IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received July 28, 2020, accepted September 3, 2020, date of publication September 14, 2020, date of current version September 25, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3024066

Big Data Management and Analytics Metamodel for IoT-Enabled Smart Buildings

MUHAMMAD RIZWAN BASHIR[1], **ASIF QUMER GILL**[2], **GHASSAN BEYDOUN**[1], **AND BRAD MCCUSKER**[3]

[1]Faculty of Engineering and Information Technology, School of Information, Systems, and Modelling, University of Technology Sydney, Ultimo, NSW 2007, Australia
[2]Faculty of Engineering and Information Technology, School of Computer Science, University of Technology Sydney, Ultimo, NSW 2007, Australia
[3]SURROUND Australia Pty Ltd., Sydney, NSW 2000, Australia

Corresponding author: Muhammad Rizwan Bashir (rizwan.bashir@student.uts.edu.au)

This work was supported in part by the Australian Government through the Research Training Program (RTP), and in part by the SURROUND Australia Pty Ltd.

ABSTRACT Big data management and analytics, in the context of IoT (Internet of Things)-enabled smart buildings, is a challenging task. It is a diffused and complex area of knowledge due to the diversity of IoT devices and the nature of data generated by the IoT devices. Many international bodies have developed metamodels for IoT-enabled ecosystems to allow knowledge sharing. However, these are often narrow in focus and deal with only the IoT aspects without taking into account the management and analytics of big data generated by the IoT devices. Hence, in this article we propose a metamodel for the Integrated Big Data Management and Analytics (IBDMA) framework for IoT-enabled smart buildings. The IBDMA Metamodel can be used to facilitate interoperability between existing big data management and analytics ecosystems deployed in smart buildings or other smart environments. We import the metamodel into a knowledge graph management tool and by considering a case study we validate the metamodel using this tool. The evaluation results demonstrate that IBDMA Metamodel is indeed suitable for its intended purpose.

INDEX TERMS IoT, big data management, metamodel, smart buildings.

I. INTRODUCTION

Big data management and analytics for IoT-enabled smart buildings involve a high level of complexity and rely on different sources of knowledge distributed across time, space and people. Hence, in this article, we advocate the development of an integrated big data management and analytics (IBDMA) metamodel for IoT-enabled smart buildings. This enables IoT and big data practitioners to address the big data challenges in IoT enabled big data ecosystems. The metamodel is part of the Integrated Big Data Management and Analytics (IBDMA) framework. The IBDMA framework has two parts: 1) Reference Architecture; 2) Metamodel. The reference architecture has been submitted for publication in another journal. The scope of this article is limited to the metamodel development of the IBDMA framework. The paper aims to use the generic representational layer (the metamodel) to provide a unified view of common concepts and actions that apply in various IoT-enabled ecosystems. The IBDMA metamodel developed will provide a set of generic concepts useful to an IoT-enabled ecosystem, while not necessarily providing all required details required by every single specific facility within the ecosystem on hand. Some details are hidden behind the general concept we use, and we leave them to each individual user to extend based on their specific problem within the IoT-enabled big data ecosystem.

This research was initiated in [1], where we illustrated the big data pipeline for IoT enabled smart building. Metamodeling has been endorsed by the efforts of the Object Management Group (OMG) [2]. We use it in our work to integrate existing attempts to represent IoT and big data knowledge in a reusable form and to provide an integrated and unified point of access. We illustrate our unification approach. We present the result and validation of the metamodel which generalizes most of the concepts used in existing IoT and big data practices as described in existing relevant architectures and models. The rest of this

The associate editor coordinating the review of this manuscript and approving it for publication was Shree Krishna Sharma.

article is organized as follows: Section II provides some background and related works to this research. Section III provides the details on the five key elements of the Integrated Big Data Management and Analytics (IBDMA) Framework. Section IV provides the reference architecture of the IBDMA Framework using a smart building use case. Section V provides the details on the metamodel development process and enlists details on the validation of the IBDMA metamodel presenting three practical effective case studies for a smart building. Section VI lists the major contributions and limitations of the presented metamodel. Finally, Section VII concludes this article with a discussion of possible future extensions of this research. Due to the limitation on the paper size, we reduced the size of some figures to fit the margins. The high resolution figures are provided at the GitHub repository [3].

II. RESEARCH BACKGROUND AND RELATED WORK

A metamodeling process generally aims to create a collection of classes to describe domain concepts to represent domain entities, actions or states. This collection of concepts is the metamodel. A metamodel also contains the specification of modelling environment for certain domain and defines the syntax and the semantics of the domain. It can be viewed from three different perspectives: i) as a set of building blocks and rules used to build new models, ii) as a model of a domain of interest and iii) as an instance of another model. In our context, a metamodel is a fundamental building block that defines the concepts and the relationships between those concepts in the IoT enabled big data ecosystem [4].

Various metamodels have been developed by researchers so the stakeholders can better understand the IoT enabled ecosystems. In [5], the authors present a metamodeling framework for designing smart cyber-physical environments. Their framework provides a common vocabulary to model applications by exploiting concepts and relationships between concepts specific to the smart environment domains. Moreover, a set of general guidelines was presented to drive the analysis, the design and the implementation of smart environments. This article, however, only provides a generic high-level vocabulary of concepts without providing any specific use cases for the smart buildings.

In [6], the authors present a meta-model that enables the extraction of valuable knowledge and deep insights from the Big Data. To achieve this, their paper proposes a meta-model for two layers related to Big Data: Data Sources and Ingestion. While, this work is important, however, this highlights the need for a more comprehensive metamodel to provide support across the Big Data lifecycle stages such storage, analysis and visualization in the context of smart buildings.

In [7], the authors introduce a metamodel-based approach for IoT systems development. They discuss three particular levels linked to the analysis, design and implementation phases of smart objects metamodel. Their stated purpose is to provide a seamless support among the different phases of smart objects development process. However, their paper does not actually address the big data management and analytics challenges for the effective management of the smart buildings.

In [8], the authors present the data landscape metamodel, which helps organizations express their challenges and solutions with regards to gathering value out of data. However, a detailed and comprehensive metamodel is missing that covers the "data" and other elements of smart buildings and their interactions.

In [9], the authors propose a new approach for software design of a smart building system that involves design and process metamodels. This approach provides a common vocabulary for smart building concepts, attributes, and the relationship between concepts. It also provides the ability to formalize safety properties and functions of the components in a smart building. This allows users an increase in the effectiveness of software development by embedding a domain knowledge in the metamodel. However, this article lacks the big data management and analytics aspects of the smart buildings.

In [10], the authors propose the use of UML (Unified Modelling Language) standard for modelling the big data extract process at a conceptual level with the use of new specific stereotypes proposed by the UML deployment diagrams and other using the approach of the ETL (Extract, Transform, Load) process in data warehouses. The paper presents case studies based on three tools used in the extraction process, Sqoop, Flume and Data Click. However, this article lacks the metamodel that addresses the data management issues in the smart buildings.

In [11], the authors analyze IoT use into manufacturing, its foundation principles, available elements and technologies for the man-things-software communication already developed in this area. The paper proposes an architecture for IoT applied to the industry, a metamodel of integration (IoT, Social Networks, Cloud and Industry 4.0) for generation of applications for the Industry 4.0, and the manufacturing monitoring prototype implemented with the Raspberry Pi microcomputer, a cloud storage server and a mobile device for controlling an online production process. This article, however, lacks the high-level architecture as well as the metamodel for the data management and analysis of IoT data in smart environments.

In [12], the authors propose an extension of the Smart Environments Metamodel (SEM) framework for the development of a smart office application devoted to recognize and predict some simple workers' activities. However, the applicability of the presented metamodel is very limited and lacks the necessary concepts and relationships that could be re-used or scaled up for other smart environments. Moreover, aspects of big data management and analytics are also missing from the paper.

In [13], the authors introduce a general semi-structured metamodel (GSMM) based on the use of a generic graph that can be instantiated to a concrete data model. This is prescribed through providing values for a restricted set of parameters and some high-level constraints, themselves represented as graphs. The metamodel aims to evaluate, integrate and access data models in a uniform way. Although this article provides a foundation for the understanding of the metamodel concepts, however, it is very generic in its representation and lacks its effectiveness for the smart building and big data domains.

In [14], the authors present a new approach which leverages semantic models and rules to enable selective data filtering to sending the cloud. They propose the use of Platform Independent Model, based on semantic web technologies to facilitate sharing and reusing semantic rules in IoT gateways. They also propose a platform specific model which encompasses a set of rules and concepts that match the specific features and functionalities of sensor nodes to perform data filtering. However, that paper lacks the explanation on the complete end-to-end data processing workflow from sensing of environment to controlling the environment. It also does not focus on the big data management and analytics challenges in the context of smart buildings.

In [15], the authors describe a metamodel-based approach that enables a data scientist to different data models to an enterprise data model using UML class diagrams, the UML Profile mechanism, OCL, and prescribed model transformations. An executable data mapper for Enterprise data management transfers and consolidates data from operational information systems into an enterprise database. However, that paper does not present a metamodel which could be used to address the big data management and analytics challenges for smart buildings.

In [16], the authors provide a survey of IoT, Cloud Computing, Big Data and Sensors with the aim to find their common operations and integrating them. New data collection methods are proposed for smart building which could result in efficient energy management of smart buildings. However, it fails to present the architecture and metamodel for big data management and analytics of smart buildings.

In [17], the authors presented the design and implementation of a low-cost occupancy detection system. To reduce the energy consumption of the HVAC system. However, this article focuses only on one aspect of the smart building and lacks to provide the reference architecture and a metamodel which can be used by the researchers and practitioners to address the big data challenges in smart buildings.

In summary, it can be observed from the literature review and related work analysis that there is growing interest among community in the topic of IoT, Big Data, Smart Building and their metamodel. Although the prior studies provide a good foundation for the understanding of a generic metamodel development process and usage. However, none present a consolidated and comprehensive framework which provides both a reference architecture and a metamodel to address the challenges associated with the big management and analytics in the context of smart buildings. A number of very high level and generic metamodels exist. But before nay can be tailored to a specific use case, the metamodel needs to be well tested and validated before it is deemed fit. This work bypasses that need by providing a comprehensive metamodel specifically targeted for the big data management and analytics of smart buildings. We earlier developed the reference architecture as part of the IBDMA framework and submitted it for publication in another journal, but that research did not include the development of the IBDMA Metamodel. This is the focus of this article. Hence to set the context, we first present the contextual elements of the IBDMA framework and its reference architecture before discussing the IBDMA metamodel.

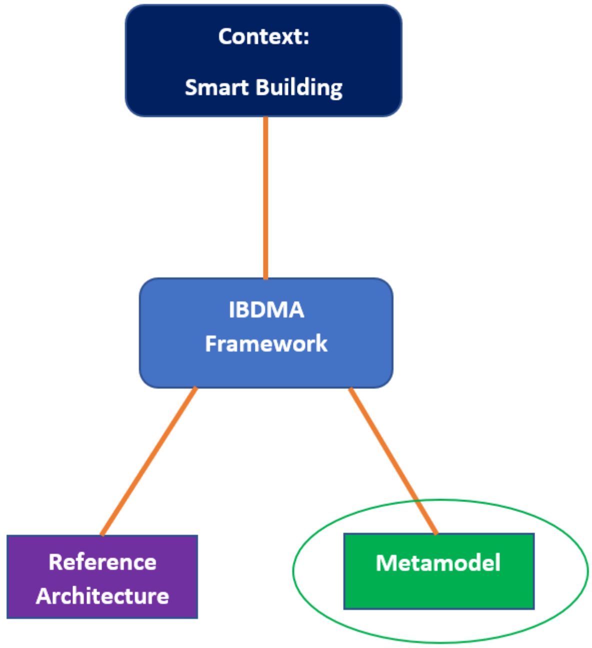

FIGURE 1. IBDMA framework.

