnt": "Point",
"INVFinTUnldEvnt": "Point",
"Comp3DLTULEvnt": "Point",
"CmpRatioULEvnt": "Point",
"ClgDPUnldEvnt": "Point",

"ClgLPUnldEvnt": "Point",

"EconCapacity": "Setpoint",

"HtgCapacity": "Heating_Temperature_Setpoint",

"ClgCapacity": "Cooling_Temperature_Setpoint",

"DACIgSetpt": "Setpoint",

"DefaultDATClgSetpt": "Min_Discharge_Air_Temperature_Setpoint",

"OccHeatSP": "Occupied_Heating_Temperature_Setpoint",

"OccCoolSP": "Occupied_Cooling_Temperature_Setpoint",

"UnoccHeatSetpt": "Unoccupied_Heating_Temperature_Setpoint",

"UnoccCoolSetpt": "Unoccupied_Cooling_Temperature_Setpoint",

"LocalSpaceTmp": "Zone_Air_Temperature_Sensor",

"LocalOATemp": "Outside_Air_Temperature_Sensor",

"TimeToNextState": "Occupancy_Command",

"SpaceDewPt": "Humidity_Sensor",

"DAHtgSetpt": "Discharge_Air_Temperature_Heating_Setpoint",

"DefaultDATHtgSetpt": "Setpoint",

"SpaceTempInput": "Zone_Temperature_Sensor",

"SpaceRHNetln": "HVAC",

"OutdoorTempInput": "Zone_Temperature_Sensor",

"AlarmValue": "Alarm",

"ActiveWarning": "Alarm",

"ActiveProblem": "Alarm",

"ActiveFault": "Alarm",

"ReheatCapacity": "Capacity_Sensor",

"DewpointSP": "Dewpoint_Setpoint",

"HumiditySP": "Humidity_Setpoint",

"EffDATempSP": "Discharge_Air_Temperature_Setpoint",

"ReceiveHeartbeat": "Occupancy_Command",

"HeatEnablePct": "On_Off_Command",

"HeatEnable": "On_Off_Command",

"CoolEnablePct": "On_Off_Command",

"CoolEnable": "On_Off_Command",

"DACMWUSpt": "Heating_Temperature_Setpoint",

"MinDehumLCTSpt": "Discharge_Air_Temperature_Cooling_Setpoint",

"Comp3Hrs": "Time",

"SupplyFanHrs": "Time",

"RF_EFHrs": "Time",

"CmpClgHrs": "Time",

"TenantORHrs": "Time",

"DehumHrs":"Time",

"HeatingHrs":"Time",

"ERWhlHrs": "Time",

"EffectOccup": "Occupied_Mode_Status",

"HtgStatus": "Heating_On_Off_Status",

"EconoStatus": "Economizer_Status",

"ClgStatus": "Cooling_On_Off_Status",

"UnitState": "Mode_Status",

"ClearAlarms": "Alarm",

"NextState": "Command",

"EmergOverride": "Emergency_Power_Off_Command", "OccManCmd": "Occupancy_Command",

"ApplicCmd": "Command",

"CurrentState": "Occupancy_Command",

"ExhRetFanCtrl": "Mode_Command",

"UnitSupport": "On_Off_Status",

"ERWhlOnOff": "On_Off_Status",

"DehumStatus": "Dehumidification_On_Off_Status",

"CtlrTempSrc": "Mode_Status",

"AHULoc/Net": "Mode_Status",

"MWUStatus": "Mode_Status"

}

D.2 space_room_map.json

{
"Room100": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a99a"],
"Room101": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a999"],
"Room102": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a988"],
"Room103": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a990"],
"Room104": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a995"],
"Room105": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a98b"],
"Room106": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a997"],
"Room107": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a994"],
"Room108": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a98e"],
"Room109": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a98d"],
"Room110": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a989"],
"Room111": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a99d"],
"Room112": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a993"],
"Room114": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a99c"],
"Room116": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a99e"],
"Room117": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a998"],
"Lobby120": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a99f"],
"Room121": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a991"],
"Room122": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a992"],
"Room124": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a9ba"],
"Room126": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a9e5"],
"Room200": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d2"],

"Room201": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a9db"],
"Room202": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a9df"],
"Room203": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a9de"],
"Room204": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a9c5"],
"Room205": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d9"],
"Room206": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a9c2"],
"Room207": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a924"],
"Room208": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a9c9"],
"Room209": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d8"],
"Room210": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a9c8"],
"Room212": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d1"],
"Room214": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d0"],
"Room215": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a9c0"],
"Room216": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a9da"],
"Room217": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a9c1"],
"Room218": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d7"],
"Room220": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d6"],
"Room221": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a926"],
"Room222": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d5"],
"Room223": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d3"],
"Room224": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d4"],
"Room225": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a925"],
"Room226": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a9cb"],
"Room227": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a9dc"],
"Room228": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a9ca"],
"Room231": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a9c3"],
"Room233": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a9cc"],
"Room235": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a9cd"],
"Room237": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a9ce"],
"Room240": ["space_93d5acf4-73a6-4f03-a072-3fc1eab1a9dd"],
"RoomERU": ["space_bccb3261-733c-482c-8b55-56175a058e2b"]
}

D.3 botbacnt_uuid_map.json

{
"B60A.NAE227.Bacnet1.Room110.FC-1-1.ZoneTemp":
["distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14e448b"],
"B60A.NAE227.Bacnet1.Room110.FC-1-1.ZoneTempSetPoint":
["distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14ec853"],
"B60A.NAE227.Bacnet1.Room122.FC-2-1.UnitCommand":
["distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14eac48"],
"B60A.NAE227.Bacnet1.Room122.FC-2-1.UnitStatus":
["distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14ea48d"],
"B60A.NAE227.Bacnet1.Room220.FC-5-2.MultiFunctionCode":
["distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14eb558"],

"B60A.NAE227.Bacnet1.Room110.FC-1-1.FanSpeed":
["distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14ebd85"],
"B60A.NAE227.Bacnet1.Room110.FC-1-1.IndoorFanStatus":
["distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14ea4ae"],
"B60A.NAE227.Bacnet1.Room110.FC-1-1.CompressorStatus":
["distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14ea48d"],
"B60A.NAE227.Bacnet1.Room110.FC-1-1.Alarm":
["distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14eabd1"],
"B60A.NAE227.Bacnet1.Room110.FC-1-1.MultiFunctionCode":
["distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14ebda5"],
"B60A.NAE227.Bacnet1.Room110.FC-1-1.MultiFunctionCode":
["distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14eb532"]
}

APPENDIX E. PYTHON SCRIPT FOR BAS METADATA PARSING

```
# Install necessary Python packages : rdfLib
from rdflib import RDFS, RDF, Namespace, Graph, URIRef, Literal
from common import *
import json
import pandas as pd
import re

# Load mapping rules for conversion
with open('metadata/point_bacant_map.json', 'r') as fp:
    point_bacant_map = json.load(fp)
with open('metadata/space_room_map.json', 'r') as fp:
    space_room_map = json.load(fp)
with open('metadata/botbacnet_uuid_map.json', 'r') as fp:
    botbacnet_uuid_map = json.load(fp)

# Load raw BAS metadata CSV file
df = pd.read_csv('metadata/cadell_metadata_raw.csv')

######################Construct knowledge from Brick
schema###############################
brickg = Graph()
brickg.parse('Brick/dist/Brick.ttl', format='turtle') # Load Brick schema.
subclasses_query = """
PREFIX brick: <https://brickschema.org/schema/1.0.3/Brick#>
select ?tagset where {{
?tagset rdfs:subClassOf+ brick:{0}.
```

```
}}
"""
point_query = subclasses_query.format('Point')
equip_query = subclasses_query.format('Equipment')
loc_query = subclasses_query.format('Location')
points = [str(row[0]).split('#')[-1] for row in brickg.query(point_query)]
equips = [str(row[0]).split('#')[-1] for row in brickg.query(equip_query)]
locs = [str(row[0]).split('#')[-1] for row in brickg.query(loc_query)]


# Cleanup rules
equips = [equip for equip in equips if \
        not 'Command' in equip and
        not 'Sensor' in equip and
        not 'Status' in equip and
        not 'Alarm' in equip
       ]
######################Start data serialization into ttl
##################################
g = Graph() # Initialize a graph
RDFS # predefined namespace as 'http://www.w3.org/2000/01/rdf-schema#'
RDF # predefined namespace as 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'
BRICK = Namespace('https://brickschema.org/schema/1.0.3/Brick#')
BF = Namespace('https://brickschema.org/schema/1.0.3/BrickFrame#')
#EX = Namespace('http://example.com#')
EX = Namespace('https://dbl.gatech.edu/project/BIM_BAS_IoT_Metadata#')
INST = Namespace('https://dbl.gatech.edu/project/BIM_BAS_IoT_Contextual#')
g.bind('ex', EX)
g.bind('inst',INST)
g.bind('brick', BRICK)
```

```
g.bind('bf', BF)
g.bind('rdfs', RDFS)
g.bind('rdf', RDF)


for row in df.iterrows():
    vendor_name = row[1]['Vendor_Given_Name']
    uuid = row[1]['Uuid']
    ############# start Parsing ####################
    # Extract all entities in vendor_name
    entity_dict = dict()
    raw_names = vendor_name.split('.')  # This building uses '.' as a delimiter but not
always it's comprehensive.
    words = []
    for word in raw_names:
        words.append(word)
        # split more if needed
        # if 'FC' in word:
        #     words += word.split('-')
        # else:
        #     words.append(word)


    # Apply RE rules to match tagsets.
    for word in words:
        for key, tagset in point_bacant_map.items():
            if re.findall(key, word):
                if tagset in points:
                    entity_dict[vendor_name] = tagset
                else:
                    entity_dict[word] = tagset  # We will use a word as an entity name.
```

```
############# Add Brick Relationships
# Add instance relationships
for entity, tagset in entity_dict.items():
    g.add((EX[entity], RDF['type'], BRICK[tagset]))

# Add Location <-> Others
## Find all location entities
loc_entities = list()
for entity, tagset in entity_dict.items():
    if tagset in locs:
        loc_entities.append(entity)

## Assign hasLocation relationships to all non-location entities with the location
entities.
for entity, tagset in entity_dict.items():
    if tagset not in locs:
        for loc_entity in loc_entities:
            g.add((EX[entity], BF['hasLocation'], EX[loc_entity]))

# # Zone corresponds to a Fan Coil Unit.
# fc_entity = None
# for entity, tagset in entity_dict.items():
#     if tagset == 'Room':
#         fc_entity = 'FC_' + entity
#         g.add((EX[fc_entity], RDF['type'], BRICK['Fan_Coil_Unit']))
#         g.add((EX[fc_entity], BF['feeds'], EX[entity]))
# if fc_entity:
#         entity_dict[fc_entity] = 'Fan_Coil_Unit'
```

```
# Add Equip <-> Point
## Find all equip entities
equip_entities = list()
for entity, tagset in entity_dict.items():
    if tagset in equips:
        equip_entities.append(entity)

## Assign isPointOf relationships to all Point entities
for entity, tagset in entity_dict.items():
    if tagset in points:
        for equip_entity in equip_entities:
            g.add((EX[entity], BF['isPointOf'], EX[equip_entity]))

## Add inclusive relationships among equipments
# if len(equip_entities) > 1:
#     rank_equip_list = [(equip, equip_orders.index(equip))]
#     sorted(equip_entities)

# Add UUID
for entity, tagset in entity_dict.items():
    if tagset in points:
        g.add((EX[entity], BRICK['hasUuid'], Literal(str(uuid))))

# Add zone-room inclusive relationships
for space, rooms in space_room_map.items():
    for room in rooms:
        g.add((INST[room], RDF['type'], BRICK['Space']))
```

```
        g.add((INST[room], BF['hasPart'], EX[space]))

    # # Add BOT_space-BRICK_room map:
    # for rooms, spaces in space_room_map.items():
    #     for space in spaces:
    #         g.add((INST[space], RDF['type'], BRICK['Room']))
    #         g.add((INST[space], BF['isPartOf'], EX[rooms]))

    # Add BRICK_points-BOT_distribution_control_element mapping:
    for brickpoints, botbacnets in botbacnet_uuid_map.items():
        for botbacnet in botbacnets:
            g.add((INST[botbacnet], RDF['type'], BRICK['Point']))
            g.add((INST[botbacnet], BF['isControlledBy'], EX[brickpoints]))

# generate output turtle file
print (g.serialize(format='turtle'))
exportfile = open("BAS_Metadata_Export.ttl", "wb")
exportfile.write(g.serialize(format='turtle'))
```

APPENDIX F. AN EXAMPLE OF BAS METADATA SERIALIZED IN *TTL


```
@prefix bf: <https://brickschema.org/schema/1.0.3/BrickFrame#> .
@prefix brick: <https://brickschema.org/schema/1.0.3/Brick#> .
@prefix ex: <https://dbl.gatech.edu/project/BIM_BAS_IoT_Metadata#> .
@prefix inst: <https://dbl.gatech.edu/project/BIM_BAS_IoT_Contextual#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14e448b a brick:Point ;
   bf:isControlledBy ex:B60A.NAE227.Bacnet1.Room110.FC-1-1.ZoneTemp .

inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14ea390 a brick:Point ;
   bf:isControlledBy ex:B60A.NAE227.Bacnet1.Room110.FC-1-1.UnitStatus .

inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14ea48d a brick:Point ;
   bf:isControlledBy ex:B60A.NAE227.Bacnet1.Room110.FC-1-1.CompressorStatus,
      ex:B60A.NAE227.Bacnet1.Room110.FC-1-1.UnitStatus .

inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14ea4ae a brick:Point ;
   bf:isControlledBy ex:B60A.NAE227.Bacnet1.Room110.FC-1-1.IndoorFanStatus .

inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14eabd1 a brick:Point ;
   bf:isControlledBy ex:B60A.NAE227.Bacnet1.Room110.FC-1-1.Alarm .

inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14eac48 a brick:Point ;
   bf:isControlledBy ex:B60A.NAE227.Bacnet1.Room110.FC-1-1.UnitCommand .

inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14eb532 a brick:Point ;
   bf:isControlledBy ex:B60A.NAE227.Bacnet1.Room110.FC-1-1.MultiFunctionCode-1 .

inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14ebda5 a brick:Point ;
   bf:isControlledBy ex:B60A.NAE227.Bacnet1.Room110.FC-1-1.MultiFunctionCode-2 .

inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14eb558 a brick:Point ;
   bf:isControlledBy ex:B60A.NAE227.Bacnet1.Room110.FC-1-1.OperationMode .

inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14ebd85 a brick:Point ;
   bf:isControlledBy ex:B60A.NAE227.Bacnet1.Room110.FC-1-1.FanSpeed .
```

```
inst:distributioncontrolelement_a5e22f44-9e68-4456-840a-3ebfa14ec853 a brick:Point ;
    bf:isControlledBy ex:B60A.NAE227.Bacnet1.Room110.FC-1-1.ZoneTempSetPoint .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a924 a brick:Space ;
    bf:hasPart ex:Room207 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a925 a brick:Space ;
    bf:hasPart ex:Room225 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a926 a brick:Space ;
    bf:hasPart ex:Room221 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a988 a brick:Space ;
    bf:hasPart ex:Room102 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a989 a brick:Space ;
    bf:hasPart ex:Room110 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a98b a brick:Space ;
    bf:hasPart ex:Room105 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a98d a brick:Space ;
    bf:hasPart ex:Room109 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a98e a brick:Space ;
    bf:hasPart ex:Room108 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a990 a brick:Space ;
    bf:hasPart ex:Room103 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a991 a brick:Space ;
    bf:hasPart ex:Room121 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a992 a brick:Space ;
    bf:hasPart ex:Room122 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a993 a brick:Space ;
    bf:hasPart ex:Room112 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a994 a brick:Space ;
    bf:hasPart ex:Room107 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a995 a brick:Space ;
    bf:hasPart ex:Room104 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a997 a brick:Space ;
```

