Building Operating Systems: A Cloud-based Architecture for Enabling Knowledge Representation and Improving the Adaptability in Smart Buildings

Adrian Taboada-Orozco[a], Kokou Yetongnon[b] and Christophe Nicolle[c]
Laboratory CIAD, Univ. Bourgogne Franche-Comté, CIAD EA 7533, 21000 Dijon, France

Keywords: Internet of Things, Knowledge Representation, Smart Building, Building Automation Systems, Building Operating Systems, Smart Cities, Adaptability.

Abstract: Hardware and software's constant evolution opens new horizons for developing systems focused on buildings. Thanks to this evolution, the smartization of them is possible. The target of building automation systems is always to reduce human intervention, thus avoiding errors and maximizing the building resources. However, these systems still require an operator that knows the building's topology, the elements inside, the dwellers' preferences, and how to operate the automation system. The operator uses this knowledge to adapt the automation system by configuring parameters and thresholds to deal with the most common issues in buildings. This paper introduces a new approach named WITTYM-BOS (W-BOS) that consists of a unique Building Operating System (BOS) architecture, doted with a knowledge representation comparable with the operators' knowledge. In this way, we transfer the abilities of an operator to improve the adaptability of a building automation system like BOS. To construct the knowledge, we combine static and dynamic information such as Industrial Foundation Classes (IFC) and data coming from the Internet of Things (IoT). We construct a preliminary prototype to illustrate our concept and validate use cases. Our work opens new horizons to innovative applications that profit from the easy understandability of our approach.

1 INTRODUCTION

Automation systems have been evolving with the exponential improvement of hardware and software. Furthermore, automation has reached a unique inflection point towards the smartization of objects. Since the conception of the first Building Automation System (BAS) in 1850 (Martirano and Mitolo, 2020), they have been reducing human intervention progressively. This fact has been assumed as smartization of buildings (Buckman et al., 2014). However, this term is not precise and also evolves. A clear example is Domotic, early considered a smart building system (Buckman et al., 2014). Domotic displays data and allows local control. Although Domotic automates building elements like lights or doors, it is still dependent on an operator. In this case, the domotic itself does not make the building smart (Sleman and Moeller, 2011). Instead, the operator uses its knowledge of building topology and elements inside to solve dwellers' issues. Beyond the operator's inner knowledge, it possesses a capacity to adapt the system to new scenarios.

The rigid architecture of BAS has pointed out specific management of building tasks such as monitoring thermal comfort or energy consumption. It is this specificity that has impeded its adaptation to new scenarios. BAS systems like Heating Ventilation and Air Conditioner (HVAC) controllers, Programmable Logic Controllers (PLC), Supervisory Control and Data Acquisition (SCADA), and Building Management Systems (BMS) are examples of rigid building automation systems that require highly trained experts to manipulate the data and operate the system.

The main issue of BAS remains on how to break with their rigidity and how to provide adaptability to them. We believe that by transferring the operator's knowledge and simplifying its access and the control of the entire system (hardware and software), we make a building fully adaptable to new issues. Consequently, the smartization of the building is possible. While the term smart is still debatable, some

[a] https://orcid.org/0000-0002-2396-3286
[b] https://orcid.org/0000-0002-6949-1050
[c] https://orcid.org/0000-0002-8118-5005

Taboada-Orozco, A., Yetongnon, K. and Nicolle, C.
Building Operating Systems: A Cloud-based Architecture for Enabling Knowledge Representation and Improving the Adaptability in Smart Buildings.
DOI: 10.5220/0010650400003064
In *Proceedings of the 13th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2021) - Volume 2: KEOD*, pages 162-169
ISBN: 978-989-758-533-3; ISSN: 2184-3228

researchers support our belief in equating smartness, and adaptability (Beetz et al., 2009). The term adaptability has a crucial connotation in the automation field. This term is used to describe the ergonomic of the system towards end-users (ISO9241-210, 2019) and it is the final barrier of BAS systems.

Nowadays, the term Building Operating System (BOS) emerges to define a system able to solve and simplify the complex underlying hardware and software of BAS. BOS is defined as an intermediate system between field equipment and services (Sleman and Moeller, 2011). This definition has not yet been clarified. BOS is cutting-edge technology, which does not have a clear definition nor a defined architecture. However, BOS does have a clear objective, which consists of providing adaptability to buildings systems (Vermorel, 2020).

In this paper, we address the central issue of BAS by proposing a knowledge representation named Building Brain Knowledge Representation (B^2rainKR) incorporated as a module in a unique BOS architecture. We called the entire system WITTYM-BOS (W-BOS). W-BOS is an enriching environment composed of modules to manage the flux of information and access to end-users. The function of B^2rainKR is to improve even more the adaptability that BOS already provides. In this way, the system can be adaptable without requiring a deep understanding of the building features. Besides, we propose an innovative cloud-based BOS architecture that harmonizes data flux and creates the most favorable conditions for users to develop applications.