III. IBDMA FRAMEWORK

The IBDMA framework is composed of two parts. The first is the reference architecture and the second is the metamodel as shown in Figure 1. The reference architecture has been developed prior to this article and has been submitted for publication. In this article, we present the metamodel as encircled in Figure 1. The context for the IBDMA Framework is Smart Building with a view to improve residents' comfort and safety. The IBDMA framework enables researchers, big data architects, IoT professionals and data engineers to manage and analyze big data generated from IoT devices.

The IBDMA Framework (as discussed here [18], [19]) has five key contextual elements named; People, Process, Technology, Information and Facility as shown in Figure 2.

As shown in Figure 2, the core element of the IBDMA framework is "People" which includes both, 'policy makers and developers' of the IoT ecosystem and 'residents' of the

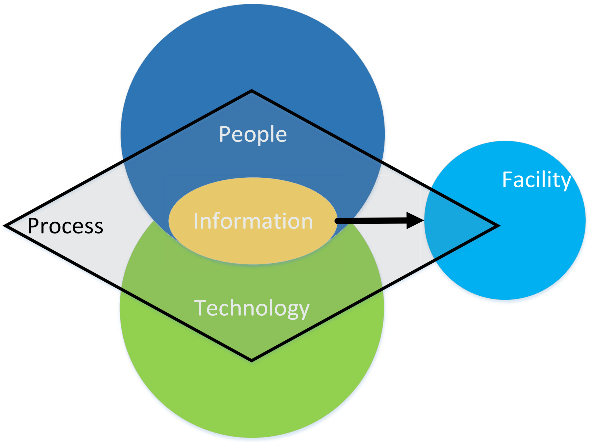

FIGURE 2. IBDMA framework – contextual elements.

smart buildings on the other hand. The 'residents' are the beneficiaries of the IoT enabled smart buildings ecosystem developed in accordance with the policies and requirements defined by the 'policy makers and developers'. Based on the policies and requirements compiled by "People", "Processes" are identified. These "Processes" govern the "Technology" stack to be used for the implementation. The amalgamation of "People", "Processes" and "Technology" results in useful "Information" which ultimately enables us to autonomously manage the smart "Facility". All the elements of the IBDMA framework are linked together by the "Process" element of the framework. These five elements of the IBDMA framework and how they interact with each other are next discussed in detail.

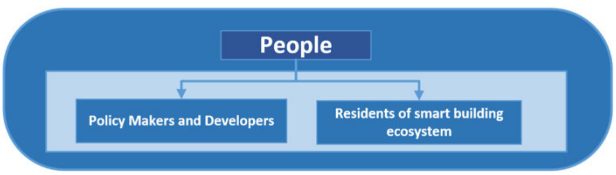

FIGURE 3. First element of IBDMA framework - people.

A. PEOPLE

"People" forms the first element of the IBDMA framework. This element includes the 'policy makers and developers' on one hand of the IoT-enabled smart buildings ecosystem, and the 'residents' of the smart building on the other hand as demonstrated in Figure 3. This is essentially similar to the concept of 'human in the loop' [20]. "People" in the IBDMA framework are involved broadly during two phases of the IoT enabled smart building ecosystem:

- During the initial phase of policy making and requirements compilation (policy makers and developers).
- As beneficiaries of the IoT enabled smart building ecosystem (residents).

During the initial phase, "People" start developing policies and requirements for the design and development of the smart buildings. They highlight the key requirements of the stakeholders and propose and devise an optimized solution meeting the requirements of the beneficiaries as well as the stakeholders. They decide how various aspects of the smart building will work together to make smart buildings more secure and comfortable for the residents of the buildings.

On the beneficiary end, 'people' include the residents of the smart buildings which take advantage of all the efforts of the policy makers, designers and developers of the IoT-enabled smart buildings.

In case of smart buildings, the IBDMA proposes that people define requirements such as what features do, they want to implement in the smart building in order to improve the comfortability of the residents of the building. This may include improved garbage management, improved luminosity levels management, improved parking space management, improved security of the building and so on. Based on these requirements, the "processes" are identified which are used for the successful implementation of the requirements. These "processes" in turn govern the "technology" stack which includes tools and software packages required from the implementation of the "processes" e.g. Apache Flume [21] for data ingestion, Apache Spark [22], [23] for data analysis, Tableau [24] or Microsoft Power BI [25] for data visualization etc. The "process" and "technology" elements of the IBDMA framework are explained in more detail in the next sections.

Once the policy makers and developers (people) have successfully established the goals and requirements of the IoT enabled smart building, 'processes' are identified and implemented to start ingesting IoT data and to perform data analysis on the received data, and that is why 'process' is the second element of the IBDMA framework.

B. PROCESS

The second element of the IBDMA framework is the "Process" which plays a vital role in the overall big data management and analytics strategy. Processes define functionalities and how different functionalities should be integrated to deliver a practical and comprehensive solution. To address big data management and analytics challenges, processes should be transparent and streamlined to have an effective and practical solution.

Based on the policies, requirements and goals defined and identified by the 'people', 'processes' are implemented. Hence, 'people' element serves as the input to the 'process' element as the processes are identified, defined and chosen by 'people' (policy makers and developers). Since we aim for the IoT smart building sensors data for this research, the IBDMA framework proposes the following processes to be part of IoT enabled smart buildings ecosystem which include; monitoring the environment where IoT sensors are deployed, sourcing data from the IoT sensors, ingesting data into a central database, storing the data at a centralized

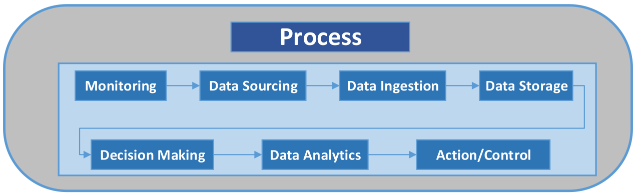

FIGURE 4. Second element of IBDMA framework - process.

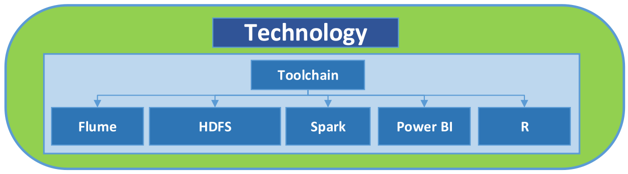

FIGURE 5. Third element of IBDMA framework - technology.

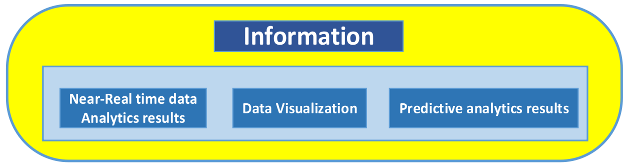

FIGURE 6. Fourth element of IBDMA framework - information.

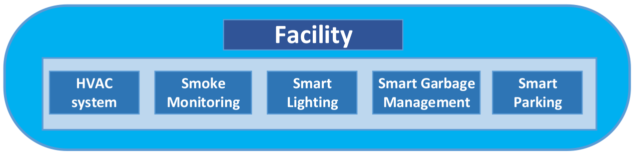

FIGURE 7. Fifth element of IBDMA framework - facility.

location, near-real-time data analytics, decision making, near-real-time visualization and near-real-time autonomous control of the smart facility within the smart building as shown in Figure 4. These facilities may include oxygen levels management, disaster management, garbage management, parking management etc.

Any IoT ecosystem initiates with monitoring the environment, which is achieved by the IoT sensors. There could be a wide variety of IoT sensors that could be deployed in a smart environment depending on the requirements of the users and stakeholders. These sensors 'monitor' various parameters within the environment depending on their specific type. On monitoring the environment, these sensors 'generate/source' digital data. This data is then 'ingested' into a centralized location, so data can be 'stored' and analyzed. The 'analysis' of the data can be useful in many ways. It can be used to obtain useful insights about the smart environment where these sensors are deployed. It can also be used to manage disastrous situations, maintain a comfortable environment for the users, to figure out any faults within the smart environment and to autonomously control various parameters within the smart environment. IBDMA captures and encompasses all these processes under the second element of the IBDMA framework known is the 'process'. Elements in the IBDMA framework are linked by 'Process' as shown in Figure 2. This will become more evident later in Figure 8 where the reference architecture of the framework is presented.

Once the 'people' have identified the 'processes', the underlying 'technology' stack is defined based on the identified 'processes. Successful implementation of processes relies on the selection of appropriate tools and software packages. These tools and software packages fall under "technology" and that is why it is the third element of the IBDMA framework.

C. TECHNOLOGY

The third element of the IBDMA framework is "technology". Defining and choosing the optimal technology is critical to the successful implementation of a big data management and analytics strategy. Based on the "processes" identified by the "people", the "technology" stack is chosen for the successful implementation of the "processes". 'Technology' in IBDMA comprises of the tools and software packages used in the implementation of the IBDMA. Some of the tools we used in the implementation of the IBDMA framework for the smart building data are presented in Figure 5.

For presenting the reference architecture and its implementation, we developed a virtual sensor application. This application was developed in Python using PyCharm [26]; which is a Python IDE for professional developers by JetBrains. This data is then stored in HDFS (Hadoop Distributed File System). HDFS is the primary data storage system used by Hadoop application. It employs a NameNode [27] and DataNode [27] architecture to implement a distributed file system that provides high-performance access to data across highly scalable Hadoop clusters.

Data pipelines for ingesting the virtual sensor data to HDFS are developed and deployed using Apache Flume. Flume is a highly distributed, reliable and configurable tool used to collect, aggregate and transports large amounts of streaming data like log files, events, IoT data etc. from a number of different sources to a centralized data store. It is robust and fault tolerant with tunable reliability mechanisms and many failover and recovery mechanisms. It uses a simple extensible data model that allows for online analytical application.

The sensor data stored in HDFS is analyzed using Apache Spark. Apache Spark is a fast, in-memory data processing engine with elegant and expressive development APIs to allow data workers to efficiently execute streaming, machine learning or SQL workloads that require fast iterative access to datasets [28]. It has Resilient Distributed Dataset (RDD) as its architectural foundation. RDD is a read-only multiset of data items distributed over a cluster of machines,

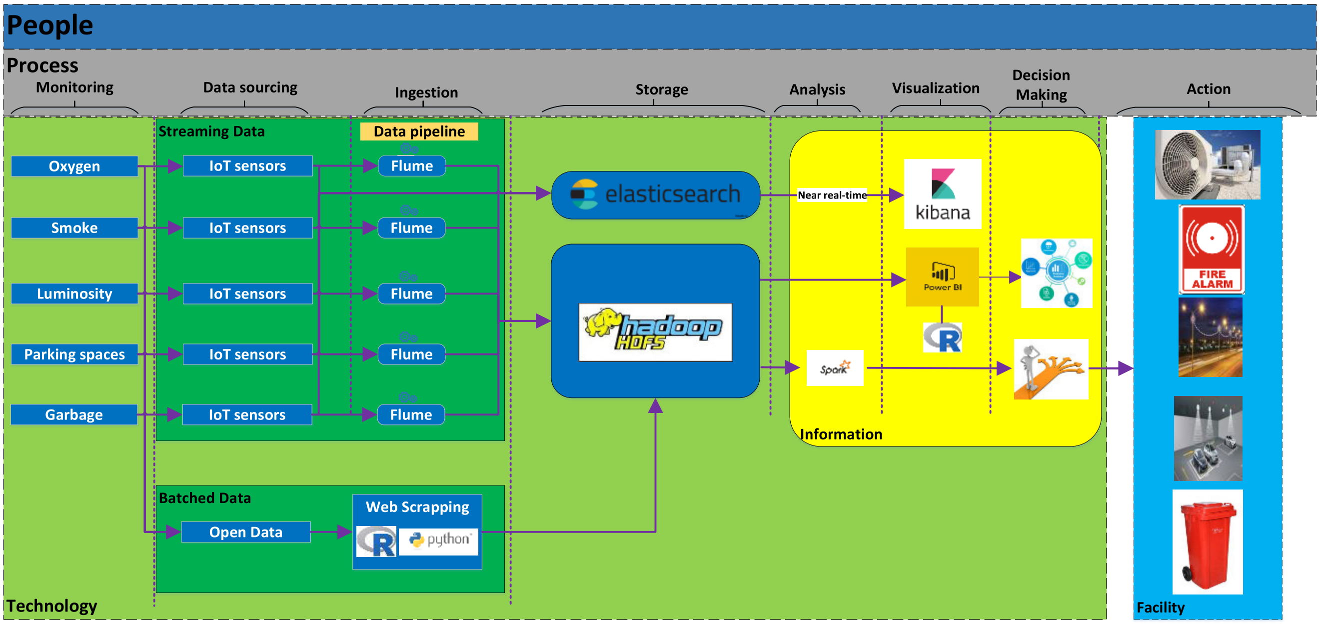

FIGURE 8. IBDMA reference architecture.

that is maintained in a fault tolerant way. Apache Spark analyzes sensor data and enables various controls within the smart building autonomously without any human intervention. This autonomous system keeps a safe, comfortable and healthy environment for the residents of the smart building.