bf:hasPart ex:Room106 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a998 a brick:Space ;
bf:hasPart ex:Room117 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a999 a brick:Space ;
bf:hasPart ex:Room101 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a99a a brick:Space ;
bf:hasPart ex:Room100 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a99c a brick:Space ;
bf:hasPart ex:Room114 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a99d a brick:Space ;
bf:hasPart ex:Room111 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a99e a brick:Space ;
bf:hasPart ex:Room116 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a99f a brick:Space ;
bf:hasPart ex:Lobby120 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9ba a brick:Space ;
bf:hasPart ex:Room124 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9c0 a brick:Space ;
bf:hasPart ex:Room215 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9c1 a brick:Space ;
bf:hasPart ex:Room217 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9c2 a brick:Space ;
bf:hasPart ex:Room206 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9c3 a brick:Space ;
bf:hasPart ex:Room231 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9c5 a brick:Space ;
bf:hasPart ex:Room204 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9c8 a brick:Space ;
bf:hasPart ex:Room210 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9c9 a brick:Space ;
bf:hasPart ex:Room208 .

```
inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9ca a brick:Space ;
   bf:hasPart ex:Room228 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9cb a brick:Space ;
   bf:hasPart ex:Room226 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9cc a brick:Space ;
   bf:hasPart ex:Room233 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9cd a brick:Space ;
   bf:hasPart ex:Room235 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9ce a brick:Space ;
   bf:hasPart ex:Room237 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d0 a brick:Space ;
   bf:hasPart ex:Room214 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d1 a brick:Space ;
   bf:hasPart ex:Room212 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d2 a brick:Space ;
   bf:hasPart ex:Room200 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d3 a brick:Space ;
   bf:hasPart ex:Room223 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d4 a brick:Space ;
   bf:hasPart ex:Room224 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d5 a brick:Space ;
   bf:hasPart ex:Room222 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d6 a brick:Space ;
   bf:hasPart ex:Room220 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d7 a brick:Space ;
   bf:hasPart ex:Room218 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d8 a brick:Space ;
   bf:hasPart ex:Room209 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9d9 a brick:Space ;
   bf:hasPart ex:Room205 .
```

```
inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9da a brick:Space ;
    bf:hasPart ex:Room216 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9db a brick:Space ;
    bf:hasPart ex:Room201 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9dc a brick:Space ;
    bf:hasPart ex:Room227 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9dd a brick:Space ;
    bf:hasPart ex:Room240 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9de a brick:Space ;
    bf:hasPart ex:Room203 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9df a brick:Space ;
    bf:hasPart ex:Room202 .

inst:space_93d5acf4-73a6-4f03-a072-3fc1eab1a9e5 a brick:Space ;
    bf:hasPart ex:Room126 .

inst:space_bccb3261-733c-482c-8b55-56175a058e2b a brick:Space ;
    bf:hasPart ex:RoomERU .

<https://dbl.gatech.edu/project/BIM_BAS_IoT_Metadata#AHULoc/Net.RoomERU>      a
brick:Mode_Status ;
    brick:hasUuid "MSV_38" ;
    bf:hasLocation ex:RoomERU .

ex:ActiveFault.RoomERU a brick:Alarm ;
    brick:hasUuid "AV_26" ;
    bf:hasLocation ex:RoomERU .

ex:ActiveProblem.RoomERU a brick:Alarm ;
    brick:hasUuid "AV_25" ;
    bf:hasLocation ex:RoomERU .

ex:ActiveWarning.RoomERU a brick:Alarm ;
    brick:hasUuid "AV_24" ;
    bf:hasLocation ex:RoomERU .

ex:AirFlwSw.RoomERU a brick:Air_Flow_Loss_Alarm ;
    brick:hasUuid "BI_1" ;
    bf:hasLocation ex:RoomERU .

ex:AlarmValue.RoomERU a brick:Alarm ;
```

```
   brick:hasUuid "AV_27" ;
   bf:hasLocation ex:RoomERU .

ex:ApplicCmd.RoomERU a brick:Command ;
   brick:hasUuid "MSV_5" ;
   bf:hasLocation ex:RoomERU .

ex:B095E_U10U8 a brick:Hot_Water_Pump ;
   bf:hasInput
"http://ionsvr2.fac.gatech.edu/ion/default.aspx?dgm=OPEN_TEMPLATE_DIAGRAM&
node=GTECH.B095E_U10U&logServerName=QUERYSERVER.IONSVR2&logServer
Handle=327952" ;
   bf:hasLocation ex:Room108 .

ex:B095E_U11U8 a brick:Lighting_System ;
   bf:hasInput
"http://ionsvr2.fac.gatech.edu/ion/default.aspx?dgm=OPEN_TEMPLATE_DIAGRAM&
node=GTECH.B095E_U11U8&logServerName=QUERYSERVER.IONSVR2&logServe
rHandle=327952" ;
   bf:hasLocation ex:Room108 .

ex:B095E_U1H1 a brick:Electrical_Meter ;
   bf:hasInput
"http://ionsvr2.fac.gatech.edu/ion/default.aspx?dgm=OPEN_TEMPLATE_DIAGRAM&
node=GTECH.B095E_U1H1&logServerName=QUERYSERVER.IONSVR2&logServer
Handle=327952" ;
   bf:hasLocation ex:Room108 .

ex:B095E_U2U1 a brick:Elevator ;
   bf:hasInput
"http://ionsvr2.fac.gatech.edu/ion/default.aspx?dgm=OPEN_TEMPLATE_DIAGRAM&
node=GTECH.B095E_U2U1&logServerName=QUERYSERVER.IONSVR2&logServer
Handle=327952" ;
   bf:hasLocation ex:Room108 .

ex:B095E_U3U2 a brick:Electrical_Meter ;
   bf:hasInput
"http://ionsvr2.fac.gatech.edu/ion/default.aspx?dgm=OPEN_TEMPLATE_DIAGRAM&
node=GTECH.B095E_U3U2&logServerName=QUERYSERVER.IONSVR2&logServer
Handle=327952" ;
   bf:hasLocation ex:Room108 .

ex:B095E_U4U1 a brick:Boiler ;
   bf:hasInput
"http://ionsvr2.fac.gatech.edu/ion/default.aspx?dgm=OPEN_TEMPLATE_DIAGRAM&
node=GTECH.B095E_U4U1&logServerName=QUERYSERVER.IONSVR2&logServer
```

```
Handle=327952" ;
   bf:hasLocation ex:Room108 .

ex:B095E_U5U1 a brick:Electrical_System ;
   bf:hasInput
"http://ionsvr2.fac.gatech.edu/ion/default.aspx?dgm=OPEN_TEMPLATE_DIAGRAM&
node=GTECH.B095E_U5U1&logServerName=QUERYSERVER.IONSVR2&logServer
Handle=327952" ;
   bf:hasLocation ex:Room108 .

ex:B095E_U6U1 a brick:Lighting_System ;
   bf:hasInput
"http://ionsvr2.fac.gatech.edu/ion/default.aspx?dgm=OPEN_TEMPLATE_DIAGRAM&
node=GTECH.B095E_U6U1&logServerName=QUERYSERVER.IONSVR2&logServer
Handle=327952" ;
   bf:hasLocation ex:Room108 .

ex:B095E_U7U5 a brick:Lighting_System ;
   bf:hasInput
"http://ionsvr2.fac.gatech.edu/ion/default.aspx?dgm=OPEN_TEMPLATE_DIAGRAM&
node=GTECH.B095E_U7U5&logServerName=QUERYSERVER.IONSVR2&logServer
Handle=327952" ;
   bf:hasLocation ex:Room108 .

ex:B095E_U8U1 a brick:Electrical_System ;
   bf:hasInput
"http://ionsvr2.fac.gatech.edu/ion/default.aspx?dgm=OPEN_TEMPLATE_DIAGRAM&
node=GTECH.B095E_U8U1&logServerName=QUERYSERVER.IONSVR2&logServer
Handle=327952" ;
   bf:hasLocation ex:Room108 .

ex:B095E_U9U8 a brick:HVAC ;
   bf:hasInput
"http://ionsvr2.fac.gatech.edu/ion/default.aspx?dgm=OPEN_TEMPLATE_DIAGRAM&
node=GTECH.B095E_U9U8&logServerName=QUERYSERVER.IONSVR2&logServer
Handle=327952" ;
   bf:hasLocation ex:Room108 .

ex:B60A.NAE227.Bacnet1.Lobby120.FC-1-4.Alarm a brick:Temperature_Alarm ;
   brick:hasUuid "BI_771" ;
   bf:hasLocation ex:Lobby120 ;
   bf:isPointOf ex:FC-1-4 .

ex:B60A.NAE227.Bacnet1.Lobby120.FC-1-4.CompressorStatus                    a
brick:Start_Stop_Status ;
   brick:hasUuid "BI_797" ;
```

```
   bf:hasLocation ex:Lobby120 ;
   bf:isPointOf ex:FC-1-4 .

ex:B60A.NAE227.Bacnet1.Lobby120.FC-1-4.FanSpeed a brick:Speed_Sensor ;
   brick:hasUuid "MSO_775" ;
   bf:hasLocation ex:Lobby120 ;
   bf:isPointOf ex:FC-1-4 .

ex:B60A.NAE227.Bacnet1.Lobby120.FC-1-4.IndoorFanStatus a brick:Fan_Status ;
   brick:hasUuid "BI_798" ;
   bf:hasLocation ex:Lobby120 ;
   bf:isPointOf ex:FC-1-4 .

ex:B60A.NAE227.Bacnet1.Lobby120.FC-1-4.MultiFunctionCode a
brick:Mode_Command ;
   brick:hasUuid "MSI_772",
      "MSO_773" ;
   bf:hasLocation ex:Lobby120 ;
   bf:isPointOf ex:FC-1-4 .

ex:B60A.NAE227.Bacnet1.Lobby120.FC-1-4.OperationMode a brick:Fan_Status ;
   brick:hasUuid "MSI_774" ;
   bf:hasLocation ex:Lobby120 ;
   bf:isPointOf ex:FC-1-4 .

ex:B60A.NAE227.Bacnet1.Lobby120.FC-1-4.UnitCommand a
brick:Start_Stop_Command ;
   brick:hasUuid "BP_513" ;
   bf:hasLocation ex:Lobby120 ;
   bf:isPointOf ex:FC-1-4 .

ex:B60A.NAE227.Bacnet1.Lobby120.FC-1-4.UnitStatus a brick:Start_Stop_Status ;
   brick:hasUuid "BI_770" ;
   bf:hasLocation ex:Lobby120 ;
   bf:isPointOf ex:FC-1-4 .

ex:B60A.NAE227.Bacnet1.Lobby120.FC-1-4.ZoneTemp a brick:Temperature_Sensor ;
   brick:hasUuid "AI_1289" ;
   bf:hasLocation ex:Lobby120 ;
   bf:isPointOf ex:FC-1-4 .

ex:B60A.NAE227.Bacnet1.Lobby120.FC-1-4.ZoneTempSetPoint a
brick:Zone_Temperature_Setpoint ;
   brick:hasUuid "AV_778" ;
   bf:hasLocation ex:Lobby120 ;
   bf:isPointOf ex:FC-1-4 .
```

```
ex:B60A.NAE227.Bacnet1.Lobby120.FC-1-5.Alarm a brick:Temperature_Alarm ;
    brick:hasUuid "BI_1027" ;
    bf:hasLocation ex:Lobby120 ;
    bf:isPointOf ex:FC-1-5 .

ex:B60A.NAE227.Bacnet1.Lobby120.FC-1-5.CompressorStatus                          a
brick:Start_Stop_Status ;
    brick:hasUuid "BI_1053" ;
    bf:hasLocation ex:Lobby120 ;
    bf:isPointOf ex:FC-1-5 .

ex:B60A.NAE227.Bacnet1.Lobby120.FC-1-5.FanSpeed a brick:Speed_Sensor ;
    brick:hasUuid "MSP_1031" ;
    bf:hasLocation ex:Lobby120 ;
    bf:isPointOf ex:FC-1-5 .

ex:B60A.NAE227.Bacnet1.Lobby120.FC-1-5.IndoorFanStatus a brick:Fan_Status ;
    brick:hasUuid "BI_1054" ;
    bf:hasLocation ex:Lobby120 ;
    bf:isPointOf ex:FC-1-5 .

ex:B60A.NAE227.Bacnet1.Lobby120.FC-1-5.MultiFunctionCode                         a
brick:Mode_Command ;
    brick:hasUuid "MSI_1028",
        "MSO_1029" ;
    bf:hasLocation ex:Lobby120 ;
    bf:isPointOf ex:FC-1-5 .

ex:B60A.NAE227.Bacnet1.Lobby120.FC-1-5.OperationMode a brick:Fan_Status ;
    brick:hasUuid "MSI_1030" ;
    bf:hasLocation ex:Lobby120 ;
    bf:isPointOf ex:FC-1-5 .

ex:B60A.NAE227.Bacnet1.Lobby120.FC-1-5.UnitCommand                               a
brick:Start_Stop_Command ;
    brick:hasUuid "BO_769" ;
    bf:hasLocation ex:Lobby120 ;
    bf:isPointOf ex:FC-1-5 .

ex:B60A.NAE227.Bacnet1.Lobby120.FC-1-5.UnitStatus a brick:Start_Stop_Status ;
    brick:hasUuid "BI_1026" ;
    bf:hasLocation ex:Lobby120 ;
    bf:isPointOf ex:FC-1-5 .

ex:B60A.NAE227.Bacnet1.Lobby120.FC-1-5.ZoneTemp a brick:Temperature_Sensor ;
```

brick:hasUuid "AI_2057" ;
bf:hasLocation ex:Lobby120 ;
bf:isPointOf ex:FC-1-5 .

ex:B60A.NAE227.Bacnet1.Lobby120.FC-1-5.ZoneTempSetPoint a brick:Zone_Temperature_Setpoint ;
brick:hasUuid "AV_1034" ;
bf:hasLocation ex:Lobby120 ;
bf:isPointOf ex:FC-1-5 .

ex:B60A.NAE227.Bacnet1.Lobby120.FC-1-6.Alarm a brick:Temperature_Alarm ;
brick:hasUuid "BI_1283" ;
bf:hasLocation ex:Lobby120 ;
bf:isPointOf ex:FC-1-6 .

ex:B60A.NAE227.Bacnet1.Lobby120.FC-1-6.CompressorStatus a brick:Start_Stop_Status ;
brick:hasUuid "BI_1309" ;
bf:hasLocation ex:Lobby120 ;
bf:isPointOf ex:FC-1-6 .

ex:B60A.NAE227.Bacnet1.Lobby120.FC-1-6.FanSpeed a brick:Speed_Sensor ;
brick:hasUuid "MSP_1287" ;
bf:hasLocation ex:Lobby120 ;
bf:isPointOf ex:FC-1-6 .

ex:B60A.NAE227.Bacnet1.Lobby120.FC-1-6.IndoorFanStatus a brick:Fan_Status ;
brick:hasUuid "BI_1310" ;
bf:hasLocation ex:Lobby120 ;
bf:isPointOf ex:FC-1-6 .

ex:B60A.NAE227.Bacnet1.Lobby120.FC-1-6.MultiFunctionCode a brick:Mode_Command ;
brick:hasUuid "MSI_1284",
"MSO_1285" ;
bf:hasLocation ex:Lobby120 ;
bf:isPointOf ex:FC-1-6 .

ex:B60A.NAE227.Bacnet1.Lobby120.FC-1-6.OperationMode a brick:Fan_Status ;
brick:hasUuid "MSI_1286" ;
bf:hasLocation ex:Lobby120 ;
bf:isPointOf ex:FC-1-6 .