B^2rainKR module represents the operator's knowledge and is made with ontologies that re-use static and dynamic information of the building. The static part is represented by ontologies based on the standard Industrial Foundation Classes (IFC) files (ISO 16739:2018). IFC files contain geometric information of the buildings as well as their elements. The dynamic part is the information coming from the Internet Of Things (IoT). This information populates the ontological representation. Precisely, we re-use existing ontologies such as SAREF4Bldg[1] and Building Topology Ontology (BOT)[2] ontologies to construct B^2rainKR.

To illustrate how W-BOS can deal with an emergent issue, we evaluate a use case in our cloud-based prototype. In this experiment, we display five single steps to create a "hello-world" application that solves the use case issue.

The rest of the paper is as follows. Related work is described in section 2. Our approach is explained in section 3. The procedure to create applications and interfaces is illustrated in section 4. Finally, section 5 discusses and concludes the paper.

2 RELATED WORK

Research in BOS systems focuses on opening systems to end-users not only as consumers but also as developers. BOS proposal approaches have firstly defined BOS as middleware hardware and software placed in the fog computing layer. However, nowadays, new cloud services and the boom of IoT have extended and re-located BOS.

The author (Fierro and Culler, 2015) introduces XBOS as an open-source system, aiming to monitor real-time conditions in buildings and control actuators. Similarly, (Dawson-Haggerty et al., 2013) introduces BOSS, an approach that monitors and controls field devices. These works describe complete architectures to simplify the information flux from IoT to the application layer. However, they neglect the openness aspect, not allowing clear access to data. Besides, these systems are presented as fully middleware APIs.

A step forward is presented in the work of the authors (Kciuk, 2014), (Rosen et al., 2004) and (Wenjiang et al., 2009), they introduce OpenWRT, HomeOS, and VxWorks systems respectively. OpenWRT is a GNU Linux environment aimed to extend the services of a communication router. In this way, extending the communication purpose of the router into an administrator of tasks. HomeOS is a system part of Microsoft research aimed to optimize tasks in house applications. In other words, it seeks to automatize domotic tasks. HomeOS points out to reduce human intervention in automation systems, and it also includes a space for applications. VxWorks' approach aims to exploit real-time data coming from sensors. The architecture of VxWorks allows users to create subroutines dedicated to specific tasks. From a global perspective, these systems enable end-user to intervene and create dedicated services. However, these architectures do not conceive important aspects like integration of data, security, and scalability. In the case of HomeOS, the exciting aspect is the conception of a set of re-configurable applications. HomeOS marks a milestone towards BOS and open services for buildings.

Nowadays, cloud services are mature enough and stable to host complex applications that require high processing skills. It is the case of BIM; some proposals described in (Dave et al., 2018) and (Döllner and Hagedorn, 2007) proposes to extend the web

[1] https://saref.etsi.org/saref4bldg/v1.1.2/

[2] https://w3c-lbd-cg.github.io/bot/

BIM platforms to display and handle IoT data. These works aim to open this information to a collaborative environment.

BIM is closely associated with BOS. They share the same common interest, which is buildings. BIM platforms are excellent ways to display data in 3D or 2D virtual models. The authors (Dibley et al., 2012) and (Curry et al., 2013) propose to use BIM as an extended BAS. In their work, they present a semantic model to integrate IoT data into BIM. The idea is to exploit the geometrical description of BIM with current information coming from the physical building. Despite this enriching integration, there is still an issue concerning complexity. BIM complexity is challenging itself. These platforms already integrate data of different users in a collaborative environment, including administrative information and different format files (images, diagrams, or pictures.). Adding more complexity to BIM can disrupt its scalability. Recent commercial proposals have described architectures considering BIM as part of BOS to display data (Anonymous, 2019).