For data visualization of the sensor data stored in HDFS, we used Microsoft Power BI to connect to HDFS data storage, which extracts the data from HDFS and creates dashboards of the data. To perform predictive analytics, we used R integration with Power BI and performed predictive analytics on the sensor data stored in HDFS.

Power BI can only be used for static data visualization, since IoT sensors generate data streams at regular intervals, it is imperative to have near-real time data visualization to have a deeper insight about the environment of the smart building in near-real time, so any hazards can be dealt with in near-real time. To enable near-real time visualization, we used Elasticsearch [29] and Kibana [30]. Data generated from the sensors was stored and indexed in Elasticsearch, and Kibana was then used to visualize it in near-real time by setting up an automatic refresh interval. Elasticsearch is an open-source, RESTful, distributed search and analytics engine built on Apache Lucene [31]. New data can be sent, called documents, to Elasticsearch using the API or ingestion tools such as Logstash [32]. Elasticsearch automatically stores the original document and adds a searchable reference to the document in the cluster's index. We can then search and retrieve the document using the Elasticsearch API [33]. Kibana is an open source data visualization plugin for Elasticsearch. It provides visualization capabilities on top of the content indexed on an Elasticsearch cluster [34].

D. INFORMATION

'Information' is another vital element of the IBDMA which stems from the intersection of "People", "Process" and "Technology". As mentioned in the previous sections, 'people' identify the 'processes' based on the policies and requirements they define for the IoT ecosystem. The 'processes' ultimately govern the underlying 'technology' for the successfully implementation of the 'processes'. On implementing the 'processes' and 'technologies' for the IoT enabled smart buildings solution, useful information in various forms is generated which can not only be used for decision making but can also be used to control various 'facilities' of the smart building autonomously which in-turn benefits the residents (people) of the smart building. The residents are also included in the 'people' element as they are the beneficiary of the whole IoT enabled smart building ecosystem as explained in the previous section.

The "information" can be used to autonomously control various "facilities" in the smart building which may include HVAC system, lighting, garbage, parking, security, elevators, vending machines, water pumps and many more. Moreover, this 'information' also enables the effective management of disasters within the smart building. For example, if the data sent from one of the smoke detection sensor is high enough indicating a fire situation, this 'information' which stems from the intersection of "people", "processes" and "technology" can be used to set up an autonomous system to not only sound an alarm but also to try to eliminate fire by actuating the fire extinguisher in that particular location of the smart building.

"Information" in the IBDMA framework includes data visualization and data analysis results obtained by using the

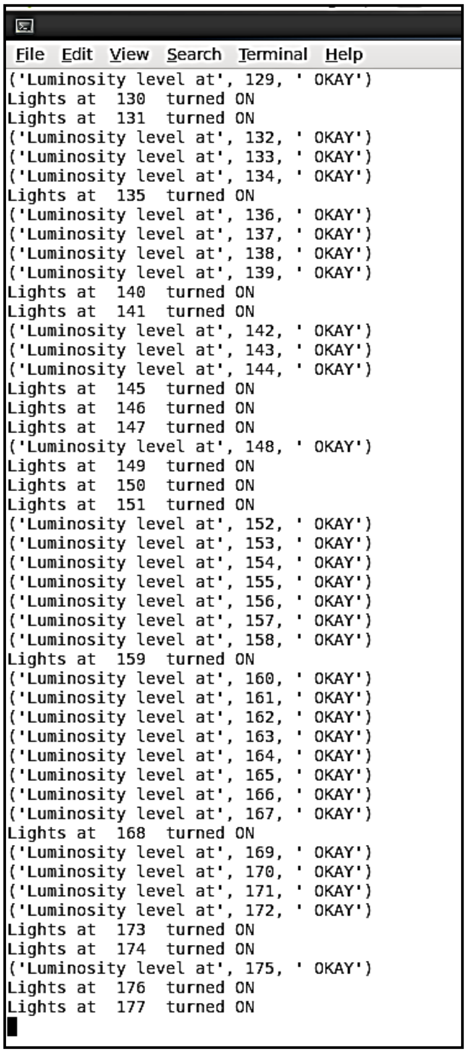

FIGURE 9. Smart building case study - control messages.

'technology' stack as defined in the previous section. For example, the data visualization done in Power BI and Kibana is 'information' based on which certain decisions can be made and certain 'facilities' can be effectively managed. Similarly, the data analysis results obtained by using Apache Spark as discussed in the previous section also serve as 'information'. Some of the key components of "information" as proposed by IBDMA framework are presented in Figure 6. Since "process" element of the IBDMA framework joins and overlaps all other element of the framework, the "processes" joining "information" to other elements of the framework include 'data analysis', 'data visualization' and 'decision making'. This is shown in Figure 8.

TABLE 1. IBDMA metamodel concepts.

IBDMA Concepts	Concepts Definition
People	Represents a human being capable of performing a role in smart building.
Users/residents	Represents the residents of the smart building. For example, a University's smart building will include students, teachers and administrative staff.
Building developers	Represents the developers and architects of the smart building.
Physical Entity	Represents an identifiable object within the physical environment that is of interest to the users/residents for the completion of their goals. Physical Entities may include any object or environment from humans to computers; from electronic appliances daily use items.
Virtual Entity	Provides a digital representation of the Physical Entity. Examples may include, database entries, objects (or instances of a class in an object-oriented programming language), and virtual sensors.
Digital Artefact	Represents the property of virtual entity.
Active Digital Artefact	Represents a running software application or service that may access other services or resources.
Passive Digital Artefact	Represents passive software elements such as database entries that can be digital representations of the physical entity.
Augmented Entity	Represents the concept which enables everyday objects to become part of digital processes, and hence can be regarded as constituting the "thing" in Internet of Things.
Facility	Represents a physical structure or building.
Environment	Represents a particular room, area or environment within the Smart building.
Resource	Represents software components that provide data from or are used in the actuation on Physical Entities.
On-device resource	Represents the resources hosted onto the Devices, that are associated with the Physical Entity. They include executable code for accessing, processing, and storing Sensor information, as well as code for controlling Actuators.
Information	Represents the analysis results obtained after performing the analysis of IoT sensor data.
Data	Represents the data generated by the IoT sensors.
Data Process	Represents various steps in the data analysis process.
Device	Represents a physical object capable of generating or acting on the data. This includes IoT sensors and actuators.
Actuators	Represent devices capable of actuating controls within the smart building.
Sensors	Represent IoT sensors which monitor smart building environment and generate data.
Agent	Agent represents an entity consisting of the grouping of all types of devices i.e. Sensors and Actuators.

E. FACILITY

The fifth and last element of the IBDMA framework is the "facility" which in the context of IoT ecosystem, includes the autonomous smart systems within the smart building to improve the comfort, safety and living conditions for the residents (people) of the building. Some examples of the 'facility' may include; smart energy management, HVAC, smart parking management, smart waste management, smart lighting, etc. as shown in Figure 7. To address the big data

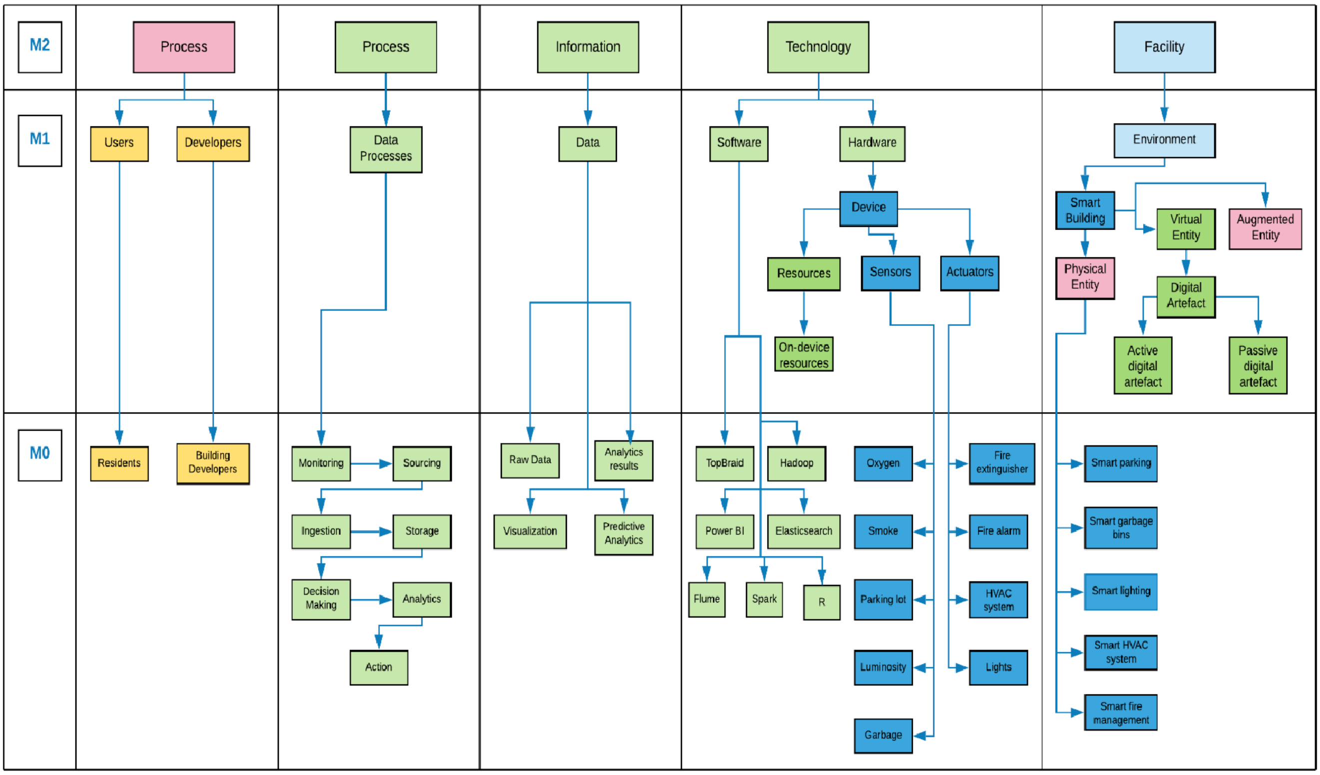

FIGURE 10. **Designation of IBDMA concepts into metamodel layers.**

management and analytics challenge, it is important to clearly understand the target facility to develop optimized big data management and analytics solution.