ex:B60A.NAE227.Bacnet1.Lobby120.FC-1-6.UnitCommand a brick:Start_Stop_Command ;
brick:hasUuid "BO_1025" ;

```
    bf:hasLocation ex:Lobby120 ;
    bf:isPointOf ex:FC-1-6 .

ex:B60A.NAE227.Bacnet1.Lobby120.FC-1-6.UnitStatus a brick:Start_Stop_Status ;
    brick:hasUuid "BI_1282" ;
    bf:hasLocation ex:Lobby120 ;
    bf:isPointOf ex:FC-1-6 .

ex:B60A.NAE227.Bacnet1.Lobby120.FC-1-6.ZoneTemp a brick:Temperature_Sensor ;
    brick:hasUuid "AI_2313" ;
    bf:hasLocation ex:Lobby120 ;
    bf:isPointOf ex:FC-1-6 .

ex:B60A.NAE227.Bacnet1.Lobby120.FC-1-6.ZoneTempSetPoint                    a
brick:Zone_Temperature_Setpoint ;
    brick:hasUuid "AV_1290" ;
    bf:hasLocation ex:Lobby120 ;
    bf:isPointOf ex:FC-1-6 .

ex:B60A.NAE227.Bacnet1.Room110.FC-1-1.OperationMode a brick:Fan_Status ;
    brick:hasUuid "MSI_6" ;
    bf:hasLocation ex:Room110 ;
    bf:isPointOf ex:FC-1-1 .

ex:B60A.NAE227.Bacnet1.Room110.FC-1-1.UnitCommand                    a
brick:Start_Stop_Command ;
    brick:hasUuid "BO_1" ;
    bf:hasLocation ex:Room110 ;
    bf:isPointOf ex:FC-1-1 .

ex:B60A.NAE227.Bacnet1.Room110.FC-1-1.UnitStatus a brick:Start_Stop_Status ;
    brick:hasUuid "BI_2" ;
    bf:hasLocation ex:Room110 ;
    bf:isPointOf ex:FC-1-1 .

ex:B60A.NAE227.Bacnet1.Room112.FC-1-2.Alarm a brick:Temperature_Alarm ;
    brick:hasUuid "BI_259" ;
    bf:hasLocation ex:Room112 ;
    bf:isPointOf ex:FC-1-2 .

ex:B60A.NAE227.Bacnet1.Room112.FC-1-2.CompressorStatus                    a
brick:Start_Stop_Status ;
    brick:hasUuid "BI_285" ;
    bf:hasLocation ex:Room112 ;
    bf:isPointOf ex:FC-1-2 .
```

```
ex:B60A.NAE227.Bacnet1.Room112.FC-1-2.FanSpeed a brick:Speed_Sensor ;
   brick:hasUuid "MSO_263" ;
   bf:hasLocation ex:Room112 ;
   bf:isPointOf ex:FC-1-2 .

ex:B60A.NAE227.Bacnet1.Room112.FC-1-2.IndoorFanStatus a brick:Fan_Status ;
   brick:hasUuid "BI_286" ;
   bf:hasLocation ex:Room112 ;
   bf:isPointOf ex:FC-1-2 .

ex:B60A.NAE227.Bacnet1.Room112.FC-1-2.MultiFunctionCode                    a
brick:Mode_Command ;
   brick:hasUuid "MSI_260",
      "MSO_261" ;
   bf:hasLocation ex:Room112 ;
   bf:isPointOf ex:FC-1-2 .

ex:B60A.NAE227.Bacnet1.Room112.FC-1-2.OperationMode a brick:Fan_Status ;
   brick:hasUuid "MSI_262" ;
   bf:hasLocation ex:Room112 ;
   bf:isPointOf ex:FC-1-2 .

ex:B60A.NAE227.Bacnet1.Room112.FC-1-2.UnitCommand                          a
brick:Start_Stop_Command ;
   brick:hasUuid "BO_7169" ;
   bf:hasLocation ex:Room112 ;
   bf:isPointOf ex:FC-1-2 .

ex:B60A.NAE227.Bacnet1.Room112.FC-1-2.UnitStatus a brick:Start_Stop_Status ;
   brick:hasUuid "BI_258" ;
   bf:hasLocation ex:Room112 ;
   bf:isPointOf ex:FC-1-2 .

ex:B60A.NAE227.Bacnet1.Room112.FC-1-2.ZoneTemp a brick:Temperature_Sensor ;
   brick:hasUuid "AI_265" ;
   bf:hasLocation ex:Room112 ;
   bf:isPointOf ex:FC-1-2 .

ex:B60A.NAE227.Bacnet1.Room112.FC-1-2.ZoneTempSetPoint                     a
brick:Zone_Temperature_Setpoint ;
   brick:hasUuid "AV_266" ;
   bf:hasLocation ex:Room112 ;
   bf:isPointOf ex:FC-1-2 .

ex:B60A.NAE227.Bacnet1.Room114.FC-1-3.Alarm a brick:Temperature_Alarm ;
   brick:hasUuid "BI_515" ;
```

```
   bf:hasLocation ex:Room114 ;
   bf:isPointOf ex:FC-1-3 .

ex:B60A.NAE227.Bacnet1.Room114.FC-1-3.CompressorStatus a
brick:Start_Stop_Status ;
   brick:hasUuid "BI_541" ;
   bf:hasLocation ex:Room114 ;
   bf:isPointOf ex:FC-1-3 .

ex:B60A.NAE227.Bacnet1.Room114.FC-1-3.FanSpeed a brick:Speed_Sensor ;
   brick:hasUuid "MSO_519" ;
   bf:hasLocation ex:Room114 ;
   bf:isPointOf ex:FC-1-3 .

ex:B60A.NAE227.Bacnet1.Room114.FC-1-3.IndoorFanStatus a brick:Fan_Status ;
   brick:hasUuid "BI_542" ;
   bf:hasLocation ex:Room114 ;
   bf:isPointOf ex:FC-1-3 .

ex:B60A.NAE227.Bacnet1.Room114.FC-1-3.MultiFunctionCode a
brick:Mode_Command ;
   brick:hasUuid "MSI_516",
      "MSO_517" ;
   bf:hasLocation ex:Room114 ;
   bf:isPointOf ex:FC-1-3 .

ex:B60A.NAE227.Bacnet1.Room114.FC-1-3.OperationMode a brick:Fan_Status ;
   brick:hasUuid "MSI_518" ;
   bf:hasLocation ex:Room114 ;
   bf:isPointOf ex:FC-1-3 .

ex:B60A.NAE227.Bacnet1.Room114.FC-1-3.UnitCommand a
brick:Start_Stop_Command ;
   brick:hasUuid "BO_257" ;
   bf:hasLocation ex:Room114 ;
   bf:isPointOf ex:FC-1-3 .

ex:B60A.NAE227.Bacnet1.Room114.FC-1-3.UnitStatus a brick:Start_Stop_Status ;
   brick:hasUuid "BI_514" ;
   bf:hasLocation ex:Room114 ;
   bf:isPointOf ex:FC-1-3 .

ex:B60A.NAE227.Bacnet1.Room114.FC-1-3.ZoneTemp a brick:Temperature_Sensor ;
   brick:hasUuid "AI_521" ;
   bf:hasLocation ex:Room114 ;
   bf:isPointOf ex:FC-1-3 .
```

ex:B60A.NAE227.Bacnet1.Room114.FC-1-3.ZoneTempSetPoint a brick:Zone_Temperature_Setpoint ;
brick:hasUuid "AV_522" ;
bf:hasLocation ex:Room114 ;
bf:isPointOf ex:FC-1-3 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-1.Alarm a brick:Temperature_Alarm ;
brick:hasUuid "BI_2051" ;
bf:hasLocation ex:Room122 ;
bf:isPointOf ex:FC-2-1 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-1.CompressorStatus a brick:Start_Stop_Status ;
brick:hasUuid "BI_2077" ;
bf:hasLocation ex:Room122 ;
bf:isPointOf ex:FC-2-1 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-1.FanSpeed a brick:Speed_Sensor ;
brick:hasUuid "MSP_2055" ;
bf:hasLocation ex:Room122 ;
bf:isPointOf ex:FC-2-1 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-1.IndoorFanStatus a brick:Fan_Status ;
brick:hasUuid "BI_2078" ;
bf:hasLocation ex:Room122 ;
bf:isPointOf ex:FC-2-1 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-1.MultiFunctionCode a brick:Mode_Command ;
brick:hasUuid "MSI_2052",
"MSO_2053" ;
bf:hasLocation ex:Room122 ;
bf:isPointOf ex:FC-2-1 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-1.OperationMode a brick:Fan_Status ;
brick:hasUuid "MSI_2054" ;
bf:hasLocation ex:Room122 ;
bf:isPointOf ex:FC-2-1 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-1.ZoneTemp a brick:Temperature_Sensor ;
brick:hasUuid "AI_2569" ;
bf:hasLocation ex:Room122 ;
bf:isPointOf ex:FC-2-1 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-1.ZoneTempSetPoint a

```
brick:Zone_Temperature_Setpoint ;
   brick:hasUuid "AV_2058" ;
   bf:hasLocation ex:Room122 ;
   bf:isPointOf ex:FC-2-1 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-2.Alarm a brick:Temperature_Alarm ;
   brick:hasUuid "BI_2307" ;
   bf:hasLocation ex:Room122 ;
   bf:isPointOf ex:FC-2-2 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-2.CompressorStatus a
brick:Start_Stop_Status ;
   brick:hasUuid "BI_2333" ;
   bf:hasLocation ex:Room122 ;
   bf:isPointOf ex:FC-2-2 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-2.FanSpeed a brick:Speed_Sensor ;
   brick:hasUuid "MSP_2311" ;
   bf:hasLocation ex:Room122 ;
   bf:isPointOf ex:FC-2-2 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-2.IndoorFanStatus a brick:Fan_Status ;
   brick:hasUuid "BI_2334" ;
   bf:hasLocation ex:Room122 ;
   bf:isPointOf ex:FC-2-2 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-2.MultiFunctionCode a
brick:Mode_Command ;
   brick:hasUuid "MSI_2308",
      "MSO_2309" ;
   bf:hasLocation ex:Room122 ;
   bf:isPointOf ex:FC-2-2 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-2.OperationMode a brick:Fan_Status ;
   brick:hasUuid "MSI_2310" ;
   bf:hasLocation ex:Room122 ;
   bf:isPointOf ex:FC-2-2 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-2.UnitCommand a
brick:Start_Stop_Command ;
   brick:hasUuid "BO_2049" ;
   bf:hasLocation ex:Room122 ;
   bf:isPointOf ex:FC-2-2 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-2.UnitStatus a brick:Start_Stop_Status ;
   brick:hasUuid "BI_2306" ;
```

```
    bf:hasLocation ex:Room122 ;
    bf:isPointOf ex:FC-2-2 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-2.ZoneTemp a brick:Temperature_Sensor ;
    brick:hasUuid "AI_4876" ;
    bf:hasLocation ex:Room122 ;
    bf:isPointOf ex:FC-2-2 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-2.ZoneTempSetPoint a
brick:Zone_Temperature_Setpoint ;
    brick:hasUuid "AV_2314" ;
    bf:hasLocation ex:Room122 ;
    bf:isPointOf ex:FC-2-2 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-3.Alarm a brick:Temperature_Alarm ;
    brick:hasUuid "BI_1539" ;
    bf:hasLocation ex:Room122 ;
    bf:isPointOf ex:FC-2-3 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-3.CompressorStatus a
brick:Start_Stop_Status ;
    brick:hasUuid "BI_1565" ;
    bf:hasLocation ex:Room122 ;
    bf:isPointOf ex:FC-2-3 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-3.FanSpeed a brick:Speed_Sensor ;
    brick:hasUuid "MSP_1543" ;
    bf:hasLocation ex:Room122 ;
    bf:isPointOf ex:FC-2-3 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-3.IndoorFanStatus a brick:Fan_Status ;
    brick:hasUuid "BI_1566" ;
    bf:hasLocation ex:Room122 ;
    bf:isPointOf ex:FC-2-3 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-3.MultiFunctionCode a
brick:Mode_Command ;
    brick:hasUuid "MSI_1540",
        "MSO_1541" ;
    bf:hasLocation ex:Room122 ;
    bf:isPointOf ex:FC-2-3 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-3.OperationMode a brick:Fan_Status ;
    brick:hasUuid "MSI_1542" ;
    bf:hasLocation ex:Room122 ;
    bf:isPointOf ex:FC-2-3 .
```

ex:B60A.NAE227.Bacnet1.Room122.FC-2-3.UnitCommand a brick:Start_Stop_Command ;
brick:hasUuid "BO_1281" ;
bf:hasLocation ex:Room122 ;
bf:isPointOf ex:FC-2-3 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-3.UnitStatus a brick:Start_Stop_Status ;
brick:hasUuid "BI_1538" ;
bf:hasLocation ex:Room122 ;
bf:isPointOf ex:FC-2-3 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-3.ZoneTemp a brick:Temperature_Sensor ;
brick:hasUuid "AI_1545" ;
bf:hasLocation ex:Room122 ;
bf:isPointOf ex:FC-2-3 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-3.ZoneTempSetPoint a brick:Zone_Temperature_Setpoint ;
brick:hasUuid "AV_1546" ;
bf:hasLocation ex:Room122 ;
bf:isPointOf ex:FC-2-3 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-4.Alarm a brick:Temperature_Alarm ;
brick:hasUuid "BI_1795" ;
bf:hasLocation ex:Room122 ;
bf:isPointOf ex:FC-2-4 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-4.CompressorStatus a brick:Start_Stop_Status ;
brick:hasUuid "BI_1821" ;
bf:hasLocation ex:Room122 ;
bf:isPointOf ex:FC-2-4 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-4.FanSpeed a brick:Speed_Sensor ;
brick:hasUuid "MSP_1799" ;
bf:hasLocation ex:Room122 ;
bf:isPointOf ex:FC-2-4 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-4.IndoorFanStatus a brick:Fan_Status ;
brick:hasUuid "BI_1822" ;
bf:hasLocation ex:Room122 ;
bf:isPointOf ex:FC-2-4 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-4.MultiFunctionCode a brick:Mode_Command ;

```
   brick:hasUuid "MSI_1796",
      "MSO_1797" ;
   bf:hasLocation ex:Room122 ;
   bf:isPointOf ex:FC-2-4 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-4.OperationMode a brick:Fan_Status ;
   brick:hasUuid "MSI_1798" ;
   bf:hasLocation ex:Room122 ;
   bf:isPointOf ex:FC-2-4 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-4.UnitCommand                    a
brick:Start_Stop_Command ;
   brick:hasUuid "BO_1537" ;
   bf:hasLocation ex:Room122 ;
   bf:isPointOf ex:FC-2-4 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-4.UnitStatus a brick:Start_Stop_Status ;
   brick:hasUuid "BI_1794" ;
   bf:hasLocation ex:Room122 ;
   bf:isPointOf ex:FC-2-4 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-4.ZoneTemp a brick:Temperature_Sensor ;
   brick:hasUuid "AI_1801" ;
   bf:hasLocation ex:Room122 ;
   bf:isPointOf ex:FC-2-4 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-4.ZoneTempSetPoint               a
brick:Zone_Temperature_Setpoint ;
   brick:hasUuid "AV_1802" ;
   bf:hasLocation ex:Room122 ;
   bf:isPointOf ex:FC-2-4 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-5.Alarm a brick:Temperature_Alarm ;
   brick:hasUuid "BI_2563" ;
   bf:hasLocation ex:Room122 ;
   bf:isPointOf ex:FC-2-5 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-5.CompressorStatus               a
brick:Start_Stop_Status ;
   brick:hasUuid "BI_2589" ;
   bf:hasLocation ex:Room122 ;
   bf:isPointOf ex:FC-2-5 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-5.FanSpeed a brick:Speed_Sensor ;
   brick:hasUuid "MSP_2567" ;
   bf:hasLocation ex:Room122 ;
```

```
   bf:isPointOf ex:FC-2-5 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-5.IndoorFanStatus a brick:Fan_Status ;
   brick:hasUuid "BI_2590" ;
   bf:hasLocation ex:Room122 ;
   bf:isPointOf ex:FC-2-5 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-5.MultiFunctionCode                          a
brick:Mode_Command ;
   brick:hasUuid "MSI_2564",
      "MSO_2565" ;
   bf:hasLocation ex:Room122 ;
   bf:isPointOf ex:FC-2-5 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-5.OperationMode a brick:Fan_Status ;
   brick:hasUuid "MSI_2566" ;
   bf:hasLocation ex:Room122 ;
   bf:isPointOf ex:FC-2-5 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-5.UnitCommand                                a
brick:Start_Stop_Command ;
   brick:hasUuid "BO_2305" ;
   bf:hasLocation ex:Room122 ;
   bf:isPointOf ex:FC-2-5 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-5.UnitStatus a brick:Start_Stop_Status ;
   brick:hasUuid "BI_2562" ;
   bf:hasLocation ex:Room122 ;
   bf:isPointOf ex:FC-2-5 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-5.ZoneTemp a brick:Temperature_Sensor ;
   brick:hasUuid "AI_2825" ;
   bf:hasLocation ex:Room122 ;
   bf:isPointOf ex:FC-2-5 .

ex:B60A.NAE227.Bacnet1.Room122.FC-2-5.ZoneTempSetPoint                           a
brick:Zone_Temperature_Setpoint ;
   brick:hasUuid "AV_2570" ;
   bf:hasLocation ex:Room122 ;
   bf:isPointOf ex:FC-2-5 .

ex:B60A.NAE227.Bacnet1.Room203.FC-4-3.Alarm a brick:Temperature_Alarm ;
   brick:hasUuid "BI_4867" ;
   bf:hasLocation ex:Room203 ;
   bf:isPointOf ex:FC-4-3 .
```

ex:B60A.NAE227.Bacnet1.Room203.FC-4-3.CompressorStatus a brick:Start_Stop_Status ;
brick:hasUuid "BI_4893" ;
bf:hasLocation ex:Room203 ;
bf:isPointOf ex:FC-4-3 .