Buildings are intricate and large structures composed of many spaces, elements, and devices. Over time, ontologies have been widely used to represent buildings' knowledge and leveraged to solve dwellers' most common issues such as thermal comfort, energy consumption, and personal security in many kinds of research. The author (Compton et al., 2012) proposes Semantic Sensor Networks (SSN/SOSA) represent devices in buildings like IoT. In their work, these authors point out to use SSN and reasoners to manage building. Nevertheless, the scope of this ontology is limited to observation and actuation, it lacks Spatio-temporal concepts. These are limited, and there is no clear definition of the most common building elements and their properties like unities or types. The authors (Rasmussen et al., 2018) have combined SSN and BOT to complete the geospatial gap of SSN. However, these approaches lack of expressiveness of BOT about building materials, dimensions, and the topology itself. Besides, the authors have increased non-standardized terminology to describe the missing part to deal with this gap. The authors (Balaji et al., 2016) propose using Brick Schema Ontology (BRICK) to represent building knowledge. Brick possesses a rich vocabulary of building elements. This ontology is oriented towards design and construction (Esnaola-Gonzalez et al., 2016). Although Brick overcomes and achieves a good description, but it does not describe relations between physical spaces. The author (Schneider, 2017) proposes to complete this gap by evaluating Brick's alignment to BOT, demonstrating consistency between them. IFC-based approaches emerge as a plausibly way to standardize building vocabulary in ontologies involving building features. For instance, the author (Esnaola-Gonzalez et al., 2016) argues that IFC is being widely implemented in Building Information Modeling (BIM) field and has been accepted by most building designers, so building development should be close tied with IFC. The IfcOWL ontology proposed by the author (Pauwels et al., 2017) takes EXPRESS schemes from IFC to model the representation of the building. The transformation from IFC files into ifcOwl has been suggested by (Beetz et al., 2009) in its work, the author proposes a semiautomatic transformation. Although ifcOWL contains high alignment with the IFC standard, the ontology is complex and includes many classes that might not be useful to represent the context of devices. The author (de Farias et al., 2015), proposes IfcWoD as an adaptation between IFC and OWL that allows full exploitation of OWL. IfcWod aims to reduce data redundancy by avoiding some classes of IfcOwl. This approach is interesting to reduce the complexity of IfcOwl. However, our objective is to construct an object-oriented knowledge representation. In this way reduce even more complexity of IFC to OWL transformation. The author (Daniele et al., 2015) proposes the SAREF4Bldg ontology as an IFC-based approach to describe building physical elements. This ontology extends SAREF ontology by adding 72 classes that include building objects. It is a summarized version of ifcOWL that only takes building elements. IFC files contain implicit information of building topology like adjacency of spaces. This fact is interesting to exploit but not necessarily practical. The study of the author (Schneider, 2017) reveals an alignment of SAREF4Bldg with BOT showing high compatibility and simplicity compared with IfcOWL. We remark that our work aims to profit from the most recent technology to construct a simple and effective architecture located entirely at the cloud computing level. Moreover, we do not consider BOS as only a middleware layer. Instead, we believe that the functions of BOS must go further and improve the understandability of the building. We strongly believe that transferring knowledge to a BOS system is crucial to achieve full adaptability. Otherwise, W-BOS would fall into a miscellaneous misconception of smart systems similarly to domotic.

3 W-BOS FRAMEWORK

W-BOS is an architecture composed of five interconnected modules in opposition to monolithic systems developed in this domain. The main objective of a modular BOS is to guarantee interoperability between systems and the extensibility of services. In this way, it is possible to replace one module without affecting the others. Beyond, we aim to create the most favorable conditions to create applications by enabling access to data and knowledge through a set of APIs functions. Additionally, our system counts with extra tools to become even easier to use this knowledge (e.g., reasoners). W-BOS profits from recent mature technology such as cloud services and IoT. Thanks to this groundbreaking technology, we propose a cloud-based solution for buildings. As illustrated in figure 1, W-BOS is composed of five autonomous modules divided into two groups management and knowledge modules.

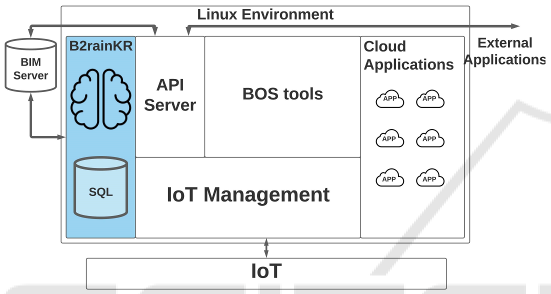

Figure 1: Modular W-BOS architecture, IoT Block and BIM server.

3.1 Knowledge Modules

These modules are tools and knowledge-based information. The main target is to transfer the operator's knowledge to the W-BOS system, which is challenging, considering that the operator had to build this knowledge by hard training.

To describe buildings' features based on a common vocabulary of elements, equipment, and type of spaces can be tedious and need high expertise. Even in this way, the description can be imprecise. To avoid misinterpretation and allow interoperability between systems, a standard reference is needed. The standard IFC emerges as a plausible solution to define and name with high accuracy building elements. Our aim with these modules is to move towards open systems following a standard guideline.

The representation of the static elements of a building gives good spatial information. However, this is not enough to construct knowledge. Buildings are dunk in data that need to be taken into account. Related work trusts on complex multi-protocol devices to obtain this information from various field devices. Nowadays, the use of IoT has simplified this. Many IoT are used as a direct source of data and as gateways to homogenize communication. IoT are a direct source of dynamic data that completes the representation of knowledge.