The "information" generated from the intersection of "people", "process" and "technology" helps to autonomously control the "facility" as shown in Figure 2. For this research, we consider the autonomous control of five smart facilities which include HVAC system, smart lighting, fire detection, garbage management and parking management.

Since the "process" element joins and overlaps all the elements of the IBDMA framework, the "process" that overlaps the "facility" element of the IBDMA framework is "action" as shown in Figure 8.

TABLE 2. **IBDMA metamodel relationships.**

IBDMA relationships
Originates
Processes
Generates
Interfaces with
Installs
Helps maintain
Is composed of
Is monitored by
Has information on/acts on
Is associated with
Interacts with
Contains
Represents
Hosts

IV. IBDMA FRAMEWORK REFERENCE ARCHITECTURE

This section presents details about the IBDMA Framework reference architecture for a smart building use case. The smart building has 1000 sensors of five types: Oxygen, Temperature, Parking, Luminosity and Garbage monitoring sensors. There are 200 sensors of each type. The sensors generate data in real-time which gets sent to data sinks: 1) HDFS and 2) Elasticsearch. This data is ingested into Hadoop using Apache Flume. Once the data in Elasticsearch is indexed, it can be visualized in near real-time in Kibana. Flume agents push the data to HDFS where it is made available for Power BI for batched data visualization and predictive analytics by integrating R scripts within Power BI. The data in HDFS is analyzed in near using Apache Spark. Based on the data generated by the virtual sensors, the PySpark algorithm outputs messages on the terminal simulating how certain facilities (HVAC system, fire alarms, lights, parking spaces and garbage bins) in the smart building are being monitored and controlled. The complete reference architecture is presented in Figure 8.

The results obtained from the 'Data Analysis', 'Data Visualization' and 'Decision-Making' steps as shown

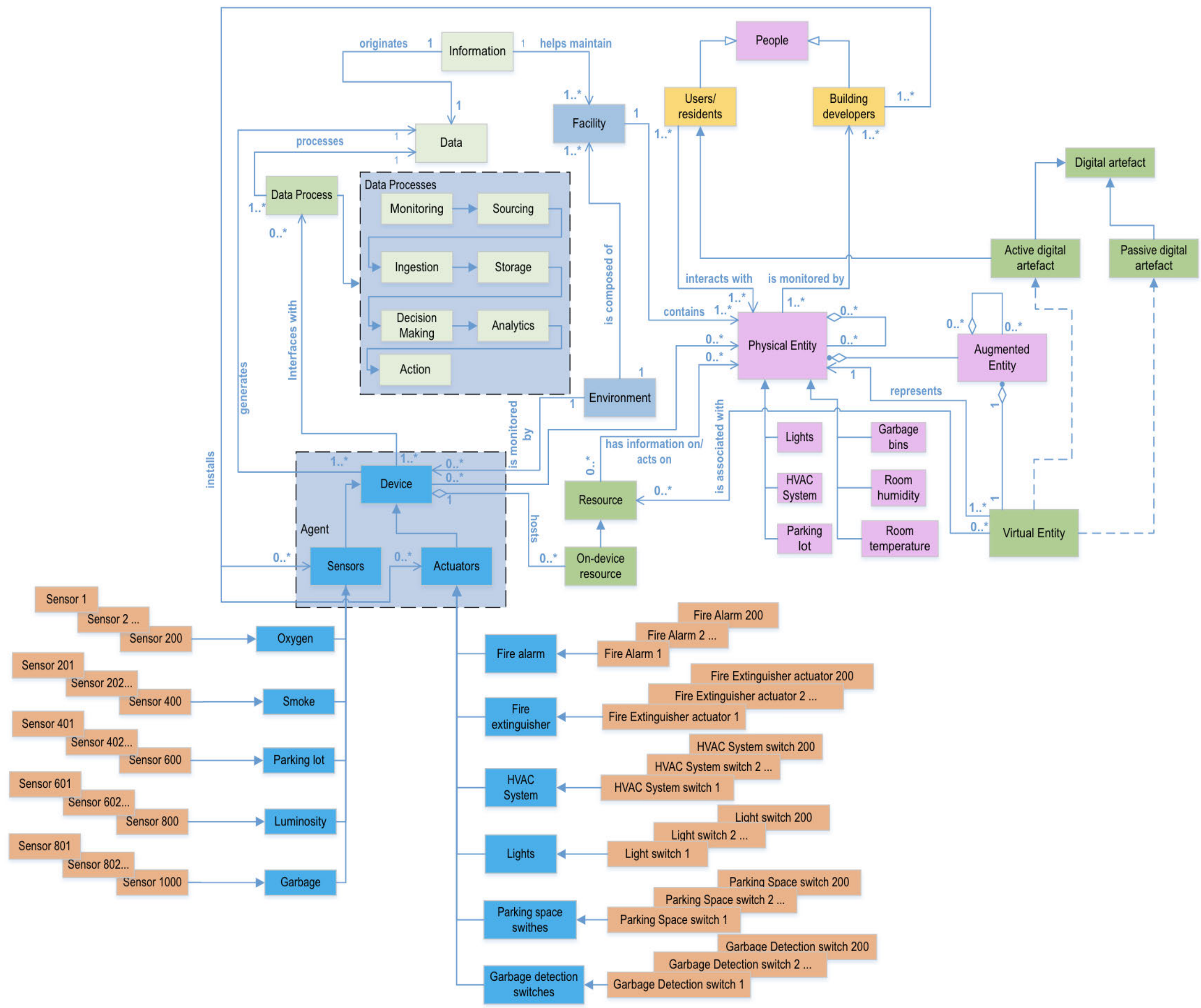

FIGURE 11. **IBDMA metamodel.**

in Figure 8 enable us to control the virtual smart building application scenario by simulating and activating various facilities in the smart building. This is done by triggering various actions based on the values generated from the virtual IoT sensors. We simulate the triggering actions of various control in a virtualized smart building environment by printing out text messages on the terminal.

To control and maintain the oxygen concentration in the smart building, the PySpark algorithm constantly monitors and analyzes incoming oxygen data in near real-time. If the value detected is below the minimum threshold level of oxygen concentration, the associated HVAC system is turned ON. This is denoted by printing out *"HVAC system X turned ON"* where X represents a particular location in the smart building. When the oxygen concentration returns into the acceptable range, the HVAC system is turned OFF. We denote this by printing *"HVAC system X turned OFF"*. If, however, oxygen concentration is within the acceptable range, the HVAC system remains idle and the PySpark algorithm outputs *"Oxygen level at X OKAY"*.

For smoke detectors, if the value detected by any smoke detection sensor is above a certain threshold, indicating that there is a fire scenario and in that scenario the fire alarm associated with that smoke detector is turned ON. We illustrate this by printing "Fire alarm X turned ON" on the terminal window where X represents the location in the smart building where smoke is detected. If there is no fire or hazardous gases, our system will print a message saying, "No fire at X".

For parking spaces sensors, when a parking space is filled, the framework issues a message "Parking X is occupied". When the parking space is empty, the message shown to the residents is "Parking X is empty" where X represents the location of the parking space. The residents can then park their car to the empty parking spaces. The building admin can take useful decisions by analyzing the data from the parking

TABLE 3. IBDMA metamodel concepts mapped with model relationships found in literature.

IBDMA Concepts	ArchiMate Concepts	FAML Concept	Adaptive Architecture Meta Model Concept	TOGAF Concept	ISO/IEC/IEE 42010 Concept	IoT reference model (Book: Enabling Things to Talk. Chapter 7 IoT reference model) Springer Open	BIM (Springer book)
People			Human		Stakeholders	user	People
Users/residents	Business role	role		Actor/role		human user	Users
Building developers							Developer
Physical Entity						Physical Entity	Entity
Augmented Entity						Augmented Entity	
Digital Artefact	Artifact	Facet				Digital Artefact	Artifacts
Active Digital Artefact						Active Digital Artefact	
Passive Digital Artefact						Passive Digital Artefact	
Virtual Entity						Virtual Entity	
Facility	Facility		Facility				Facility
Environment		Environment	Environment		Environment		Environment
Resource		Resource				Resource	Resource
On-device resource						On-device resource	
Information			Information				Information
Data				Data Entity			Data
Data Process	Technology Process		Technology				Process
Device	Device					Device	Devices
Actuators						Actuators	Actuators
Sensors						Sensors	Sensors
Agent		Agent					

TABLE 4. IBDMA metamodel relationships mapped with model relationships found in literature.

IBDMA relationships	ArchiMate	FAML	Adaptive Architecture metamodel	TOGAF	ISO/IEC 42010 relationships	IoT reference model (Book: Enabling Things to Talk. Chapter 7 IoT reference model) Springer Open	BIM (Springer book)
originates		generated from					
processes							
generates							generates
interfaces with							
connects to							
installs							
helps maintain							
is composed of	composition	composition	composed of			composition	composition
is monitored by					has	monitors	
has information on/acts on						has information on/acts on	
is associated with	association	association	associated with		association	is associated with	association
interacts with						interacts with	
contains						contains	
represents						represents	
hosts						hosts	

spaces sensors. For example, they can see if there is a need to build another parking area to improve the comfort of the residents.

To control and maintain good luminosity levels in the smart building, the PySpark algorithm keeps on monitoring the incoming luminosity sensor data in near real-time. If the value detected by the algorithm is below the minimum threshold level of luminosity, the associated lights will be turned ON. This, in our research is illustrated by printing out "Lights at X turned ON" where X represents the sensor id or location. If, however, the luminosity level detected by the sensors is within the acceptable range, the proposed framework does

TABLE 5. IBDMA metamodel relationship between concepts.

Concept 1	Relationship	Concept 2
Building Developers	Association - *'Installs'*	Sensors
Building Developers	Association - *'Installs'*	Actuators
Device	Association - *'Generates'*	Data
Device	Association - *'Interfaces with'*	Data Process
Data Process	Association - *'Processes'*	Data
Information	Association - *'Originates'*	Data
Information	Association - *'Helps Maintain'*	Facility
Environment	Association - *'Is Composed Of'*	Facility
Environment	Association - *'Is Monitored By'*	Device
Device	Association - *'Hosts'*	On-Device Resources
Resource	Association - *'Has Information On/Acts On'*	Physical Entity
Facility	Association - *'Contains'*	Physical Entity
Virtual Entity	Association - *'Is Associated With'*	Resource
User/residents	Association - *'Interacts With'*	Physical Entity
Virtual Entity	Association - *'Represents'*	Physical Entity
Physical Entity	Association - *'Is Monitored By'*	Building Developers

not turn ON or OFF the lights. In this case, the PySpark algorithm prints "Luminosity level at X OKAY" illustrating that luminosity levels are okay at that location. When the system analyzes that the lights at a particular location needs to be turned off, the system displays a message stating, "Lights at X turned OFF".

For garbage bins sensors, if a particular garbage detection sensor detects that the garbage at a particular location is full, the system issues a message saying, "Garbage at X is Full". If, however, the garbage at a particular location has more space for garbage, the system displays a message saying, "Garbage at X has space". Using this data, the smart building admin can effectively manage the garbage of the building. The admin can also check at which times and days of the week the garbage is more and at which locations it is more as compared to other garbage locations. Similarly, the admin can analyze the data to see if they need to develop new garbage locations or provide more garbage collectors to a particular location.