ex:B60A.NAE227.Bacnet1.Room203.FC-4-3.FanSpeed a brick:Speed_Sensor ;
brick:hasUuid "MSP_4871" ;
bf:hasLocation ex:Room203 ;
bf:isPointOf ex:FC-4-3 .

ex:B60A.NAE227.Bacnet1.Room203.FC-4-3.IndoorFanStatus a brick:Fan_Status ;
brick:hasUuid "BI_4894" ;
bf:hasLocation ex:Room203 ;
bf:isPointOf ex:FC-4-3 .

ex:B60A.NAE227.Bacnet1.Room203.FC-4-3.MultiFunctionCode a brick:Mode_Command ;
brick:hasUuid "MSI_4868",
"MSO_4869" ;
bf:hasLocation ex:Room203 ;
bf:isPointOf ex:FC-4-3 .

ex:B60A.NAE227.Bacnet1.Room203.FC-4-3.OperationMode a brick:Fan_Status ;
brick:hasUuid "MSI_4870" ;
bf:hasLocation ex:Room203 ;
bf:isPointOf ex:FC-4-3 .

ex:B60A.NAE227.Bacnet1.Room203.FC-4-3.UnitCommand a brick:Start_Stop_Command ;
brick:hasUuid "BO_4609" ;
bf:hasLocation ex:Room203 ;
bf:isPointOf ex:FC-4-3 .

ex:B60A.NAE227.Bacnet1.Room203.FC-4-3.UnitStatus a brick:Start_Stop_Status ;
brick:hasUuid "BI_4866" ;
bf:hasLocation ex:Room203 ;
bf:isPointOf ex:FC-4-3 .

ex:B60A.NAE227.Bacnet1.Room203.FC-4-3.ZoneTemp a brick:Temperature_Sensor ;
brick:hasUuid "AI_6921" ;
bf:hasLocation ex:Room203 ;
bf:isPointOf ex:FC-4-3 .

ex:B60A.NAE227.Bacnet1.Room203.FC-4-3.ZoneTempSetPoint a brick:Zone_Temperature_Setpoint ;

brick:hasUuid "AV_4874" ;
bf:hasLocation ex:Room203 ;
bf:isPointOf ex:FC-4-3 .

ex:B60A.NAE227.Bacnet1.Room208.FC-3-1.Alarm a brick:Temperature_Alarm ;
brick:hasUuid "BI_2819" ;
bf:hasLocation ex:Room208 ;
bf:isPointOf ex:FC-3-1 .

ex:B60A.NAE227.Bacnet1.Room208.FC-3-1.CompressorStatus a
brick:Start_Stop_Status ;
brick:hasUuid "BI_2845" ;
bf:hasLocation ex:Room208 ;
bf:isPointOf ex:FC-3-1 .

ex:B60A.NAE227.Bacnet1.Room208.FC-3-1.FanSpeed a brick:Speed_Sensor ;
brick:hasUuid "MSP_2823" ;
bf:hasLocation ex:Room208 ;
bf:isPointOf ex:FC-3-1 .

ex:B60A.NAE227.Bacnet1.Room208.FC-3-1.IndoorFanStatus a brick:Fan_Status ;
brick:hasUuid "BI_2846" ;
bf:hasLocation ex:Room208 ;
bf:isPointOf ex:FC-3-1 .

ex:B60A.NAE227.Bacnet1.Room208.FC-3-1.MultiFunctionCode a
brick:Mode_Command ;
brick:hasUuid "MSI_2820",
"MSO_2821" ;
bf:hasLocation ex:Room208 ;
bf:isPointOf ex:FC-3-1 .

ex:B60A.NAE227.Bacnet1.Room208.FC-3-1.OperationMode a brick:Fan_Status ;
brick:hasUuid "MSI_2822" ;
bf:hasLocation ex:Room208 ;
bf:isPointOf ex:FC-3-1 .

ex:B60A.NAE227.Bacnet1.Room208.FC-3-1.UnitCommand a
brick:Start_Stop_Command ;
brick:hasUuid "BO_2561" ;
bf:hasLocation ex:Room208 ;
bf:isPointOf ex:FC-3-1 .

ex:B60A.NAE227.Bacnet1.Room208.FC-3-1.UnitStatus a brick:Start_Stop_Status ;
brick:hasUuid "BI_2818" ;
bf:hasLocation ex:Room208 ;

```
  bf:isPointOf ex:FC-3-1 .

ex:B60A.NAE227.Bacnet1.Room208.FC-3-1.ZoneTemp a brick:Temperature_Sensor ;
  brick:hasUuid "AI_3081" ;
  bf:hasLocation ex:Room208 ;
  bf:isPointOf ex:FC-3-1 .

ex:B60A.NAE227.Bacnet1.Room208.FC-3-1.ZoneTempSetPoint a
brick:Zone_Temperature_Setpoint ;
  brick:hasUuid "AV_2826" ;
  bf:hasLocation ex:Room208 ;
  bf:isPointOf ex:FC-3-1 .

ex:B60A.NAE227.Bacnet1.Room210.FC-3-2.Alarm a brick:Temperature_Alarm ;
  brick:hasUuid "BI_3075" ;
  bf:hasLocation ex:Room210 ;
  bf:isPointOf ex:FC-3-2 .

ex:B60A.NAE227.Bacnet1.Room210.FC-3-2.CompressorStatus a
brick:Start_Stop_Status ;
  brick:hasUuid "BI_3101" ;
  bf:hasLocation ex:Room210 ;
  bf:isPointOf ex:FC-3-2 .

ex:B60A.NAE227.Bacnet1.Room210.FC-3-2.FanSpeed a brick:Speed_Sensor ;
  brick:hasUuid "MSP_3079" ;
  bf:hasLocation ex:Room210 ;
  bf:isPointOf ex:FC-3-2 .

ex:B60A.NAE227.Bacnet1.Room210.FC-3-2.IndoorFanStatus a brick:Fan_Status ;
  brick:hasUuid "BI_3102" ;
  bf:hasLocation ex:Room210 ;
  bf:isPointOf ex:FC-3-2 .

ex:B60A.NAE227.Bacnet1.Room210.FC-3-2.MultiFunctionCode a
brick:Mode_Command ;
  brick:hasUuid "MSI_3076",
    "MSO_3077" ;
  bf:hasLocation ex:Room210 ;
  bf:isPointOf ex:FC-3-2 .

ex:B60A.NAE227.Bacnet1.Room210.FC-3-2.OperationMode a brick:Fan_Status ;
  brick:hasUuid "MSI_3078" ;
  bf:hasLocation ex:Room210 ;
  bf:isPointOf ex:FC-3-2 .
```

ex:B60A.NAE227.Bacnet1.Room210.FC-3-2.UnitCommand a brick:Start_Stop_Command ;
brick:hasUuid "BO_2817" ;
bf:hasLocation ex:Room210 ;
bf:isPointOf ex:FC-3-2 .

ex:B60A.NAE227.Bacnet1.Room210.FC-3-2.UnitStatus a brick:Start_Stop_Status ;
brick:hasUuid "BI_3074" ;
bf:hasLocation ex:Room210 ;
bf:isPointOf ex:FC-3-2 .

ex:B60A.NAE227.Bacnet1.Room210.FC-3-2.ZoneTemp a brick:Temperature_Sensor ;
brick:hasUuid "AI_4361" ;
bf:hasLocation ex:Room210 ;
bf:isPointOf ex:FC-3-2 .

ex:B60A.NAE227.Bacnet1.Room210.FC-3-2.ZoneTempSetPoint a brick:Zone_Temperature_Setpoint ;
brick:hasUuid "AV_3082" ;
bf:hasLocation ex:Room210 ;
bf:isPointOf ex:FC-3-2 .

ex:B60A.NAE227.Bacnet1.Room212.FC-4-1.Alarm a brick:Temperature_Alarm ;
brick:hasUuid "BI_4355" ;
bf:hasLocation ex:Room212 ;
bf:isPointOf ex:FC-4-1 .

ex:B60A.NAE227.Bacnet1.Room212.FC-4-1.CompressorStatus a brick:Start_Stop_Status ;
brick:hasUuid "BI_4381" ;
bf:hasLocation ex:Room212 ;
bf:isPointOf ex:FC-4-1 .

ex:B60A.NAE227.Bacnet1.Room212.FC-4-1.FanSpeed a brick:Speed_Sensor ;
brick:hasUuid "MSP_4359" ;
bf:hasLocation ex:Room212 ;
bf:isPointOf ex:FC-4-1 .

ex:B60A.NAE227.Bacnet1.Room212.FC-4-1.IndoorFanStatus a brick:Fan_Status ;
brick:hasUuid "BI_4382" ;
bf:hasLocation ex:Room212 ;
bf:isPointOf ex:FC-4-1 .

ex:B60A.NAE227.Bacnet1.Room212.FC-4-1.MultiFunctionCode a brick:Mode_Command ;
brick:hasUuid "MSI_4356",

```
    "MSO_4357" ;
  bf:hasLocation ex:Room212 ;
  bf:isPointOf ex:FC-4-1 .

ex:B60A.NAE227.Bacnet1.Room212.FC-4-1.OperationMode a brick:Fan_Status ;
  brick:hasUuid "MSI_4358" ;
  bf:hasLocation ex:Room212 ;
  bf:isPointOf ex:FC-4-1 .

ex:B60A.NAE227.Bacnet1.Room212.FC-4-1.UnitCommand                    a
brick:Start_Stop_Command ;
  brick:hasUuid "BO_4097" ;
  bf:hasLocation ex:Room212 ;
  bf:isPointOf ex:FC-4-1 .

ex:B60A.NAE227.Bacnet1.Room212.FC-4-1.UnitStatus a brick:Start_Stop_Status ;
  brick:hasUuid "BI_4354" ;
  bf:hasLocation ex:Room212 ;
  bf:isPointOf ex:FC-4-1 .

ex:B60A.NAE227.Bacnet1.Room212.FC-4-1.ZoneTemp a brick:Temperature_Sensor ;
  brick:hasUuid "AI_6665" ;
  bf:hasLocation ex:Room212 ;
  bf:isPointOf ex:FC-4-1 .

ex:B60A.NAE227.Bacnet1.Room212.FC-4-1.ZoneTempSetPoint               a
brick:Zone_Temperature_Setpoint ;
  brick:hasUuid "AV_4362" ;
  bf:hasLocation ex:Room212 ;
  bf:isPointOf ex:FC-4-1 .

ex:B60A.NAE227.Bacnet1.Room214.FC-4-2.Alarm a brick:Temperature_Alarm ;
  brick:hasUuid "BI_4611" ;
  bf:hasLocation ex:Room214 ;
  bf:isPointOf ex:FC-4-2 .

ex:B60A.NAE227.Bacnet1.Room214.FC-4-2.CompressorStatus               a
brick:Start_Stop_Status ;
  brick:hasUuid "BI_4637" ;
  bf:hasLocation ex:Room214 ;
  bf:isPointOf ex:FC-4-2 .
```

APPENDIX G. Sample ER_BACnet MVD.xlsx

Information Unit		Entity/Pset/Functional Part	Man	Rec	Opt	Mapped Concept
BACnet Analog Input Object Type		IfcController				
	Object Geometric Representation	IfcProduct.Representation->IfcProductDefinitionShape		√		Box Geometry; FootPrint Annotation Geometry
	Object Placement	IfcProduct.ObjectPlacement->IfcObjectPlacement		√		Product local placement
	Object Contained in Space	IfcElement.ContainedInStructure ->IfcRelContainedInSpatialStructure		√		Spatial Containment
	Object Connected From	IfcElement.ConnectedFrom->IfcRelConnectsElements		√		Path Connectivity
	Decomposes	IfcObjectDefinition.Decomposes->IfcRelAggregates		√		Object Definition Composition
	Object Identifier	IfcControllort.GlobalId->IfcGloballyUniqueId	√			Software Identity
	Object Name	IfcControllor.Name->IfcLable	√			Object User Identity
	Object Type	IfcController.PredefinedType->IfcControllerTypeEnum	√			BACnet Controllor Predefined Type; Object Predefined Type

	Present Value	Pset_ControllerTypeMultiPosition.Value->IfcInteger	√			PropertySets for Objects
	Description	IfcRoot.Description->IfcText			√	-
	Status Flags	Pset_BACnetAnalogInputObject.StatusFlags ->IfcLabel:: PEnum_BACnetStatusFlag	√			PropertySets for Objects
	Event State	Pset_BACnetAnalogInputObject.EventState ->IfcLabel:: PEnum_BACnetEventState	√			PropertySets for Objects
	Out of Service	Pset_BACnetAnalogInputObject.OutOfService ->IfcBoolean	√	`		PropertySets for Objects
	Units	Pset_BACnetAnalogInputObject.Units ->IfcLabel	√			PropertySets for Objects
	Property List	IfcPropertySet.HasPropertites->IfcProperty	√			PropertySets for Objects
	Tags	IfcElement.Tag->IfcIdentifier			√	Element Occurrence Attributes
BACnet Analog Output Object Type		IfcController				
	Object Geometric Representation	IfcProduct.Representation->IfcProductDefinitionShape		√		Box Geometry; FootPrint Annotation Geometry
	Object Placement	IfcProduct.ObjectPlacement->IfcObjectPlacement		√		Product local placement
	Object Contained in Space	IfcElement.ContainedInStructure ->IfcRelContainedInSpatialStructure		√		Spatial Containment

	Object Connected From	IfcElement.ConnectedFrom->IfcRelConnectsElements		√		Path Connectivity
	Decomposes	IfcObjectDefinition.Decomposes->IfcRelAggregates		√		Object Definition Composition
	Object Identifier	IfcRoot.GlobalId->IfcGloballyUniqueId	√			Software Identity
	Object Name	IfcRoot.Name->IfcLable	√			Object User Identity
	Object Type	IfcController.PredefinedType->IfcControllerTypeEnum	√			BACnet Controllor Predefined Type; Object Predefined Type
	Present Value	Pset_ControllerTypeMultiPosition.Value->IfcInteger	√			PropertySets for Objects
	Description	IfcRoot.Description->IfcText			√	-
	Status Flags	Pset_BACnetAnalogOutputObject.StatusFlags ->IfcLabel:: PEnum_BACnetStatusFlag	√			PropertySets for Objects
	Event State	Pset_BACnetAnalogOutputObject.EventState ->IfcLabel:: PEnum_BACnetEventState	√			PropertySets for Objects
	Out of Service	Pset_BACnetAnalogOutputObject.OutOfService ->IfcBoolean	√	`		PropertySets for Objects
	Units	Pset_BACnetAnalogOutputObject.Units ->IfcLabel	√			PropertySets for Objects
	Priority Array	Pset_BACnetAnalogOutputObject.PriorityArray ->IfcLabel:: PEnum_BACnetPriorityValue	√			PropertySets for Objects
	Relinquish Default	Pset_BACnetAnalogOutputObject.RelinquishDefault->IfcReal	√			PropertySets for Objects
	Property List	IfcPropertySet.HasPropertites->IfcProperty	√			PropertySets for Objects