The results obtained on the Cloudera terminal screen while performing the data analysis are presented in Figure 9.

V. METAMODEL FOR BIG DATA MANAGEMENT AND ANALYTICS FOR IoT ENABLED SMART BUILDINGS

The IBDMA metamodel is the second component of IBDMA Framework as shown in Figure 1. To construct IBDMA metamodel a set of relevant metamodels and architectures were first selected. IBDMA concepts and relationships between the concepts are rooted in the existing literature. To develop the IBDMA metamodel, we followed a six step Metamodeling Creation Process adapted from [35] and [36]. These six steps include:

- Step 1: Define IBDMA concepts and relationships from IBDMA framework reference architecture and the use case
- Step 2: Mapping similar concepts and relationships onto relevant domain metamodels and architectures
- Step 3: Reconciliation of definitions
- Step 4: Designation of concepts

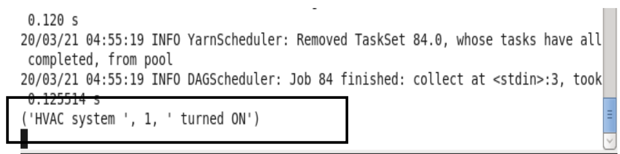

FIGURE 12. Output of validation scenario 1.

- Step 5: Identification of relationships and resultant IBDMA Metamodel
- Step 6: Validation of metamodel

A. DEFINE METAMODEL CONCEPTS AND RELATIONSHIPS FROM IBDMA FRAMEWORK REFERENCE ARCHITECTURE AND THE USE CASE

This step includes reviewing the IBDMA framework reference architecture and the use case as earlier discussed in Sections II and III. Following the review of both the IBDMA framework contextual elements and its reference architecture (Figure 2 and Figure 8 respectively), we gathered concepts and relationships that are used in the smart environments in general and IoT-enabled smart buildings in particular. We also gathered the instances of the concepts from the smart building use case discussed in Section III. Based on our analysis we summarized the following concepts as part of the IBDMA framework metamodeling process as shown in TABLE 1.

Similarly, on reviewing the IBDMA framework contextual elements and its reference architecture as presented in section II and II respectively, we identified the Relationships in the Metamodel as shown in TABLE 2.

The next step is to ensure that the concepts and their relationships can generate concepts and relationships of other relevant metamodels.

B. STEP 2: MAPPING SIMILAR CONCEPTS AND RELATIONSHIPS ONTO RELEVANT DOMAIN METAMODELS AND ARCHITECTURES

This step provides a preliminary validation to ensure that IBDMA metamodel is semantically adequate and can generate other concepts and relationships in relevant metamodels

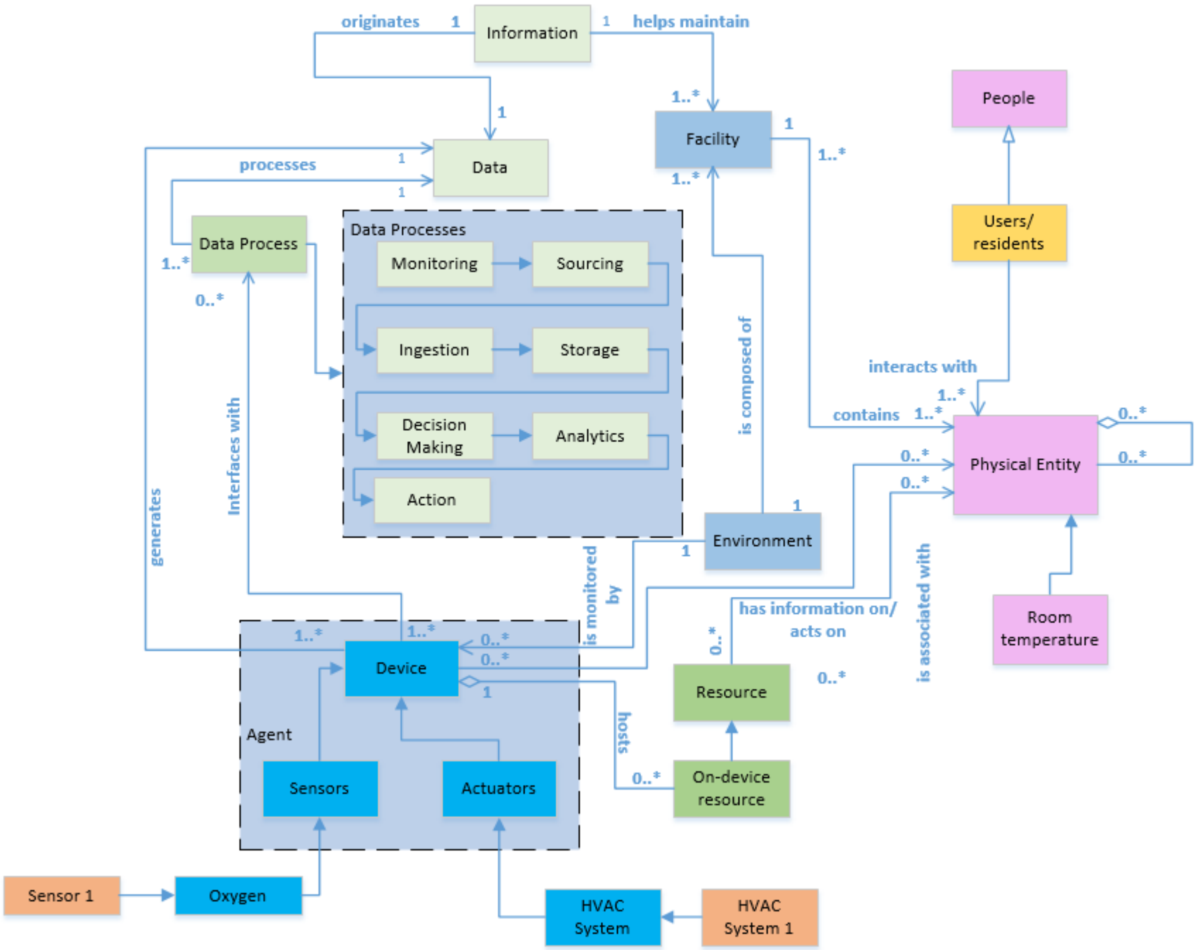

FIGURE 13. **Resultant IBDMA metamodel for scenario 1.**

and architectures. For this purpose, we considered and short-listed the following seven most relevant metamodels and architectures from literature.

1- ArchiMate
2- FAML
3- Adaptive Architecture metamodel
4- TOGAF
5- ISO/IEC/IEE 42010
6- IoT reference model [37]
7- BIM (Building Information Model) [38]

On review and analysis of the above in detail, we mapped the IBDMA metamodel concepts defined in the previous step to similar concepts founds in those architectures and metamodels. TABLE 3 lists the mapping of the IBDMA metamodel concepts onto the concepts found in the above mentioned seven metamodels and architectures. It can be seen from TABLE 3 that most of the IBDMA metamodel concepts are found in the relevant metamodels and architectures we chose for our analysis.

Similarly, we chose the same seven metamodels and architectures for our analysis of mapping the IBDMA metamodel relationships onto the relationships found in these metamodels and architectures. The resulting relationship mapping table can be found in TABLE 4 below.

It can be seen from TABLE 3 and TABLE 4 that although most of the concepts and relationships we defined for IBDMA metamodel can be mapped to the concepts and relationships of only a few metamodels found in literature. However, none of the available metamodels in the literature captures the concepts and relationships we defined in a single metamodel. Moreover, there are few relationships that cannot be found in any other metamodel as can be seen from TABLE 4. This strengthens the claims that IBDMA metamodel captures all the concepts and their relationships comprehensively to address big data management and analytics challenges for the smart building environments.

C. STEP 3: RECONCILIATION OF DEFINITIONS

In this step, we reconcile the differences between definitions of the concepts. The definitions of concepts chosen in the previous section are considered in choosing or synthesizing the common concept definition to be used. Since the

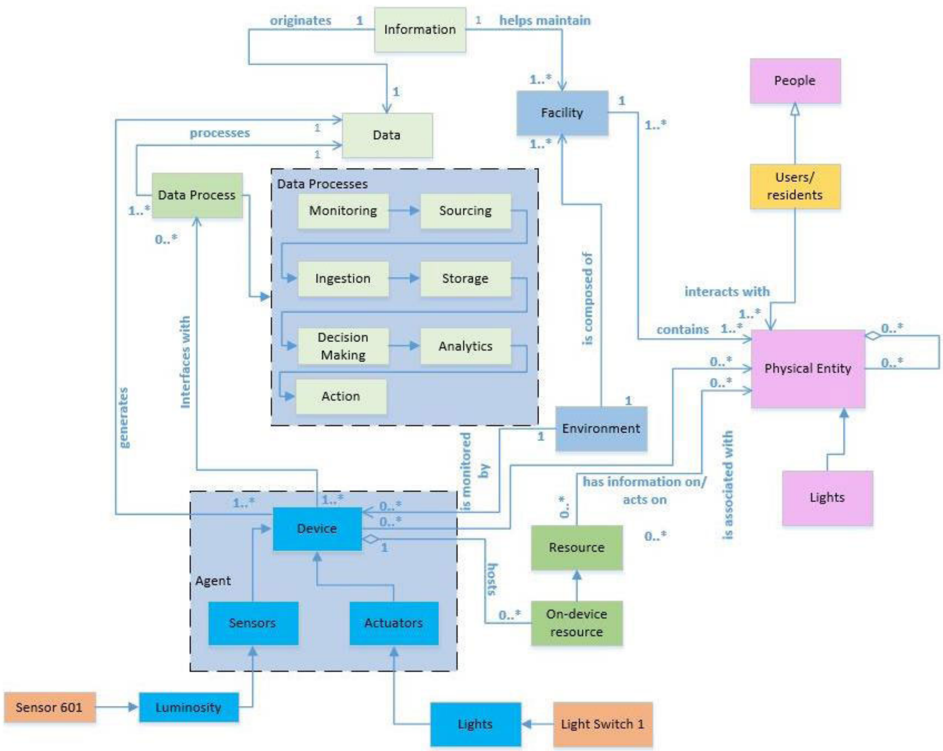

FIGURE 14. **Resultant metamodel for scenario 2.**

definitions of concepts come from various architectures and models, they were developed by people with varying perspectives and backgrounds. If there is a contradictory use of concept definition between two or more sources, then a we need to have a process to harmonize and fit the definition in the metamodel. Some architectures and models ignore explicit definitions of some of their concepts. In such cases, the reconciliation process is missing in those models. As an example, the concept of 'People' maybe defined differently in the seven chosen metamodels and architectures discussed in Step 2 as compared to the concept of People defined in IBDMA metamodel as presented in TABLE 1. However, we found that most of the concepts were defined in the same way as we defined them for IBDMA metamodel as shown in TABLE 1.

D. STEP 4: DESIGNATION OF CONCEPTS

Once the concepts have been finalized and reconciled, they are designated and arranged into metamodeling layers (M2, M1 and M0) [39]. The concepts in M2 layer are generic across all smart environments e.g. smart cities, smart buildings, smart homes, smart farms etc. The concepts in M1 are specific to smart buildings, while the concepts in M0 are actually the instances of the concepts present in M1.

The five concepts in M2 denote the five major elements of the IBDMA framework i.e. People, Process, Technology, Information and Facility. These concepts are common across various smart environments i.e. smart homes, smart offices, smart cities etc. These five concepts from M2 layer are broken down into their instances in M1 layer. The M1 layer specifically denotes concepts related to the smart buildings. In M0 layer, instances of the concepts of M1 layer are presented. Figure 10 represents the designation and arrangement of IBDMA Metamodel concepts into the metamodeling layers.

E. STEP 5: IDENTIFICATION OF RELATIONSHIPS AND RESULTANT IBDMA METAMODEL

In this step, we determine the relationships between the IBDMA metamodel concepts that are arranged into various metamodel layers. As shown in Figure 11, we use the (→), (➞) and (−◇) symbols to denote Association, Generalization and Aggregation relationships respectively. As an association example, 'Helps Maintain'' between *Information* and *Facility* concepts indicate that information helps maintain all the elements of the Facility. As a Generalization example, *Fire Alarm*, *Fire extinguisher*, *HVAC System*, *Lights*, *Parking space switches* and *Garbage Detection Switches* generalize the *Actuators* concept. As an aggregation example, *On-Device Resource* and *Device* are related by the relation 'hosts'. More examples of binary relationships are shown in Table 5. Concepts depicting hardware are shown in blue, software in green, animate (humans/animals) in yellow and

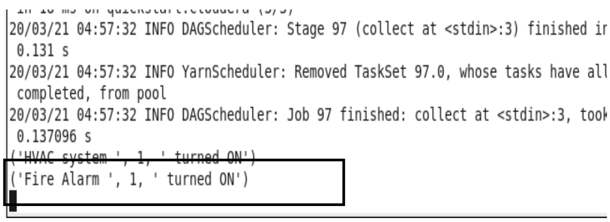

FIGURE 15. Output of validation scenario 3.

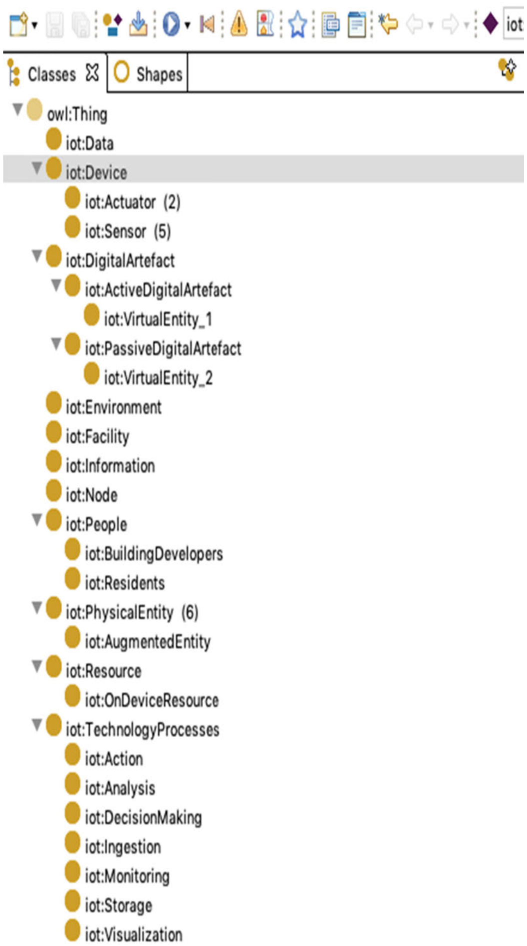

FIGURE 16. Classes in TopBraid.

concepts that fit into either multiple or no categories in pink. The relationship between the concepts are determined using the IBDMA reference architecture for the IoT-enabled smart building as presented in the previous section. The relationships between the concepts defined in Figure 11 are taken from TABLE 4. Table 5 outlines how the metamodel concepts are related to each other. By taking the IBDMA framework and its reference architecture as presented in Sections II and III, and on combining the metamodel concepts and the relationship between these concepts the resultant metamodel is developed which is presented in Figure 11.

TABLE 6. Contributions of the IBDMA Metamodel.

Sr. No	Contributions
1	The IBDMA framework provides a coherent Big Data Management and Analytics Framework by providing a reference architecture and the metamodel. Such a comprehensive framework with both a reference architecture and a metamodel has been missing in the literature.
2	IBDMA metamodel presents a comprehensive catalogue of concepts, relationships and the relationship between those concepts to address big data management and analytics challenges in Smart Buildings. Although there are generic metamodels that can be expanded to address these challenges however, these have not been tested and evaluated against the Smart Building scenarios.
3	The metamodels presented in the literature either focus on the big data analytics or on the smart environments. However, a framework or metamodel to address both these challenges simultaneously is missing. The IBDMA framework bridges this gap in the literature.
4	The IBDMA metamodel has been appropriately tested for Smart Building context and can be easily scaled for other smart contexts.
5	The IBDMA Framework will assist both researchers and practitioners to address the big data management and analytics challenges for the Smart Buildings.

TABLE 7. Limitations of the IBDMA Metamodel.

Sr. No	Limitations
1	In developing our metamodel, we considered only the models presented in English which could lead to a cultural bias. If models from other geographic regions were also considered, it could improve the completeness and applicability of the model.
2	While the IBDMA metamodel has been validated against practical use cases, however, in future research, a review of the metamodel by experts in Big Data or Smart Buildings domains could also add further credence to the metamodel.
3	IBDMA metamodel could evolve over time as other concepts and relationships are introduced in future. At this stage, there is no feedback mechanism to incorporate the expected evolution in the IBDMA metamodel. While some human intervention would be necessary, automating or partially automating this process would be beneficial. This would make the IBDMA metamodel stay current and be useful in managing Smart Building activities. Our future work will include developing automated feedback mechanisms and analytics that will facilitate the evolution of IBDMA metamodel.

F. STEP 6: VALIDATION OF METAMODEL

In this section, we use IBDMA metamodel to instantiate three practical use-cases for smart buildings. This will prove the effectiveness, completeness and comprehensiveness of the metamodel for smart building applications. We also import the developed metamodel into a knowledge graph management tool and prove the validity of the IBDMA metamodel using this tool as well.

1) METAMODEL EVALUATION AND VALIDATION SCENARIO 1

To evaluate IBDMA metamodel, we create an instance of metamodel for a specific use case. The smart building that we choose has a variety of different types of IoT sensors installed

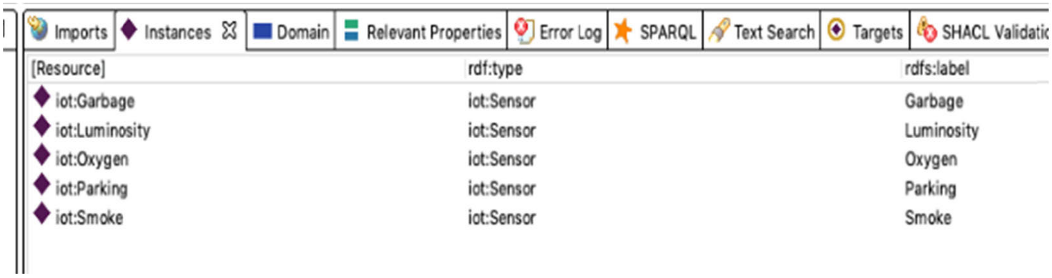

FIGURE 17. Instances of concepts in TopBraid.

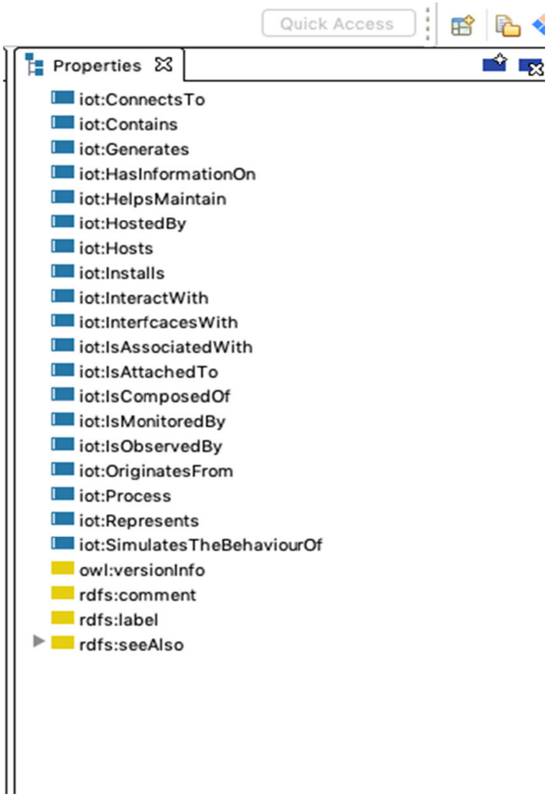

FIGURE 18. Metamodel relationships imported in TopBraid.

within the building. However, for scenario 1, we choose one oxygen sensor installed in the smart building which monitors the oxygen levels in one particular room of the smart building. For simplicity, we refer to this as 'Sensor 1' installed in room number 1 of the smart building we choose. We then implement the big data management and analytics architecture using Cloudera VM and create an end-to-end pipeline as depicted in Figure 8. This pipeline ingests the data generated by the IoT Oxygen sensor into HDFS, from where the value generated by the sensor is analyzed using Spark code and based on the value of the sensor, the smart building HVAC system is controlled. When sensor 1 generates a value, which is below the comfortable threshold level for humans, the Spark code produces an output showing, "HVAC System 1 turned ON", indicating that the HVAC system which serves room 1 where Sensor 1 is connected 1 is turned ON. This is presented in Figure 12.

Now we validate our metamodel using this scenario. As mentioned earlier, Sensor 1 generates the data about oxygen levels, this sensor being a "*Device*", 'generates' "*data*" and 'interfaces with' the "*data process*". The "*data*" generated by the sensor gets analyzed and produces useful "*information*" which 'originates' from the "*data*". This useful "*information*" 'helps maintain' the "*facility*" which 'contains' "*physical entities*". In this particular case, if the "*data process*" detects that the value generated by the oxygen sensor is too low, it triggers the "*HVAC System*" to turn ON and to make sure that the level of oxygen in room 1 remains within the acceptable range for the "*residents*" of the smart building "facility". The resultant metamodel for this particular example is presented in Figure 13. It can be seen clearly from this example scenario that the IBDMA metamodel encompasses and captures all the concepts required for validating this example use case.

2) METAMODEL EVALUATION AND VALIDATION SCENARIO 2

To evaluate the IBDMA metamodel, we consider our second use case in this section. We choose a University's smart building based in Australia as our smart building instance for this particular scenario. This building has 12 floors with various types of sensors installed in the building. One of the sensor types installed in the building is Waspmote types of sensors. Waspmote sensors installed in the building included Oxygen sensors, Carbon Dioxide sensors, Luminosity sensors, Temperature sensors, Humidity sensors and other sensors. For this scenario, we choose a Luminosity sensor installed at level 6 of the University's smart building. This sensor produces a binary value output with 1 representing good luminosity levels in the smart building room and 0 representing low luminosity levels in the room where the sensor is installed. For simplicity, we designate this sensor as sensor 601. When sensor 601 outputs a '1' value, the lights in that particular room of the smart building turns OFF. On the other hand, when sensor generates a '0' value, the light turns ON in the room.