	Current Command Priority	Pset_BACnetAnalogOutputObject.CurrentCommandPriority->IfcInteger	√			PropertySets for Objects
	Tags	IfcElement.Tag->IfcIdentifier			√	Element Occurrence Attributes
BACnet Analog Value Object Type		IfcController				
	Object Geometric Representation	IfcProduct.Representation->IfcProductDefinitionShape		√		Box Geometry; FootPrint Annotation Geometry
	Object Placement	IfcProduct.ObjectPlacement->IfcObjectPlacement		√		Product local placement
	Object Contained in Space	IfcElement.ContainedInStructure ->IfcRelContainedInSpatialStructure		√		Spatial Containment
	Object Connected From	IfcElement.ConnectedFrom->IfcRelConnectsElements		√		Path Connectivity
	Decomposes	IfcObjectDefinition.Decomposes->IfcRelAggregates		√		Object Definition Composition
	Object Identifier	IfcRoot.GlobalId->IfcGloballyUniqueId	√			Software Identity
	Object Name	IfcRoot.Name->IfcLable	√			Object User Identity
	Object Type	IfcController.PredefinedType->IfcControllerTypeEnum	√			BACnet Controllor Predefined Type; Object Predefined Type
	Present Value	Pset_ControllerTypeMultiPosition.Value->IfcInteger	√			PropertySets for Objects

	Description	IfcRoot.Description->IfcText			√	-
	Status Flags	Pset_BACnetAnalogValueObject.StatusFlags ->IfcLabel:: PEnum_BACnetStatusFlag	√			PropertySets for Objects
	Event State	Pset_BACnetAnalogValueObject.EventState ->IfcLabel:: PEnum_BACnetEventState	√			PropertySets for Objects
	Out of Service	Pset_BACnetAnalogValueObject.OutOfService ->IfcBoolean	√	`		PropertySets for Objects
	Units	Pset_BACnetAnalogInputObject.Units ->IfcLabel	√			PropertySets for Objects
	Property List	IfcPropertySet	√			PropertySets for Objects
	Tags	IfcElement.Tag->IfcIdentifier			√	Element Occurrence Attributes
BACnet Averaging Object Type		IfcController, predefined type=FLOATING				
	Object Geometric Representation	IfcProduct.Representation->IfcProductDefinitionShape		√		Box Geometry; FootPrint Annotation Geometry
	Object Placement	IfcProduct.ObjectPlacement->IfcObjectPlacement		√		Product local placement
	Object Contained in Space	IfcElement.ContainedInStructure ->IfcRelContainedInSpatialStructure		√		Spatial Containment
	Object Connected From	IfcElement.ConnectedFrom->IfcRelConnectsElements		√		Path Connectivity

	Decomposes	IfcObjectDefinition.Decomposes->IfcRelAggregates		√		Object Definition Composition
	Object Identifier	IfcRoot.GlobalId->IfcGloballyUniqueId	√			Software Identity
	Object Name	IfcRoot.Name->IfcLable	√			Object User Identity
	Object Type	IfcController.PredefinedType->IfcControllerTypeEnum	√			BACnet Controllor Predefined Type; Object Predefined Type
	Description	IfcRoot.Description->IfcText			√	-
	Minimun value	Pset_ControllerTypeFloating.Range ->IfcPropertyBoundedValue.LowerBoundValue::IfcReal	√			PropertySets for Objects
	Average Value	Pset_ControllerTypeFloating.Value->IfcReal	√			PropertySets for Objects
	Maximun Value	Pset_ControllerTypeFloating.Range ->IfcPropertyBoundedValue.UpperBoundValue::IfcReal	√			PropertySets for Objects
	Attempted Samples	Pset_BACnetAveragingObject.AttemptedSamples ->IfcInteger	√			PropertySets for Objects
	Valid Samples	Pset_BACnetAveragingObject.ValidSamples ->IfcInteger	√			PropertySets for Objects
	Object Property Reference	IfcProduct.ReferencedBy ->IfcRelAssignsToProduct.RelatedObject->IfcObject IfcObject.IsDefinedBy->IfcRelDefinesByProperties.RelatedPropertyDefinition -> IfcPropertySet	√			Produce assignment and Property Sets for Objects

	Window Interval	Pset_BACnetAveragingObject.WindowInterval ->IfcTimeMeasure	√			PropertySets for Objects
	Window Sample	Pset_BACnetAveragingObject.WindowSamples ->IfcInteger	√			PropertySets for Objects
	Property List	IfcPropertySet	√			PropertySets for Objects
	Tags	IfcElement.Tag->IfcIdentifier			√	Element Occurrence Attributes
BACnet Binary Input Object Type		IfcController				
	Object Geometric Representation	IfcProduct.Representation->IfcProductDefinitionShape		√		Box Geometry; FootPrint Annotation Geometry
	Object Placement	IfcProduct.ObjectPlacement->IfcObjectPlacement		√		Product local placement
	Object Contained in Space	IfcElement.ContainedInStructure ->IfcRelContainedInSpatialStructure		√		Spatial Containment
	Object Connected From	IfcElement.ConnectedFrom->IfcRelConnectsElements		√		Path Connectivity
	Decomposes	IfcObjectDefinition.Decompose->IfcRelAggregates		√		Object Definition Composition
	Object Identifier	IfcRoot.GlobalId->IfcGloballyUniqueId	√			Software Identity
	Object Name	IfcRoot.Name->IfcLable	√			Object User Identity
	Object Type	IfcController.PredefinedType->IfcControllerTypeEnum	√			BACnet Controllor Predefined Type; Object Predefined Type

	Present Value	Pset_ControllerTypeMultiPosition.Value->IfcInteger	√			PropertySets for Objects
	Description	IfcRoot.Description->IfcText			√	-
	Status Flags	Pset_BACnetBinaryInputObject.StatusFlags ->IfcLabel:: PEnum_BACnetStatusFlag	√			PropertySets for Objects
	Event State	Pset_BACnetBinaryInputObject.EventState ->IfcLabel:: PEnum_BACnetEventState	√			PropertySets for Objects
	Out of Service	Pset_BACnetBinaryInputObject.OutOfService ->IfcBoolean	√	`		PropertySets for Objects
	Polarity	Pset_BACnetBinaryInputObject.EventState ->IfcLabel:: PEnum_BACnetPolarity	√			PropertySets for Objects
	Property List	IfcPropertySet	√			PropertySets for Objects
	Tags	IfcElement.Tag->IfcIdentifier			√	Element Occurrence Attributes
BACnet Binary Output Object Type		IfcController				
	Object Geometric Representation	IfcProduct.Representation::IfcProductDefinitionShape		√		Box Geometry; FootPrint Annotation Geometry
	Object Placement	IfcProduct.ObjectPlacement::IfcObjectPlacement		√		Product local placement
	Object Contained in Space	IfcElement.ContainedInStructure::IfcRelContainedInSpatialStructure		√		Spatial Containment

	Object Connected From	IfcElement.ConnectedFrom::IfcRelConnectsElements		√		Path Connectivity
	Decomposes	IfcObjectDefinition.Decomposes::IfcRelAggregates		√		Object Definition Composition
	Object Identifier	IfcRoot.GlobalId::IfcGloballyUniqueId	√			Software Identity
	Object Name	IfcRoot.Name::IfcLable	√			Object User Identity
	Object Type	IfcController.PredefinedType::IfcControllerTypeEnum	√			BACnet Controllor Predefined Type; Object Predefined Type
	Present Value	Pset_ControllerTypeMultiPosition.Value::IfcInteger	√			PropertySets for Objects
	Description	IfcRoot.Description::IfcText			√	-
	Status Flags	Pset_BACnetBinaryOutputObject.StatusFlags ->IfcLabel:: PEnum_BACnetStatusFlag	√			PropertySets for Objects
	Event State	Pset_BACnetBinaryOutputObject.EventState ->IfcLabel:: PEnum_BACnetEventState	√			PropertySets for Objects
	Out of Service	Pset_BACnetBinaryOutputObject.OutOfService ->IfcBoolean	√	`		PropertySets for Objects
	Polarity	Pset_BACnetBinaryOutputObject.EventState ->IfcLabel:: PEnum_BACnetPolarity	√			PropertySets for Objects
	Priority Array	Pset_BACnetBinaryOutputObject.PriorityArray ->IfcLabel:: PEnum_BACnetPriorityValue	√			PropertySets for Objects
	Relinquish Default	Pset_BACnetBinaryOutputObject.RelinquishDefault->IfcReal	√			PropertySets for Objects
	Property List	IfcPropertySet.HasPropertites->IfcProperty	√			PropertySets for Objects

	Current Command Priority	Pset_BACnetBinaryOutputObject.CurrentCommandPriority->IfcInteger	√			PropertySets for Objects
	Tags	IfcElement.Tag->IfcIdentifier			√	Element Occurrence Attributes
BACnet Binary Value Object Type		IfcController				
	Object Geometric Representation	IfcProduct.Representation->IfcProductDefinitionShape		√		Box Geometry; FootPrint Annotation Geometry
	Object Placement	IfcProduct.ObjectPlacement->IfcObjectPlacement		√		Product local placement
	Object Contained in Space	IfcElement.ContainedInStructure ->IfcRelContainedInSpatialStructure		√		Spatial Containment
	Object Connected From	IfcElement.ConnectedFrom->IfcRelConnectsElements		√		Path Connectivity
	Decomposes	IfcObjectDefinition.Decomposes->IfcRelAggregates		√		Object Definition Composition
	Object Identifier	IfcRoot.GlobalId->IfcGloballyUniqueId	√			Software Identity
	Object Name	IfcRoot.Name->IfcLable	√			Object User Identity
	Object Type	IfcController.PredefinedType->IfcControllerTypeEnum	√			Controllor Predefined Type; Object Predefined Type
	Present Value	Pset_ControllerTypeMultiPosition.Value->IfcInteger	√			PropertySets for Objects

	Description	IfcRoot.Description->IfcText			√	-
	Status Flags	Pset_BACnetBinaryValueObject.StatusFlags ->IfcLabel:: PEnum_BACnetStatusFlag	√			PropertySets for Objects
	Event State	Pset_BACnetBinaryValueObject.EventState ->IfcLabel:: PEnum_BACnetEventState	√			PropertySets for Objects
	Out of Service	Pset_BACnetBinaryValueObject.OutOfService ->IfcBoolean	√	`		PropertySets for Objects
	Property List	IfcPropertySet	√			PropertySets for Objects
	Tags	IfcElement.Tag->IfcIdentifier			√	Element Occurrence Attributes
BACnet Calender Object Type		IfcControllor assgined to IfcWorkCalender				
	Has Assignment	IfcObjectDefinition.HasAssignment->IfcRelAssignsToControl	√			BACnet-Controller Assign to Work Calender
	Object Geometric Representation	IfcProduct.Representation->IfcProductDefinitionShape		√		Box Geometry; FootPrint Annotation Geometry
	Object Placement	IfcProduct.ObjectPlacement->IfcObjectPlacement		√		Product local placement
	Object Contained in Space	IfcElement.ContainedInStructure ->IfcRelContainedInSpatialStructure		√		Spatial Containment
	Object Connected From	IfcElement.ConnectedFrom->IfcRelConnectsElements		√		Path Connectivity
	Decomposes	IfcObjectDefinition.Decomposes->IfcRelAggregates		√		Object Definition Composition

	Object Identifier	IfcRoot.GlobalId->IfcGloballyUniqueId	√			Software Identity
	Object Name	IfcRoot.Name->IfcLable	√			Object User Identity
	Object Type	IfcController.PredefinedType->IfcControllerTypeEnum	√			BACnet Controllor Predefined Type; Object Predefined Type
	Present Value	Pset_ControllerTypeTwoPosition.Value->IfcBoolean	√			PropertySets for Objects
	Description	IfcRoot.Description->IfcText			√	-
	Date List	IfcWorkCalender.WorkingTime->IfcWorkTime	√			BACnet Controllor assign to Work Calender
	Property List	IfcPropertySet	√			PropertySets for Objects
	Tags	IfcElement.Tag->IfcIdentifier			√	Element Occurrence Attributes
BACnet Command Object Type		IfcProcedure assigned to IfcControllor				
	Has Assignment	IfcProcedure.HasAssignments ->IfcRelAssignsToControl.RelatedObjects ->IfcObject	√			BACnet-Procedure Assign to Controller
	Decomposes	IfcObjectDefinition.Decomposes->IfcRelAggregates		√		Object Definition Composition
	Object Name	IfcRoot.Name->IfcLable	√			Software Identity
	Object Idenfifier	IfcRoot.GlobalId->IfcGloballyUniqueId	√			Object User Identity
	Object Type	IfcProcedure.PredefinedType->IfcProcedureTypeEnum	√			BACnet-Procedure Predefined Type; Object Predefined Type
	Present Value	Pset-BACnetCommandObject.PresentValue->IfcInteger	√			PropertySets for Objects

	Description	IfcRoot.Description->IfcText			√	
	In Process	Pset-BACnetCommandObject.InProcess->IfcBoolean	√			PropertySets for Objects
	All Writes Successful	Pset-BACnetCommandObject.AllWritesSuccessful->IfcBoolean	√			PropertySets for Objects
	Action	Pset-BACnetCommandObject.Action->IfcInteger	√			PropertySets for Objects
	Property List	IfcPropertySet	√			PropertySets for Objects
BACnet Device Object Type		IfcController				
	Object Geometric Representation	IfcProduct.Representation->IfcProductDefinitionShape		√		Box Geometry; FootPrint Annotation Geometry
	Object Placement	IfcProduct.ObjectPlacement->IfcObjectPlacement		√		Product local placement
	Object Contained in Space	IfcElement.ContainedInStructure ->IfcRelContainedInSpatialStructure		√		Spatial Containment
	Object Connected From	IfcElement.ConnectedFrom->IfcRelConnectsElements		√		Path Connectivity
	Object Quantity	IfcObject.IsDefinedBy->IfcElementQuantity		√		Quantity Set
	Object Identifier	IfcRoot.GlobalId->IfcGloballyUniqueId	√			Software Identity
	Object Name	IfcRoot.Name->IfcLable	√			Object User Identity

	Object Type	IfcController.PredefinedType->IfcControllerTypeEnum	√			BACnet Controllor Predefined Type; Object Predefined Type
	System Status	Pset_BACnetDeviceObject.SystemStatus ->IfcLabel::Penum_BACnetDeviceStatus	√			PropertySets for Objects
	Vendor Name	Pset_ManufacturereTypeInformation.Manufactuerer->IfcLabel	√			PropertySets for Objects
	Vendor Identifier	Pset_BACnetDeviceObject.VenderIdentifier ->IfcIdentifier	√			PropertySets for Objects
	Model Name	Pset_ManufacturereTypeInformation.ModelLabel::IfcLabel	√			PropertySets for Objects
	Firmware Revision	Pset_ControllorTypeProgrammable.FirmwareVersion::IfcPropertySingleValue::IfcLabel	√			PropertySets for Objects
	Application Software Version	Pset_ControllorTypeProgrammable.SoftwareVersion::IfcPropertySingleValue::IfcLabel	√			PropertySets for Objects
	Protocol Version	Pset_BACnetDeviceObject.ProtocolVersion ->IfcText	√			PropertySets for Objects
	Protocol Revision	Pset_BACnetDeviceObject.ProtocolRevision ->IfcReal	√			PropertySets for Objects
	Protocol Service Supported	Pset_BACnetDeviceObject.ProtocolServiceSupported ->IfcLabel::PEnum_BACnetServicesSupported	√			PropertySets for Objects
	Protocol Object Type Supported	Pset_BACnetDeviceObject.SystemStatus ->IfcLabel::PEnum_BACnetObjectTypesSupported				PropertySets for Objects
	Object List	IfcObjectDefinition.IsdecomposedBy->IfcRelAggregates	√			Object Definition Decomposition

	Max APDU Length Accepted	Pset_BACnetDeviceObject.MaxAPDULengthAccepted ->IfcReal	√			PropertySets for Objects
	Segmentation Supported	Pset_BACnetDeviceObject.SegmentationSupported ->IfcLabel::PEnum_BACnetSegmentation	√			PropertySets for Objects
	APDU Timeout	Pset_BACnetDeviceObject.APDUTimeout ->IfcTimeMeasure	√			PropertySets for Objects
	Number of APDU Retries	Pset_BACnetDeviceObject.NumberOfAPDURetries ->IfcInteger	√			PropertySets for Objects
	Device Address Binding	Pset_BACnetDeviceObject.DeviceAddressBinding ->IfcPostalAddress	√			PropertySets for Objects
	Database Revision	Pset_BACnetDeviceObject.DatabaseRevision ->IfcInteger	√			PropertySets for Objects
	Property List	IfcPropertySet	√			PropertySets for Objects
	Tags	IfcElement.Tag->IfcIdentifier			√	Element Occurrence Attributes

REFERENCES

[1] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, Internet of Things (IoT): A vision, architectural elements, and future directions, Future Generation Computer Systems. 29 (2013) 1645–1660. doi:10.1016/j.future.2013.01.010.