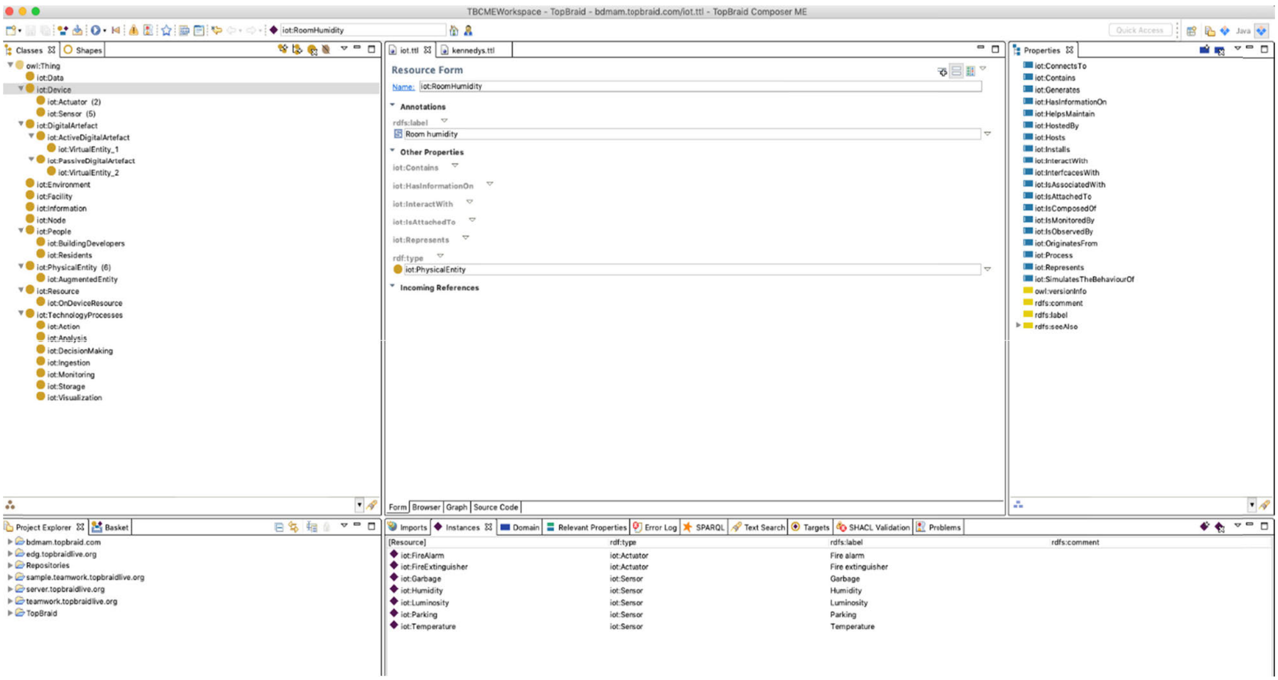

FIGURE 19. **TopBraid window with the imported Metamodel.**

Now we instantiate the metamodel for this use case. As mentioned earlier, Sensor 601 generates the data about luminosity levels, this sensor being a "Device", 'generates' "data" and 'interfaces with' the "data process". The "data" generated by the sensor gets analyzed and produces useful "information" which 'originates' from the "data". This useful "information" 'helps maintain' the "facility" which 'contains' "physical entities". In this particular case, if the "data process" detects that the value generated by the luminosity sensor is too low, it triggers the "Light" to turn ON and to make sure that the luminosity level in a particular room 1 remains within the acceptable range for the "residents" of the smart building "facility". The resultant metamodel for this particular example is presented in Figure 14. Hence, it can be seen clearly from this example scenario that IBDMA metamodel encompasses and captures all the concepts required for validating this example use case.

3) METAMODEL EVALUATION AND VALIDATION SCENARIO 3

To evaluate the metamodel, we consider our final smart building example scenario in this section by creating an instance of the IBDMA metamodel for a specific use case. The smart building, we choose has a variety of different types of IoT sensors installed within the building. However, for scenario 3, we choose one smoke detection sensor installed in the smart building which monitors the smoke levels in one particular room of the smart building. For simplicity, we refer to this as 'Sensor 201' installed in room number 1 of the smart building we choose. We then implement the big data management and analytics architecture using Cloudera VM and create an end-to-end pipeline as depicted in Figure 8. This pipeline ingests the data generated by the IoT Smoke Detection sensor into HDFS, from where the value generated by the sensor is analyzed using Spark code and based on the value of the sensor, the smart building Fire Alarm is controlled. When sensor 201 generates a value, which is above the comfortable threshold level for humans, the Spark code produces an output saying, "Fire Alarm 1 turned ON", indicating that the Fire Alarm which serves room 201 where Sensor 201 is connected is turned ON. This is presented in Figure 15.

We now validate our metamodel using this scenario, by using TopBraid as the metamodel management tool. Sensor 201 generates the data about smoke levels, this sensor being a "Device" interfaces with the "data process".

Hence the "data" generated by the sensor gets analyzed and produces useful "information". This useful information helps maintain the "facility" which contains "physical entities". In this particular case, if the "data process" detects that the value generated by the smoke detection sensor is too high, it triggers the "Fire Alarm" to turn ON and makes sure that the "Fire Alarm" at the location where the smoke was detected turns ON. This alerts the "Users" or "residents" in the building so that they can stay safe by evacuating the building. We validate the metamodel in TopBraid in the section below for this particular example scenario.

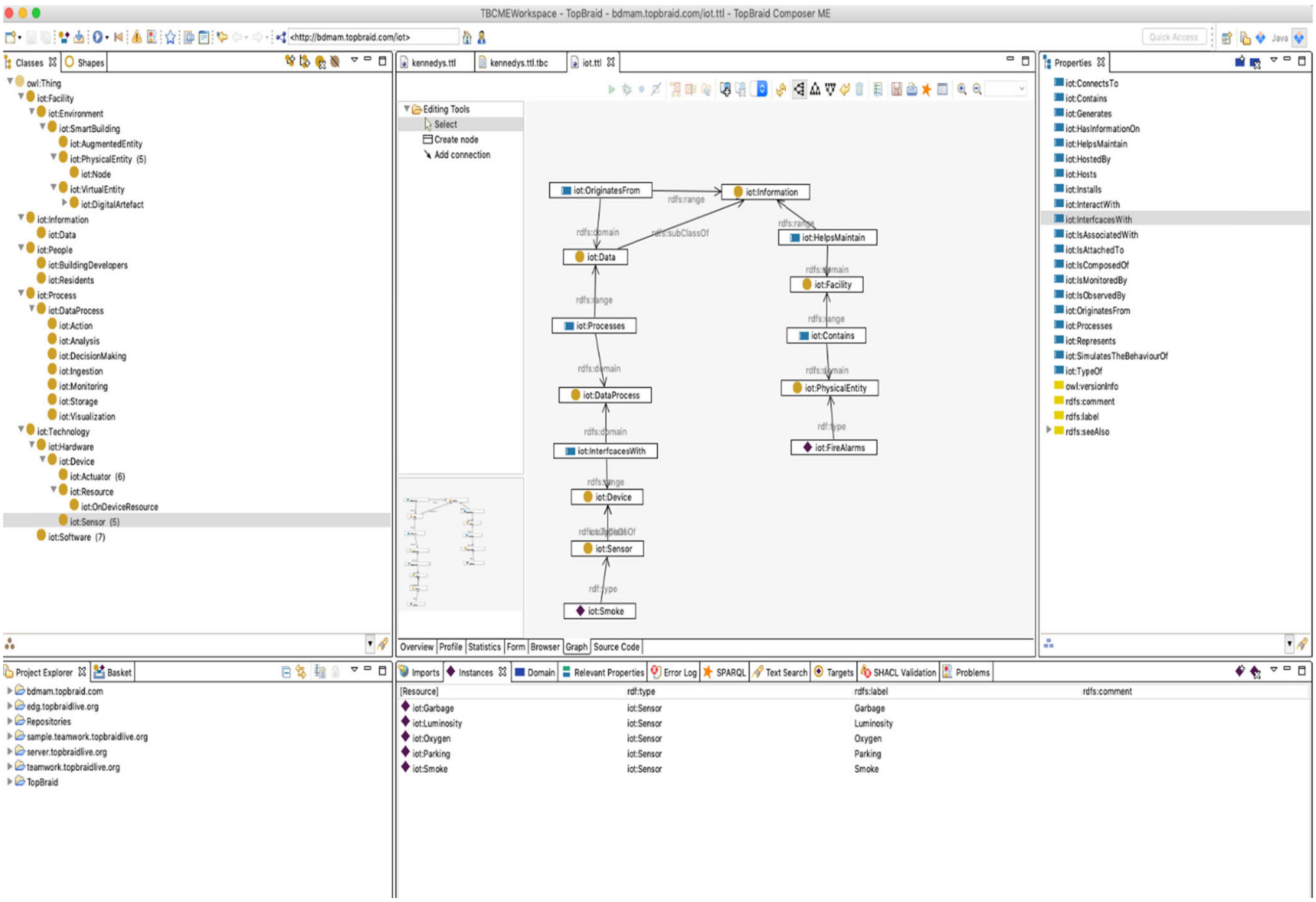

FIGURE 20. **Metamodel validation in TopBraid – evaluation and validation scenario 3.**

4) TopBraid METAMODEL IMPORT AND VALIDATION SCENARIO 3

In order to operationalize the metamodel, we import the metamodel into TopBraid EDG (Enterprise Data Governance) [40]. Topbraid is a modular set of different types of graphs, expressing knowledge about the things needed for managing and governing data. It enables the rapid assembly of dynamic ontology-driven applications by providing full support for the entire Semantic Application Lifecycle from development to deployment. We validate the metamodel using TopBraid by considering scenario 3 with a smoke detection sensor installed in the smart building.

Importing the metamodel in TopBraid involves creating classes for the concepts in the metamodel and then defining instances of those classes. Figure 16 shows the metamodel concepts of IBDMA metamodel imported into TopBraid as classes.

Next, we define relationships of the IBDMA metamodel in TopBraid. The pane on the right side of the TopBraid window shows the properties (relationships) between the classes (concepts). The pane on the bottom of TopBraid window lists the instances of a particular class (concept) that is selected in the classes pane in TopBraid as shown in Figure 18.

The pane on the bottom of TopBraid window lists the instances of a particular class (concept) that is selected in the classes pane in TopBraid as shown in Figure 18.

The metamodel concepts and relationships between the concepts on importing into TopBraid are shown in Figure 19.

Now we instantiate the metamodel for the Smoke Detection sensor 201 scenario 3 as presented in previous section. The resultant instantiation of the IBDMA metamodel for the Metamodel Validation Scenario is presented in Figure 20. The zoomed-in version of the IBDMA metamodel for this instance is presented in Figure 21. It can be seen clearly that the metamodel imported into TopBraid encompasses all the concepts and the relationship between these concepts. This consistent operationalization of IBDMA metamodel enabled much easier use of the metamodel, proving it to be valid for this third and final example use case.

VI. CONTRIBUTION AND LIMITATION OF THE IBDMA METAMODEL

This section lists the major contributions and the limitations of the IBDMA Metamodel presented in the paper.

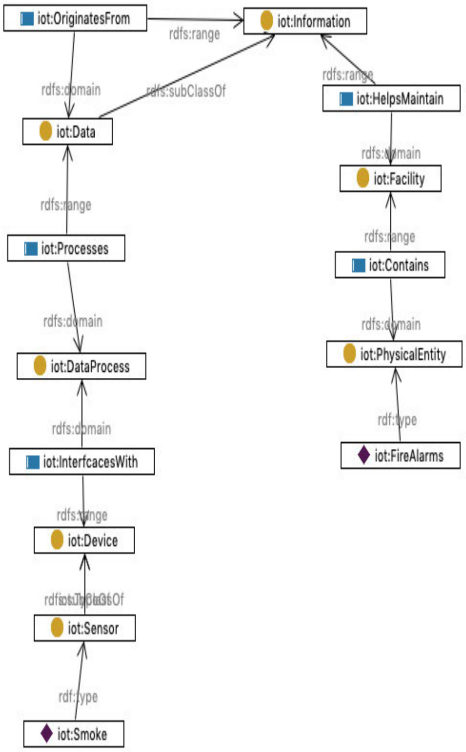

FIGURE 21. Metamodel validation scenario 2 (zoomed in) version.