[2] U.S. Energy Information Administration, Commercial buildings energy consumption survey, (2016). https://www.eia.gov/consumption/commercial/data/2012/index.php?view=microdata (accessed September 26, 2019).

[3] J. Bhatt, H.K. Verma, Design and development of wired building automation systems, Energy and Buildings. 103 (2015) 396–413. doi:10.1016/j.enbuild.2015.02.054.

[4] C. Bolchini, A. Geronazzo, E. Quintarelli, Smart buildings: A monitoring and data analysis methodological framework, Building and Environment. 121 (2017) 93–105. doi:10.1016/j.buildenv.2017.05.014.

[5] S. Hu, E. Corry, E. Curry, W.J.N. Turner, J. O'Donnell, Building performance optimisation: A hybrid architecture for the integration of contextual information and time-series data, Automation in Construction. 70 (2016) 51–61. doi:10.1016/j.autcon.2016.05.018.

[6] E. Corry, P. Pauwels, S. Hu, M. Keane, J. O'Donnell, A performance assessment ontology for the environmental and energy management of buildings, Automation in Construction. 57 (2015) 249–259. doi:10.1016/j.autcon.2015.05.002.

[7] A. Kučera, T. Pitner, Semantic BMS: Allowing usage of building automation data in facility benchmarking, Advanced Engineering Informatics. 35 (2018) 69–84. doi:10.1016/j.aei.2018.01.002.

[8] P. Pauwels, S. Zhang, Y.C. Lee, Semantic web technologies in AEC industry: A literature overview, Automation in Construction. 73 (2017) 145–165. doi:10.1016/j.autcon.2016.10.003.

[9] B. Balaji, A. Bhattacharya, G. Fierro, J. Gao, J. Gluck, D. Hong, A. Johansen, J. Koh, J. Ploennigs, Y. Agarwal, M. Bergés, D. Culler, R.K. Gupta, M.B. Kjærgaard, M.

Srivastava, K. Whitehouse, Brick: Metadata schema for portable smart building applications, Applied Energy. 226 (2018) 1273–1292. doi:10.1016/j.apenergy.2018.02.091.

[10] S. Tang, D.R. Shelden, C.M. Eastman, P. Pishdad-Bozorgi, X. Gao, A review of building information modeling (BIM) and the internet of things (IoT) devices integration: Present status and future trends, Automation in Construction. 101 (2019) 127–139. doi:10.1016/j.autcon.2019.01.020.

[11] R. Santos, A.A. Costa, A. Grilo, Bibliometric analysis and review of Building Information Modelling literature published between 2005 and 2015, Automation in Construction. 80 (2017) 118–136. doi:10.1016/j.autcon.2017.03.005.

[12] E. Corry, J. O'Donnell, E. Curry, D. Coakley, P. Pauwels, M. Keane, Using semantic web technologies to access soft AEC data, Advanced Engineering Informatics. 28 (2014) 370–380. doi:10.1016/j.aei.2014.05.002.

[13] Y. Lu, Y. Li, M. Skibniewski, Z. Wu, R. Wang, Y. Le, Information and Communication Technology Applications in Architecture, Engineering, and Construction Organizations: A 15-Year Review, Journal of Management in Engineering. 31 (2014) A4014010. doi:10.1061/(ASCE)ME.1943-5479.0000319.

[14] L. Oppermann, M. Shekow, D. Bicer, Mobile cross-media visualisations made from building information modelling data, in: Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct - MobileHCI '16, ACM, Florence, Italy, 2016: pp. 823–830. doi:10.1145/2957265.2961852.

[15] A. Genty, Virtual Reality for the construction industry, The CALLISTO-SARI project, benefits for BOUYGUES CONSTRUCTION., in: Proceedings of the 2015 Virtual Reality International Conference on ZZZ - VRIC '15, ACM, Laval, France, 2015: pp. 1–7. doi:10.1145/2806173.2806185.

[16] R. Kuenzel, J. Teizer, M. Mueller, A. Blickle, SmartSite: Intelligent and autonomous environments, machinery, and processes to realize smart road construction projects, Automation in Construction. 71 (2016) 21–33. doi:10.1016/j.autcon.2016.03.012.

[17] G.G. Lee, J. Cho, S. Ham, T. Lee, G.G. Lee, S.H. Yun, H.J. Yang, A BIM- and sensor-based tower crane navigation system for blind lifts, Automation in Construction. 26 (2012) 1–10. doi:10.1016/j.autcon.2012.05.002.

[18] Y. Fang, Y.K. Cho, F. Druso, J. Seo, Assessment of operator's situation awareness for smart operation of mobile cranes, Automation in Construction. 85 (2018) 65–75. doi:10.1016/j.autcon.2017.10.007.

[19] J.W. Park, J. Chen, Y.K. Cho, Self-corrective knowledge-based hybrid tracking system using BIM and multimodal sensors, Advanced Engineering Informatics. 32 (2017) 126–138. doi:10.1016/j.aei.2017.02.001.

[20] L.Y. Ding, C. Zhou, Q.X. Deng, H.B. Luo, X.W. Ye, Y.Q. Ni, P. Guo, Real-time safety early warning system for cross passage construction in Yangtze Riverbed Metro Tunnel based on the internet of things, Automation in Construction. 36 (2013) 25–37. doi:10.1016/j.autcon.2013.08.017.

[21] M. Arslan, Z. Riaz, A.K. Kiani, S. Azhar, Real-time environmental monitoring, visualization and notification system for construction H&S management, Journal of Information Technology in Construction. 19 (2014) 72–91. http://www.itcon.org/2014/4 (accessed October 23, 2017).

[22] Z. Riaz, M. Arslan, A.K. Kiani, S. Azhar, CoSMoS: A BIM and wireless sensor based integrated solution for worker safety in confined spaces, Automation in Construction. 45 (2014) 96–106. doi:10.1016/j.autcon.2014.05.010.

[23] E.O. Ibem, S. Laryea, Survey of digital technologies in procurement of construction projects, Automation in Construction. 46 (2014) 11–21. doi:10.1016/j.autcon.2014.07.003.

[24] K. Ding, H. Shi, J. Hui, Y. Liu, B. Zhu, F. Zhang, W. Cao, Smart Steel Bridge Construction Enabled By BIM and Internet of Things in Industry 4.0: A Framework, in: 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC), IEEE, Zhuhai, China, 2018: pp. 1–5. doi:10.1109/ICNSC.2018.8361339.

[25] S. Meža, Ž. Turk, M. Dolenc, Component based engineering of a mobile BIM-based augmented reality system, Automation in Construction. 42 (2014) 1–12. doi:10.1016/j.autcon.2014.02.011.

[26] J. Matthews, P.E.D. Love, S. Heinemann, R. Chandler, C. Rumsey, O. Olatunj, Real time progress management: Re-engineering processes for cloud-based BIM in construction, Automation in Construction. 58 (2015) 38–47. doi:10.1016/j.autcon.2015.07.004.

[27] P.R. Zekavat, S. Moon, L.E. Bernold, Performance of short and long range wireless communication technologies in construction, Automation in Construction. 47 (2014) 50–61. doi:10.1016/j.autcon.2014.07.008.

[28] M.J. Skibniewski, Information technology applications in construction safety assurance, Journal of Civil Engineering and Management. 20 (2014) 778–794. doi:10.3846/13923730.2014.987693.

[29] J. Teizer, T. Cheng, Y. Fang, Location tracking and data visualization technology to advance construction ironworkers' education and training in safety and productivity, Automation in Construction. 35 (2013) 53–68. doi:10.1016/j.autcon.2013.03.004.

[30] H. Li, M. Lu, G. Chan, M. Skitmore, Proactive training system for safe and efficient precast installation, Automation in Construction. 49 (2015) 163–174. doi:10.1016/j.autcon.2014.10.010.

[31] K. Kim, Y. Cho, S. Zhang, Integrating work sequences and temporary structures into safety planning: Automated scaffolding-related safety hazard identification and prevention in BIM, Automation in Construction. 70 (2016) 128–142. doi:10.1016/j.autcon.2016.06.012.

[32] X. Yuan, C.J. Anumba, M.K. Parfitt, Cyber-physical systems for temporary structure monitoring, Automation in Construction. 66 (2016) 1–14. doi:10.1016/j.autcon.2016.02.005.

[33] Y. Zhang, L. Bai, Rapid structural condition assessment using radio frequency identification (RFID) based wireless strain sensor, Automation in Construction. 54 (2015) 1–11. doi:10.1016/j.autcon.2015.02.013.

[34] R. Kanan, O. Elhassan, R. Bensalem, An IoT-based autonomous system for workers' safety in construction sites with real-time alarming, monitoring, and positioning strategies, Automation in Construction. 88 (2018) 73–86. doi:10.1016/j.autcon.2017.12.033.

[35] P. Vähä, T. Heikkilä, P. Kilpeläinen, M. Järviluoma, E. Gambao, Extending automation of building construction - Survey on potential sensor technologies and robotic applications, Automation in Construction. 36 (2013) 168–178. doi:10.1016/j.autcon.2013.08.002.

[36] R.Y. Zhong, Y. Peng, F. Xue, J. Fang, W. Zou, H. Luo, S. Thomas Ng, W. Lu, G.Q.P.

Shen, G.Q. Huang, Prefabricated construction enabled by the Internet-of-Things, Automation in Construction. 76 (2017) 59–70. doi:10.1016/j.autcon.2017.01.006.

[37] J. Ikonen, A. Knutas, H. Hämäläinen, M. Ihonen, J. Porras, T. Kallonen, Use of embedded RFID tags in concrete element supply chains, Journal of Information Technology in Construction. 18 (2013) 119–147. http://www.itcon.org/2013/7 (accessed October 23, 2017).

[38] Y. Niu, W. Lu, K. Chen, G.G. Huang, C. Anumba, Smart Construction Objects, Journal of Computing in Civil Engineering. (2015) 04015070. doi:10.1061/(ASCE)CP.1943-5487.0000550.

[39] Y. Niu, W. Lu, D. Liu, K. Chen, C. Anumba, G.G. Huang, An SCO-Enabled Logistics and Supply Chain–Management System in Construction, Journal of Construction Engineering and Management. 143 (2017) 04016103. doi:10.1061/(ASCE)CO.1943-7862.0001232.

[40] G. Xu, M. Li, C.-H. Chen, Y. Wei, Cloud asset-enabled integrated IoT platform for lean prefabricated construction, Automation in Construction. 93 (2018) 123–134. doi:https://doi.org/10.1016/j.autcon.2018.05.012.

[41] R.Y. Zhong, Y. Peng, F. Xue, J. Fang, W. Zou, H. Luo, S. Thomas Ng, W. Lu, G.Q.P. Shen, G.Q. Huang, Prefabricated construction enabled by the Internet-of-Things, Automation in Construction. 76 (2017) 59–70. doi:10.1016/j.autcon.2017.01.006.

[42] C.Z. Li, F. Xue, X. Li, J. Hong, G.Q. Shen, An Internet of Things-enabled BIM platform for on-site assembly services in prefabricated construction, Automation in Construction. 89 (2018) 146–161. doi:10.1016/j.autcon.2018.01.001.

[43] J. Teizer, M. Wolf, O. Golovina, M. Perschewski, M. Propach, M. Neges, M. König, Internet of Things (IoT) for Integrating Environmental and Localization Data in Building Information Modeling (BIM), in: Proceedings of the 34th International Symposium on Automation and Robotics in Construction, IAARC Publications, Taipei, Taiwan, 2017: pp. 603–609. doi:10.22260/ISARC2017/0084.

[44] B. Dave, S. Kubler, K. Främling, L. Koskela, Opportunities for enhanced lean construction management using Internet of Things standards, Automation in Construction. 61 (2016) 86–97. doi:10.1016/j.autcon.2015.10.009.

[45] A. Tezel, Z. Aziz, From Conventional To It Based Visual Management: A

Conceptual Discussion For Lean Construction, Journal of Information Technology in Construction. 22 (2017) 220–246. http://www.itcon.org/2017/12 (accessed October 23, 2017).

[46] M. Pouke, J.P. Virtanen, M. Badri, T. Ojala, Comparison of two workflows for Web-based 3D smart home visualizations, in: 2018 IEEE International Conference on Future IoT Technologies (Future IoT), IEEE, Eger, Hungary, 2018: pp. 1–8. doi:10.1109/FIOT.2018.8325599.

[47] B. Becerik-Gerber, F. Jazizadeh, N. Li, G. Calis, Application Areas and Data Requirements for BIM-Enabled Facilities Management, Journal of Construction Engineering and Management. 138 (2012) 431–442. doi:10.1061/(ASCE)CO.1943-7862.0000433.

[48] K. Chen, W. Lu, Y. Peng, S. Rowlinson, G.Q. Huang, Bridging BIM and building: From a literature review to an integrated conceptual framework, International Journal of Project Management. 33 (2015) 1405–1416. doi:10.1016/j.ijproman.2015.03.006.

[49] W. Shen, Q. Hao, Y. Xue, A loosely coupled system integration approach for decision support in facility management and maintenance, Automation in Construction. 25 (2012) 41–48. doi:10.1016/j.autcon.2012.04.003.

[50] A.H. Oti, E. Kurul, F. Cheung, J.H.M. Tah, A framework for the utilization of Building Management System data in building information models for building design and operation, Automation in Construction. 72 (2016) 195–210. doi:10.1016/j.autcon.2016.08.043.

[51] E.A. Pärn, D.J. Edwards, Conceptualising the FinDD API plug-in: A study of BIM-FM integration, Automation in Construction. 80 (2017) 11–21. doi:10.1016/j.autcon.2017.03.015.

[52] G. Williams, M. Gheisari, P.-J. Chen, J. Irizarry, BIM2MAR: An Efficient BIM Translation to Mobile Augmented Reality Applications, Journal of Management in Engineering. 31 (2013) A4014009. doi:10.1061/(ASCE)ME.1943-5479.0000315.

[53] T.W. Kang, H.S. Choi, BIM perspective definition metadata for interworking facility management data, Advanced Engineering Informatics. 29 (2013) 958–970. doi:10.1016/j.aei.2015.09.004.

[54] A. Motamedi, A. Hammad, Y. Asen, Knowledge-assisted BIM-based visual analytics for failure root cause detection in facilities management, Automation in Construction. 43 (2014) 73–83. doi:10.1016/j.autcon.2014.03.012.

[55] F. Petrushevski, Personalized lighting control based on a space model, in: Proceedings of the 2012 ACM Conference on Ubiquitous Computing, ACM, Pittsburgh, Pennsylvania, USA, 2012: p. 568. doi:10.1145/2370216.2370311.

[56] J. Lee, Y. Jeong, Y.S. Oh, J.C. Lee, N. Ahn, J. Lee, S.H. Yoon, An integrated approach to intelligent urban facilities management for real-time emergency response, Automation in Construction. 30 (2013) 256–264. doi:10.1016/j.autcon.2012.11.008.

[57] T. Gerrish, K. Ruikar, M. Cook, M. Johnson, M. Phillip, C. Lowry, BIM application to building energy performance visualisation and management: Challenges and potential, Energy and Buildings. 144 (2017) 218–228. doi:10.1016/j.enbuild.2017.03.032.

[58] M. Marzouk, A. Abdelaty, Monitoring thermal comfort in subways using building information modeling, Energy and Buildings. 84 (2014) 252–257. doi:10.1016/j.enbuild.2014.08.006.

[59] S. Habibi, Micro-climatization and real-time digitalization effects on energy efficiency based on user behavior, Building and Environment. 114 (2017) 410–428. doi:10.1016/j.buildenv.2016.12.039.

[60] J. O'Donnell, E. Corry, S. Hasan, M. Keane, E. Curry, Building performance optimization using cross-domain scenario modeling, Linked data, And complex event processing, Building and Environment. 62 (2013) 102–111. doi:10.1016/j.buildenv.2013.01.019.

[61] M. Dibley, H. Li, Y. Rezgui, J. Miles, An ontology framework for intelligent sensor-based building monitoring, Automation in Construction. 28 (2012) 1–14. doi:10.1016/j.autcon.2012.05.018.

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

[63] B. Dave, A. Buda, A. Nurminen, K. Främling, A framework for integrating BIM and IoT through open standards, Automation in Construction. 95 (2018) 35–45. doi:10.1016/j.autcon.2018.07.022.

[64] K. McGlinn, B. Yuce, H. Wicaksono, S. Howell, Y. Rezgui, Usability evaluation of a web-based tool for supporting holistic building energy management, Automation in Construction. 84 (2017). doi:10.1016/j.autcon.2017.08.033.

[65] N.M. Tomašević, M. Batić, L.M. Blanes, M.M. Keane, S. Vraneš, Ontology-based facility data model for energy management, Advanced Engineering Informatics. 29 (2015) 971–984. doi:10.1016/j.aei.2015.09.003.

[66] F. Radulovic, M. Poveda-Villalón, D. Vila-Suero, V. Rodríguez-Doncel, R. García-Castro, A. Gómez-Pérez, Guidelines for Linked Data generation and publication: An example in building energy consumption, Automation in Construction. 57 (2015) 178–187. doi:10.1016/j.autcon.2015.04.002.

[67] S.A. Kim, D. Shin, Y. Choe, T. Seibert, S.P. Walz, Integrated energy monitoring and visualization system for Smart Green City development: Designing a spatial information integrated energy monitoring model in the context of massive data management on a web based platform, Automation in Construction. 22 (2012) 51–59. doi:10.1016/j.autcon.2011.07.004.

[68] E. Patti, A. Mollame, D. Erba, D. Dalmasso, A. Osello, E. Macii, A. Acquaviva, Combining Building Information Modelling and Ambient Data in Interactive Virtual and Augmented Reality Environments, IT Professional. (2017) 1–1. doi:10.1109/MITP.2017.265104553.

[69] S. Peng, G. Su, J. Chen, P. Du, Design of an IoT-BIM-GIS Based Risk Management System for Hospital Basic Operation, in: 2017 IEEE Symposium on Service-Oriented System Engineering (SOSE), IEEE, San Francisco, CA, USA, 2017: pp. 69–74. doi:10.1109/SOSE.2017.22.

[70] J.-H. Woo, M.A. Peterson, B. Gleason, Developing a Virtual Campus Model in an Interactive Game-Engine Environment for Building Energy Benchmarking, Journal of Computing in Civil Engineering. 30 (2016) C4016005. doi:10.1061/(ASCE)CP.1943-5487.0000600.

[71] H. Ufuk Gökçe, K. Umut Gökçe, Integrated System Platform for Energy Efficient Building Operations, Journal of Computing in Civil Engineering. 28 (2014) 05014005. doi:10.1061/(ASCE)CP.1943-5487.0000288.

[72] A.H. Kazmi, M.J. O'grady, D.T. Delaney, A.G. Ruzzelli, G.M.P. O'hare, A Review of Wireless-Sensor-Network-Enabled Building Energy Management Systems, ACM Transactions on Sensor Networks (TOSN). 10 (2014) 66:1--66:43. doi:10.1145/2532644.

[73] A. Heller, C. Orthmann, Wireless technologies for the construction sector - Requirements, energy and cost efficiencies, Energy and Buildings. 73 (2014) 212–216. doi:10.1016/j.enbuild.2013.12.019.

[74] N. Li, B. Becerik-Gerber, B. Krishnamachari, L. Soibelman, A BIM centered indoor localization algorithm to support building fire emergency response operations, Automation in Construction. 42 (2014) 78–89. doi:10.1016/j.autcon.2014.02.019.

[75] J. Cho, G. Lee, S. Lee, An automated direction setting algorithm for a smart exit sign, Automation in Construction. 59 (2015) 139–148. doi:10.1016/j.autcon.2015.05.004.

[76] H. Yoon, R. Shiftehfar, S. Cho, B.F. Spencer, M.E. Nelson, G. Agha, Victim Localization and Assessment System for Emergency Responders, Journal of Computing in Civil Engineering. 30 (2016) 1–14. doi:10.1061/(ASCE)CP.1943-5487.0000483.

[77] A.Y. Chen, J.C. Chu, TDVRP and BIM Integrated Approach for In-Building Emergency Rescue Routing, Journal of Computing in Civil Engineering. 30 (2016) C4015003. doi:10.1061/(ASCE)CP.1943-5487.0000522.

[78] M.Y. Cheng, K.C. Chiu, Y.M. Hsieh, I.T. Yang, J.S. Chou, Y.W. Wu, BIM integrated smart monitoring technique for building fire prevention and disaster relief, Automation in Construction. 84 (2017) 14–30. doi:10.1016/j.autcon.2017.08.027.

[79] P. Pishdad-Bozorgi, X. Gao, C. Eastman, A.P. Self, Planning and developing facility management-enabled building information model (FM-enabled BIM), Automation in Construction. 87 (2018) 22–38. doi:10.1016/j.autcon.2017.12.004.

[80] R. Srinivasan, C. Kibert, S. Thakur, I. Ahmed, P. Fishwick, Z. Ezzell, J. Lakshmanan, Preliminary research in dynamic-BIM (D-BIM) Workbench development, in: Proceedings of the Winter Simulation Conference, Winter Simulation Conference, Berlin, Germany, 2012: p. 53. doi:10.1109/WSC.2012.6465331.

[81] B. Dong, Z. O'Neill, Z. Li, A BIM-enabled information infrastructure for building

energy Fault Detection and Diagnostics, Automation in Construction. 44 (2014) 197–211. doi:10.1016/j.autcon.2014.04.007.

[82] R. Bille, S.P. Smith, K. Maund, G. Brewer, Extending Building Information Models into Game Engines, in: K. Blackmore, K. Nesbitt, S. Smith (Eds.), Proceedings of the 2014 Conference on Interactive Entertainment - IE2014, ACM, Newcastle, NSW, Australia, 2014: pp. 1–8. doi:10.1145/2677758.2677764.

[83] W. Solihin, C. Eastman, Y.-C. Lee, D.-H. Yang, A simplified relational database schema for transformation of BIM data into a query-efficient and spatially enabled database, Automation in Construction. 84 (2017) 367–383. doi:10.1016/j.autcon.2017.10.002.

[84] A. Khalili, D.K.H. Chua, IFC-Based Graph Data Model for Topological Queries on Building Elements, Journal of Computing in Civil Engineering. 29 (2013) 04014046. doi:10.1061/(ASCE)CP.1943-5487.0000331.

[85] W. Mazairac, J. Beetz, BIMQL - An open query language for building information models, Advanced Engineering Informatics. 27 (2013) 444–456. doi:10.1016/j.aei.2013.06.001.

[86] M. Alves, P. Carreira, A.A. Costa, BIMSL: A generic approach to the integration of building information models with real-time sensor data, Automation in Construction. 84 (2017) 304–314. doi:https://doi.org/10.1016/j.autcon.2017.09.005.

[87] World Wide Web Consortium (W3C), Semantic Web - W3C, (2015). https://www.w3.org/standards/semanticweb/ (accessed October 1, 2019).

[88] P. Pauwels, W. Terkaj, EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology, Automation in Construction. 63 (2016) 100–133. doi:10.1016/j.autcon.2015.12.003.

[89] T. Berners-Lee, Linked Data - Design Issues, (2006). https://www.w3.org/DesignIssues/LinkedData.html (accessed October 1, 2019).

[90] V. Charpenay, S. Kabisch, D. Anicic, H. Kosch, An ontology design pattern for IoT device tagging systems, in: Proceedings - 2015 5th International Conference on the Internet of Things, IoT 2015, Institute of Electrical and Electronics Engineers Inc., 2015: pp. 138–145. doi:10.1109/IOT.2015.7356558.

[91] Project Haystack, Home – Project Haystack, (n.d.). https://project-haystack.org/ (accessed October 23, 2018).

[92] H. Burak Gunay, W. Shen, G. Newsham, Data analytics to improve building performance: A critical review, Automation in Construction. 97 (2019) 96–109. doi:10.1016/j.autcon.2018.10.020.

[93] ASHRAE, ASHRAE's BACnet Committee, Project Haystack and Brick Schema Collaborating to Provide Unified Data Semantic Modeling Solution | ashrae.org, (2018). https://www.ashrae.org/about/news/2018/ashrae-s-bacnet-committee-project-haystack-and-brick-schema-collaborating-to-provide-unified-data-semantic-modeling-solution (accessed October 7, 2019).

[94] L. Daniele, F. den Hartog, J. Roes, Created in Close Interaction with the Industry: The Smart Appliances REFerence (SAREF) Ontology, in: Lecture Notes in Business Information Processing, Springer Verlag, 2015: pp. 100–112. doi:10.1007/978-3-319-21545-7_9.

[95] M. Compton, P. Barnaghi, L. Bermudez, R. García-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, V. Huang, K. Janowicz, W.D. Kelsey, D. Le Phuoc, L. Lefort, M. Leggieri, H. Neuhaus, A. Nikolov, K. Page, A. Passant, A. Sheth, K. Taylor, The SSN ontology of the W3C semantic sensor network incubator group, Journal of Web Semantics. 17 (2012) 25–32. doi:10.1016/j.websem.2012.05.003.

[96] buildingSMART International, IFC Overview summary, (2019). https://technical.buildingsmart.org/standards/ifc/ (accessed June 19, 2018).

[97] P. Pauwels, K. McGlinn, S. Törmä, J. Beetz, Linked Data, in: Building Information Modeling, Springer International Publishing, Cham, 2018: pp. 181–197. doi:10.1007/978-3-319-92862-3_10.

[98] J. Bender, BACnet Ontology — BACnet Ontology, (2013). http://bacowl.sourceforge.net/index.html (accessed October 8, 2019).

[99] L. Dodds, I. Davis, Linked data patterns, (2012). http://patterns.dataincubator.org/book/ (accessed October 17, 2019).

[100] Resources - BrickSchema, (2019). https://brickschema.org/resources (accessed March 10, 2020).

[101] M.H. Rasmussen, P. Pauwels, M. Lefrançois, G.F. Schneider, Building Topology Ontology, (2020). https://w3c-lbd-cg.github.io/bot/ (accessed March 10, 2020).

[102] M. Bonduel, J. Oraskari, P. Pauwels, M. Vergauwen, R. Klein, The IFC to linked building data converter - Current status, in: CEUR Workshop Proceedings, 2018: pp. 34–43. https://github.com/jyrkioraskari/IFCtoRDF-Desktop (accessed November 4, 2019).

[103] N.V. Hoang, S. Torma, Implementation and Experiments with an IFC-to-Linked Data Converter, in: The 32nd CIB W78 Conference 2015, Eindhoven, Netherlands, 2015: pp. 285–294. http://drumbeat.cs.hut.fi/owl (accessed October 17, 2019).

[104] B. Balaji, A. Bhattacharya, G. Fierro, J. Gao, J. Gluck, D. Hong, A. Johansen, J. Koh, J. Ploennigs, Y. Agarwal, M. Berges, D. Culler, R. Gupta, M.B. Kjærgaard, M. Srivastava, K. Whitehouse, Brick: Towards a unified metadata schema for buildings, in: Proceedings of the 3rd ACM Conference on Systems for Energy-Efficient Built Environments, BuildSys 2016, ACM Press, New York, New York, USA, 2016: pp. 41–50. doi:10.1145/2993422.2993577.

[105] A. Beveridge, J.B. Hansen, J. Val, G. Leif, F. Roisin, K. Sunil, R. Tomas, Validata: RDF Validator, (2015). https://www.w3.org/2015/03/ShExValidata/ (accessed March 10, 2020).

[106] World Wide Web Consortium (W3C), RDF Validation Service, (2007). https://www.w3.org/RDF/Validator/documentation (accessed March 10, 2020).

[107] S. Tang, D.R. Shelden, C.M. Eastman, P. Pishdad-Bozorgi, X. Gao, BIM assisted Building Automation System information exchange using BACnet and IFC, Automation in Construction. 110 (2020) 103049. doi:10.1016/j.autcon.2019.103049.

[108] R. Yang, L. Wang, Multi-objective optimization for decision-making of energy and comfort management in building automation and control, Sustainable Cities and Society. 2 (2012) 1–7. doi:10.1016/j.scs.2011.09.001.

[109] K. Afsari, C.M. Eastman, D. Castro-Lacouture, JavaScript Object Notation (JSON) data serialization for IFC schema in web-based BIM data exchange, Automation in Construction. 77 (2017) 24–51. doi:10.1016/j.autcon.2017.01.011.

[110] X. Gao, P. Pishdad-Bozorgi, BIM-enabled facilities operation and maintenance: A review, Advanced Engineering Informatics. 39 (2019) 227–247.

doi:10.1016/j.aei.2019.01.005.

[111] H. Dibowski, J. Ploennigs, K. Kabitzsch, Automated design of building automation systems, IEEE Transactions on Industrial Electronics. 57 (2010) 3606–3613. doi:10.1109/TIE.2009.2032209.

[112] J. Zhang, B.-C. Seet, T. Lie, Building Information Modelling for Smart Built Environments, Buildings. 5 (2015) 100–115. doi:10.3390/buildings5010100.

[113] A.H. Oti, E. Kurul, F. Cheung, J. Tah, The Utilization of BMS in BIM for Facility Management, CIB World Building Congress. (2016) 224–235. doi:10.13140/RG.2.2.28538.49601.

[114] S. Cataldi, BIM and BACnet : Intertwining Stories, BACnet Europe Journal. (2017) 37–38. http://www.bacnet.org/Bibliography/2016-11-Cataldi-BIM.pdf (accessed February 28, 2018).

[115] American Society of Heating Refrigerating and Air-Conditioning Engineers, ANSI/ASHRAE Standard 135-2016: BACnet A Data Communivation Protocol for Building Automation and Control Networks, (2016). https://www.ashrae.org/technical-resources/bookstore/bacnet (accessed March 22, 2018).

[116] H.M. Newman, BACnet Explained, Part One, BACnet Today and Smart Grid A Supplement to ASHRAE Journal. 55 (2013) B2–B7. http://www.bacnet.org/Bibliography/BACnet-Today-13/Newman-2013.pdf (accessed March 22, 2018).