VII. CONCLUSION AND FUTURE WORK

In this article, we presented the Integrated Big Data Management and Analytics metamodel in a familiar format, UML, to increase its ease of use and broaden its appeal. The aim of the metamodel to address the big data management and analytics challenges faced by researchers and practitioners working in the Smart Buildings domain. We used the IBDMA framework and its reference architecture as a basis for our metamodel. In this work, we extracted concepts and relationship between the concepts from the IBDMA framework. We validated their semantics against several other relevant metamodels and architectures. The finalized concepts and relationships were arranged into Metamodel layers (M2 – M0). The resultant metamodel, was then validated using three practical use cases within smart buildings environments. And finally, to operationalize IBDMA metamodel, it was imported into TopBraid and further validated within TopBraid for a third use case to illustrate its effectiveness.

IBDMA metamodel is the core contribution of this article. It is intended to become an effective platform for sharing and integrating the big data management and analytics knowledge for IoT enabled smart buildings from various sources. Existing models for big data management and analytics for smart buildings are not based on metamodels but rather are based on the frameworks and architectural aspects. Their interoperability thus far remains an issue that IBDMA metamodel targets at. Existing literature provides generic metamodels for the smart environments which have not been validated thoroughly for smart building applications. This is the first work that develops an integrated metamodel for big data management and analytics for IoT enabled smart buildings and has been tested thoroughly to prove its effectiveness and completeness. The work will help researchers and practitioners in understanding the big data management and analytics challenges and how to address them in IoT enabled smart buildings. The metamodel will also be used as a tool to determine the completeness of any big data solution implementation in smart buildings. Our future work will aim to extend this metamodel for other smart environments (not just buildings) and consider a more detailed and comprehensive use case for validation.

REFERENCES

[1] M. R. Bashir and A. Q. Gill, "Towards an IoT big data analytics framework: Smart buildings systems," in *Proc. IEEE 18th Int. Conf. High Perform. Comput. Commun., IEEE 14th Int. Conf. Smart City, IEEE 2nd Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Dec. 2016, pp. 1325–1332.
[2] F. Truyen, "The fast guide to model driven architecture the basics of model driven architecture," Cephas Consulting Corp., Tustin, CA, USA, White Paper, Jan. 2006. Accessed: Sep. 16, 2020. [Online]. Available: http://www.smallake.kr/wp-content/uploads/2016/04/Cephas_MDA_Fast_Guide.pdf
[3] R. Bashir. *Github*. Accessed: Jun. 11, 2020. [Online]. Available: https://github.com/c3212218/BDMAM/tree/master/figures
[4] M. Völter, T. Stahl, J. Bettin, A. Haase, and S. Helsen, *Model-Driven Software Development: Technology, Engineering, Management*. Hoboken, NJ, USA: Wiley, 2013.
[5] F. Cicirelli, G. Fortino, A. Guerrieri, G. Spezzano, and A. Vinci, "Metamodeling of smart environments: From design to implementation," *Adv. Eng. Informat.*, vol. 33, pp. 274–284, Aug. 2017.
[6] A. Erraissi, B. Mouad, and A. Belangour, "A big data visualization layer meta-model proposition," in *Proc. 8th Int. Conf. Model. Simul. Appl. Optim. (ICMSAO)*, Apr. 2019, pp. 1–5.
[7] G. Fortino, A. Guerrieri, W. Russo, and C. Savaglio, "Towards a development methodology for smart object-oriented IoT systems: A metamodel approach," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2015, pp. 1297–1302.
[8] W. Boumans, S. Hoppenbrouwers, and I. E. Herder, "A comprehensive meta model for the current data landscape," Radboud Univ., Nijmegen, The Netherlands, Tech. Rep., 2019. Accessed: Sep. 16, 2020. [Online]. Available: https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=A+comprehensive+meta+model+for+the+current+data+landscape&btnG=
[9] M. Fatehah and V. Mezhuyev, "Design and process metamodels for modelling and verification of safety-related software applications in smart building systems," presented at the 6th Int. Conf. Inf. Technol., IoT and Smart City, Hong Kong, 2018.
[10] D. Martinez-Mosquera, S. Luján-Mora, and H. Recalde, "Conceptual modeling of big data extract processes with UML," in *Proc. Int. Conf. Inf. Syst. Comput. Sci. (INCISCOS)*, Nov. 2017, pp. 207–211.
[11] J. I. R. Molano, J. M. C. Lovelle, C. E. Montenegro, J. J. R. Granados, and R. G. Crespo, "Metamodel for integration of Internet of Things, social networks, the cloud and industry 4.0," *J. Ambient Intell. Humanized Comput.*, vol. 9, no. 3, pp. 709–723, Jun. 2018.
[12] F. Cicirelli, G. Fortino, A. Guerrieri, A. Mercuri, G. Spezzano, and A. Vinci, "A metamodel framework for edge-based smart environments," in *Proc. IEEE Int. Conf. Cloud Eng. (IC E)*, Apr. 2018, pp. 286–291.
[13] E. Damiani, B. Oliboni, E. Quintarelli, and L. Tanca, "A graph-based meta-model for heterogeneous data management," *Knowl. Inf. Syst.*, vol. 61, no. 1, pp. 107–136, Oct. 2019.
[14] A. Bali, M. Al-Osta, and G. Abdelouahed, *An Ontology-Based Approach for IoT Data Processing Using Semantic Rules*. Cham, Switzerland: Springer, 2017, pp. 61–79.

[15] R. Petrasch and R. Kongkachandra, "Towards enterprise data management: Data model mapping," in *Proc. IEEE Int. WIE Conf. Electr. Comput. Eng. (WIECON-ECE)*, Dec. 2018, pp. 72–74.
[16] A. P. Plageras, K. E. Psannis, C. Stergiou, H. Wang, and B. B. Gupta, "Efficient IoT-based sensor BIG data collection–processing and analysis in smart buildings," *Future Gener. Comput. Syst.*, vol. 82, pp. 349–357, May 2018.
[17] Y. Agarwal, B. Balaji, R. Gupta, J. Lyles, M. Wei, and T. Weng, "Occupancy-driven energy management for smart building automation," in *Proc. 2nd ACM Workshop Embedded Sens. Syst. Energy-Efficiency Building (BuildSys)*, 2010, pp. 1–6.
[18] G. A. Qumar, *Adaptive Cloud Enterprise Architecture*. Singapore: World Scientific, 2015.
[19] A. Q. Gill, "Agile enterprise architecture modelling: Evaluating the applicability and integration of six modelling standards," *Inf. Softw. Technol.*, vol. 67, pp. 196–206, Nov. 2015.
[20] D. S. Nunes, P. Zhang, and J. Sa Silva, "A survey on Human-in-the-Loop applications towards an Internet of all," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 2, pp. 944–965, 2nd Quart., 2015.
[21] *Apache Flume*. Accessed: Jun. 15, 2020. [Online]. Available: https://flume.apache.org/
[22] G. Liu, W. Zhu, C. Saunders, F. Gao, and Y. Yu, "Real-time complex event processing and analytics for smart grid," *Procedia Comput. Sci.*, vol. 61, pp. 113–119, Jan. 2015.
[23] *Apache Spark*. Accessed: Jun. 15, 2020. [Online]. Available: https://spark.apache.org/
[24] *Tableau*. Accessed: Jun. 15, 2020. [Online]. Available: https://www.tableau.com/
[25] *Power BI*. Accessed: Jun. 15, 2020. [Online]. Available: https://powerbi.microsoft.com/en-us/
[26] *PyCharm*. Accessed: Jun. 15, 2020. [Online]. Available: https://www.jetbrains.com/pycharm/
[27] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The Hadoop distributed file system," in *Proc. MSST*, vol. 10, May 2010, pp. 1–10.
[28] *Re1tReddy/Spark*. Accessed: Jun. 19, 2019. [Online]. Available: https://github.com/Re1tReddy/Spark
[29] *Elasticsearch*. Accessed: Jun. 15, 2020. [Online]. Available: https://www.elastic.co/products/elasticsearch
[30] *Kibana*. Accessed: Jun. 15, 2020. [Online]. Available: https://www.elastic.co/products/kibana
[31] *Apache Lucene*. Accessed: Jun. 15, 2020. [Online]. Available: http://lucene.apache.org/
[32] *Logstash*. Accessed: Jun. 15, 2020. [Online]. Available: https://www.elastic.co/products/logstash
[33] *30 Days of DevOps: Choosing the Right DB in AWS*. Accessed: Jun. 15, 2020. [Online]. Available: https://eoblog.net/
[34] *Kibana—Wikipedia*. Accessed: Jun. 15, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Kibana
[35] S. H. Othman and G. Beydoun, "Metamodelling approach to support disaster management knowledge sharing," Univ. Wollongong Australia, Wollongong, NSW, Australia, Tech. Rep., 2010. Accessed: Sep. 16, 2020. [Online]. Available: https://ro.uow.edu.au/infopapers/3453/
[36] G. Beydoun, G. Low, B. Henderson-Sellers, H. Mouratidis, J. J. Gomez-Sanz, J. Pavon, and C. Gonzalez-Perez, "FAML: A generic metamodel for MAS development," *IEEE Trans. Softw. Eng.*, vol. 35, no. 6, pp. 841–863, Nov. 2009.
[37] A. Bassi, M. Bauer, M. Fiedler, and R. V. Kranenburg, *Enabling Things to Talk*. Berlin, Germany: Springer, 2013.
[38] A. Borrmann, M. König, C. Koch, and J. Beetz, *Building Information Modeling: Technologische Grundlagen und industrielle Praxis*. Brunswick, Germany: Springer-Verlag, 2015.
[39] C. Atkinson and T. Kühne, "Model-driven development: A metamodeling foundation," *IEEE Softw.*, vol. 20, no. 5, pp. 36–41, Sep. 2003.
[40] *TopQuadrant*. Accessed: Jun. 15, 2020. [Online]. Available: https://www.topquadrant.com/products/topbraid-enterprise-data-governance/

MUHAMMAD RIZWAN BASHIR is currently pursuing the Ph.D. degree with the School of Software, University of Technology Sydney. His work focuses on big data management and analytics. His research interests include data science, machine learning, big data management, and the IoT. Besides this, he has been working in the industry as a Data Science, Big Data, and the IoT Professional for over ten years.

ASIF QUMER GILL received the Master of Business degree and the M.Sc. and Ph.D. degrees in computing science (software engineering). He is currently a result-oriented academic cum practitioner with extensive more than 20 years of experience in IT in various sectors, including banking, consulting, education, finance, government, nonprofit, software, and telecommunication. He is currently an Associate Professor and the Director of the DigiSAS Laboratory, School of Computer Science, University of Technology Sydney. He specializes in adaptive enterprise architecture and information-centric secure digital ecosystems.

GHASSAN BEYDOUN received the degree in computer science and the Ph.D. degree in knowledge systems from the University of New South Wales. He is currently a Professor of information systems with the School of Information, Systems, and Modelling, University of Technology Sydney. He has authored more than 150 papers in international journals and conferences. His research projects are sponsored by the Australian Research Council, the NSW State Government, and the private sector. He investigates the best uses of models in developing methodologies for distributed intelligent systems. His other research interests include cloud adoption, disaster management, decisions support systems, and their applications.

BRAD MCCUSKER has a background as an Enterprise Architect, an Accountant, and an ICT Consultant. Before co-founding SURROUND, he has worked as a Senior Enterprise Architect with Accenture, IBM Global Services, the IBM Business Consulting Group, IBM Japan, and CSC Australia. He has also worked in various senior architectural and consulting roles with ING Australia, the Australian and New Zealand Bank, the Department of Health, the Australian Treasury, the Department of Social Services, and the Department of Human Services. He is also a visiting Fellow at ANU.