[117] V. Lappalainen, K.P. Vtt, Web Service based integration of building information systems, in: Clima 2007 WellBeing Indoors, VTT, Finland, 2007. http://www.w3.org/2002/ws/ (accessed June 5, 2019).

[118] J.K.W. Wong, J. Ge, S.X. He, Digitisation in facilities management: A literature review and future research directions, Automation in Construction. 92 (2018) 312–326. doi:10.1016/j.autcon.2018.04.006.

[119] buildingSMART International, Model View Definition Summary, (2019). http://www.buildingsmart-tech.org/specifications/ifc-view-definition (accessed June 19, 2018).

[120] R. See, J. Karlshoej, D. Davis, An Integrated Process for Delivering IFC Based Data Exchange, (2012). http://iug.buildingsmart.org/idms/methods-and-guides/Integrated_IDM-MVD_ProcessFormats_14.pdf/view (accessed March 19, 2018).

[121] C.M. Eastman, Y.-S. Jeong, ; R Sacks, I. Kaner, Exchange Model and Exchange Object Concepts for Implementation of National BIM Standards, Journal of Computing in Civil Engineering. 24 (2010) 25–34. doi:10.1061/(ASCE)0887-3801(2010)24:1(25).

[122] M. Tardif, buildingSMART alliance Construction Operations Building information exchange (COBie) Project - National Institute of Building Sciences, (2008). https://www.nibs.org/page/bsa_cobie (accessed June 5, 2019).

[123] buildingSMART International, Model View Definition IFC4 Construction Operation Building Information Exchange, (2013). http://docs.buildingsmartalliance.org/MVD_COBIE/ (accessed June 5, 2019).

[124] Y. Arayici, T. Fernando, V. Munoz, M. Bassanino, Interoperability specification development for integrated BIM use in performance based design, Automation in Construction. 85 (2018) 167–181. doi:10.1016/j.autcon.2017.10.018.

[125] buildingSmart International, ifcDoc Tool Summary, (2019). https://technical.buildingsmart.org/resources/ifcdoc/ (accessed June 19, 2018).

[126] S. Pinheiro, R. Wimmer, J. O'Donnell, S. Muhic, V. Bazjanac, T. Maile, J. Frisch, C. van Treeck, MVD based Information Exchange between BIM and Building Energy Performance Simulation, Automation in Construction. 90 (2018) 91–103. doi:10.1016/j.autcon.2018.02.009.

[127] A. Andriamamonjy, D. Saelens, R. Klein, An automated IFC-based workflow for building energy performance simulation with Modelica, Automation in Construction. 91 (2018) 166–181. doi:10.1016/j.autcon.2018.03.019.

[128] M. Venugopal, C. Eastman, R. Sacks, J. Teizer, Improving the robustness of model exchanges using product modeling "concepts" for IFC schema, Computing in Civil Engineering. (2011) 760–767. doi:10.1061/41182(416)75.

[129] Y.C. Lee, C.M. Eastman, J.K. Lee, Validations for ensuring the interoperability of data exchange of a building information model, Automation in Construction. 58

(2015) 176–195. doi:10.1016/j.autcon.2015.07.010.

[130] G. Lee, Y.H. Park, S. Ham, Extended Process to Product Modeling (xPPM) for integrated and seamless IDM and MVD development, Advanced Engineering Informatics. 27 (2013) 636–651. doi:10.1016/j.aei.2013.08.004.

[131] J. Won, G. Lee, C. Cho, No-Schema Algorithm for Extracting a Partial Model from an IFC Instance Model, Journal of Computing in Civil Engineering. 27 (2013) 585–592. doi:10.1061/(ASCE)CP.1943-5487.0000320.

[132] G. Calleja-Rodriguez, R. Guruz, M.-C. Geicler, R. Steinmann, K. Linhard, G. Dangl, Collaboration requirements and interoperability fundamentals in BIM based multi-disciplinary building design processes, in: Proceedings of the 11th European Conference on Product and Process Modelling, ECPPM 2016, Limassol, Cyprus, 2016: pp. 349–354. https://www.researchgate.net/publication/310450142 (accessed October 30, 2018).

[133] E. El Asmi, S. Robert, D. Mazza, K. Zreik, Multi-physical Model Simulation and Interoperability through BIM / IFC using COMETH Simulation Engine using COMETH Simulation Engine, in: 14th International Conference on the Advances in Design Sciences and Technology, Nice, France, 2014: pp. 1–12. https://www.researchgate.net/publication/273106212 (accessed October 30, 2018).

[134] buildingSMART International, IFC4 Add2 Release, BuildingSMART International. (2019). https://standards.buildingsmart.org/IFC/DEV/IFC4_2/FINAL/HTML/ (accessed June 27, 2018).

[135] buildingSMART International, Information Delivery Manual (IDM), (2019). https://technical.buildingsmart.org/resources/information-delivery-manual/ (accessed September 17, 2018).

[136] OMG, BPMN Specification - Business Process Model and Notation, (2017). http://www.bpmn.org/ (accessed September 10, 2018).

[137] OCCS Development Committe Secretariat, OmniClass, (2017). http://www.omniclass.org/ (accessed May 14, 2019).

[138] buildingSMART International, ifcDoc Baselines, (2019). https://standards.buildingsmart.org/tools/ifcdoc/IFC4_ADD2.ifcdoc (accessed October 16, 2018).

[139] buildingSMART International, ifcDoc, (2019). https://technical.buildingsmart.org/resources/ifcdoc/ (accessed October 23, 2018).

[140] buildingSMART International, IFC Formats - buildingSMART Technical, (2019). https://technical.buildingsmart.org/standards/ifc/ifc-formats/ (accessed June 11, 2019).

[141] P. Pauwels, S. Törmä, J. Beetz, M. Weise, T. Liebich, Linked Data in Architecture and Construction, Automation in Construction. 57 (2015) 175–177. doi:10.1016/j.autcon.2015.06.007.

[142] K. Afsari, C.M. Eastman, D. Castro-Lacouture, JavaScript Object Notation (JSON) data serialization for IFC schema in web-based BIM data exchange, Automation in Construction. 77 (2017) 24–51. doi:10.1016/j.autcon.2017.01.011.

[143] buildingSMART International, IFCBUILDINGCONTROLSDOMAIN, (2019). http://www.buildingsmart-tech.org/ifc/IFC2x4/rc1/html/ifcbuildingcontrolsdomain/ifcbuildingcontrolsdomain.htm (accessed October 23, 2018).

[144] I. Bill Swan, Alerton Technologies, The Language of BACnet-Objects, Properties and Services, (1995). http://www.bacnet.org/Bibliography/ES-7-96/ES-7-96.htm (accessed March 14, 2018).

[145] Y.C. Lee, W. Solihin, C.M. Eastman, The Mechanism and Challenges of Validating a Building Information Model regarding data exchange standards, Automation in Construction. 100 (2019) 118–128. doi:10.1016/j.autcon.2018.12.025.

[146] X. Gao, P.-B. Pardis, D.R. Shelden, S. Tang, A Scalable Cyber-Physical System Data Acquisition Framework for the Smart Built Environment, in: Computing in Civil Engineering 2019, 2019: pp. 259–266. doi:doi:10.1061/9780784482445.033.

[147] Y.C. Lee, C.M. Eastman, W. Solihin, R. See, Modularized rule-based validation of a BIM model pertaining to model views, Automation in Construction. 63 (2016) 1–11. doi:10.1016/j.autcon.2015.11.006.

[148] Y.C. Lee, C.M. Eastman, J.K. Lee, Validations for ensuring the interoperability of data exchange of a building information model, Automation in Construction. 58 (2015) 176–195. doi:10.1016/j.autcon.2015.07.010.

[149] S. Tang, D.R. Shelden, C.M. Eastman, P. Pishdad-Bozorgi, X. Gao, A review of building information modeling (BIM) and the internet of things (IoT) devices integration: Present status and future trends, Automation in Construction. 101 (2019) 127–139. doi:10.1016/j.autcon.2019.01.020.

[150] X. Gao, S. Tang, P. Pishdad-Bozorgi, D.R. Shelden, Foundational Research in Integrated Building Internet of Things (IoT) Data Standards Research Report, Atlanta, GA, 2018. doi:10.13140/RG.2.2.17378.99525.

[151] Brick TTL Viewer, (2019). https://viewer.brickschema.org/ (accessed March 13, 2020).

[152] J. Barrasa, NSMNTX - Neo4j RDF & Semantics toolkit - Neo4j Graph Database Platform, (2019). https://neo4j.com/labs/nsmtx-rdf/ (accessed March 13, 2020).

[153] British Standard Institution, PAS 1192-3:2014 Specification for information management for the operational phase of assets using building information modelling, (2014). https://shop.bsigroup.com/ProductDetail/?pid=000000000030311237.

[154] H.K. Grani, Asset Information Model, PAS 1192-3 and the ISO road ahead, (2016) 1–15. https://blog.areo.io/asset-information-model/ (accessed March 13, 2020).

[155] M. Holten, G. Ferdinand, BOT : the Building Topology Ontology of the W3C Linked Building Data Group, (1900). https://www.microsoft.com/microsoft-365/ (accessed November 4, 2019).

[156] M.U. Khalid, M.K. Bashir, D. Newport, Development of a Building Information Modelling (BIM)-Based Real-Time Data Integration System Using a Building Management System (BMS), in: M. Dasbaz, C. Gorse, A. Moncaster (Eds.), Building Information Modelling, Building Performance, Design and Smart Construction, Springer, Cham, 2017: pp. 93–104. doi:10.1007/978-3-319-50346-2_7.

[157] U. Isikdag, Enhanced Building Information Models Using IoT Services and Integration Patterns, Springer International Publishing, 2015. doi:10.1007/978-3-319-21825-0.

[158] G. Desogus, E. Quaquero, A. Sanna, G. Gatto, L.C. Tagliabue, S. Rinaldi, A.L.C. Ciribini, G. Di Giuda, V. Villa, Preliminary performance monitoring plan for energy

retrofit: A cognitive building: The "Mandolesi Pavillon" at the University of Cagliari, in: 2017 AEIT International Annual Conference, IEEE, Cagliari, Italy, 2017: pp. 1–6. doi:10.23919/AEIT.2017.8240529.

[159] M.F. Alam, S. Katsikas, O. Beltramello, S. Hadjiefthymiades, Augmented and virtual reality based monitoring and safety system: A prototype IoT platform, Journal of Network and Computer Applications. 89 (2016) 109–119. doi:10.1016/j.jnca.2017.03.022.

[160] J. Yang, M.W. Park, P.A. Vela, M. Golparvar-Fard, Construction performance monitoring via still images, time-lapse photos, and video streams: Now, tomorrow, and the future, Advanced Engineering Informatics. 29 (2015) 211–224. doi:10.1016/j.aei.2015.01.011.

[161] T. Omar, M.L. Nehdi, Data acquisition technologies for construction progress tracking, Automation in Construction. 70 (2016) 143–155. doi:10.1016/j.autcon.2016.06.016.

[162] D.Y. Kim, S.A. Kim, An exploratory model on the usability of a prototyping-process for designing of Smart Building Envelopes, Automation in Construction. 81 (2017) 389–400. doi:10.1016/j.autcon.2017.03.012.

[163] M. Heidari, E. Allameh, B. De Vries, H. Timmermans, J. Jessurun, F. Mozaffar, Smart-BIM virtual prototype implementation, Automation in Construction. 39 (2014) 134–144. doi:10.1016/j.autcon.2013.07.004.

[164] Z. Pourzolfaghar, M. Helfert, Integration of Buildings Information with Live Data from IoT Devices, in: Z. Mahmood (Ed.), Connected Environments for the Internet of Things, Springer, Cham, 2017: pp. 169–185. doi:10.1007/978-3-319-70102-8_9.

[165] A. Hefnawy, A. Bouras, C. Cherifi, IoT for Smart City Services: Lifecycle Approach, in: D. Boubiche, D. Hidoussi, L. Guezouli, A. Bounceur, H. Cruz (Eds.), Proceedings of the International Conference on Internet of Things and Cloud Computing - ICC '16, ACM, Cambridge, UK, 2016: pp. 1–9. doi:10.1145/2896387.2896440.

[166] H.M. Chen, K.C. Chang, T.H. Lin, A cloud-based system framework for performing online viewing, storage, and analysis on big data of massive BIMs, Automation in Construction. 71 (2016) 34–48. doi:10.1016/j.autcon.2016.03.002.

[167] A. Botta, W. De Donato, V. Persico, A. Pescapé, Integration of Cloud computing and Internet of Things: A survey, Future Generation Computer Systems. 56 (2016) 684–700. doi:10.1016/j.future.2015.09.021.

[168] T.K.L. Hui, R.S. Sherratt, D.D. Sánchez, Major requirements for building Smart Homes in Smart Cities based on Internet of Things technologies, Future Generation Computer Systems. 76 (2017) 358–369. doi:10.1016/j.future.2016.10.026.

[169] P. Pauwels, T.M. de Farias, C. Zhang, A. Roxin, J. Beetz, J. De Roo, Querying and reasoning over large scale building data sets, in: Proceedings of the International Workshop on Semantic Big Data - SBD '16, ACM, San Francisco, California, USA, 2016: pp. 1–6. doi:10.1145/2928294.2928303.

[170] S. Peng, G. Su, J. Chen, P. Du, Design of an IoT-BIM-GIS Based Risk Management System for Hospital Basic Operation, in: Proceedings - 11th IEEE International Symposium on Service-Oriented System Engineering, SOSE 2017, 2017: pp. 69–74. doi:10.1109/SOSE.2017.22.

[171] H. Wang, Sensing Information Modelling for Smart City, in: 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity), IEEE, Chengdu, China, 2015: pp. 40–45. doi:10.1109/SmartCity.2015.44.

[172] I. Shahrour, L. Alileche, A. Alfurjani, Smart cities: System and tools used for the digital modelling of physical urban systems, in: 2017 Sensors Networks Smart and Emerging Technologies (SENSET), IEEE, Beirut, Lebanon, 2017: pp. 1–4. doi:10.1109/SENSET.2017.8125056.

[173] R. Woodhead, P. Stephenson, D. Morrey, Digital construction : From point solutions to IoT ecosystem, Automation in Construction. 93 (2018) 35–46. doi:10.1016/j.autcon.2018.05.004.

[174] M. Bilal, L.O. Oyedele, J. Qadir, K. Munir, S.O. Ajayi, O.O. Akinade, H.A. Owolabi, H.A. Alaka, M. Pasha, Big Data in the construction industry: A review of present status, opportunities, and future trends, Advanced Engineering Informatics. 30 (2016) 500–521. doi:10.1016/j.aei.2016.07.001.

[175] S. Howell, Y. Rezgui, T. Beach, Integrating building and urban semantics to empower smart water solutions, Automation in Construction. 81 (2017) 434–448. doi:10.1016/j.autcon.2017.02.004.

[176] M. Hastak, C. Koo, Theory of an Intelligent Planning Unit for the Complex Built Environment, Journal of Management in Engineering. 33 (2016). doi:10.1061/(ASCE)ME.1943-5479.0000486.

[177] Y. Peng, J.-R. Lin, J.-P. Zhang, Z.-Z. Hu, A hybrid data mining approach on BIM-based building operation and maintenance, Building and Environment. (2017). doi:10.1016/j.buildenv.2017.09.030.

[178] A.P. Plageras, K.E. Psannis, C. Stergiou, H. Wang, B.B. Gupta, Efficient IoT-based sensor BIG Data collection–processing and analysis in smart buildings, Future Generation Computer Systems. 82 (2018) 349–357. doi:10.1016/j.future.2017.09.082.

[179] S. Rowland, BIM to IoT: The persistence problem, in: C. de Carvalho, P. Escudeiro, A. Coelho (Eds.), Serious Games, Interaction, and Simulation. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Springer, Cham, 2016: pp. 127–137. doi:10.1007/978-3-319-29060-7_19.