The following subsections explain how the static and dynamic information is combined in a knowledge representation named Building Brain Knowledge Representation (B^{2}rainKR).

3.1.1 B^{2}rainKR Static Components

The static component of B^{2}rainKR is IFC information that contains all physical elements of a building and their relations between them. The IFC standard has the two most used versions IFC2x3 and IFC4. The evolution of objects has allowed buildings to get more complexity, as reflected in these two versions. IFC2x3 describes elements as a type of one single object. For example, in IFC2x3, a boiler is described as a type of energy conversion device(ifcEnergyConversionDevice). On the other hand, IFC4 contains more precise description of objects such as boilers(ifcBoiler), heaters(ifcSpaceHeater), solar panels(ifcSolarPanel) among others. This fact increases the complexity of interoperability between other systems. For instance, the standard IFC2x3 and IFC4 are widely used by Building Modeling Applications(BIM), and the transition between IFC2X3 towards IFC4 is still in process. To construct knowledge over these two versions of IFC, we use SAREF4Blds ontology. SAREF4Blds ontology contains classes of building objects based on the IFC4 version. Besides, we propose the alignment of IFC2x3 with SAREF4Blds, as shown in figure 2.

Beyond the description in SAREF4Blds of building elements like doors, walls, or electronic equipment, there is still a lack of relation between spaces, as stated in previous work. Additionally to SAREF4Blds, we use the alignment with BOT ontology proposed by the author (Schneider, 2017). The idea is to use relations like "bot:adjacentZone" and "bot:intersectsZone" to complete the description of the building. The combination and alignment are illustrated in figure 2. A significant advantage of SAREF4Bldg over ifcOWL and ifcWod is its object-oriented aim, which is particularly useful and compatible with W-BOS's goal to decrease the complexity of the entire system, even the knowledge representation. On the other hand, it is also possible to extract some classes or re-use ifcOWL and ifcWod partly. However, the question is how to discriminate the classes and properties. A work that has already been done in SAREF4Bldg ontology. Besides, SAREF4Bldg's versatility allows the interoperability between the two IFC versions(IFC2x3 and IFC4).

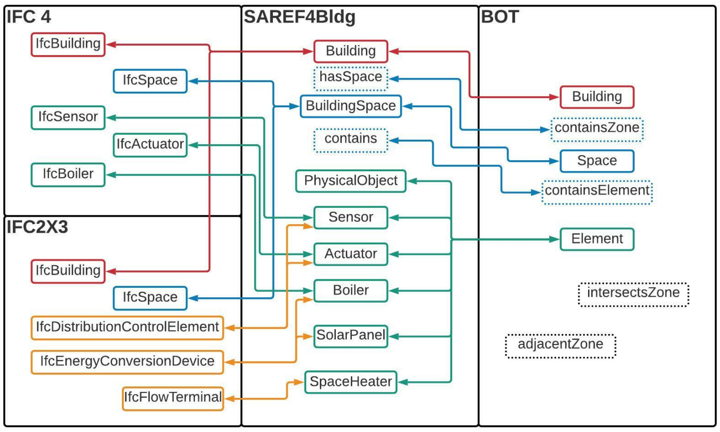

Figure 2: The windows IFC4, IFC2x3, and SAREF4Bldg represent the alignments between IFC and SAREF4Bldg concepts. The two-sense arrows indicate the equivalence, and the color code represents main concepts such as Building (red), Spaces (blue), and Objects(green). In IFC2x3 as no direct or explicit equivalence, the yellow block contains the general ancestor of IFC2x3 objects. The windows SAREF4Bldg and BOT display the alignment between them and the properties (dotted blocks).

In our prototype, we use an IFC2x3 file that belongs to the description of our laboratory building. Figure 3 shows the real physical building, and its 3D digital model using WITTYM-BIM[3] application. The WITTYM-BIM is a collaborative web-based application that supports the actors of the building construction domain by reproducing a digital twin of the physical building.

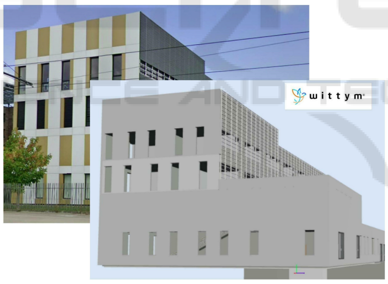

Figure 3: Physical building of our laboratory CIAD and its the 3D representation using WITTYM-BIM application.

3.1.2 B^{2}rainKR Dynamic Components

The dynamic part of B^{2}rainKR is heterogeneous data coming from IoT. The ontologies are suitable to integrate these data. An example of the manner an integration is made in B^{2}rainKR is depicted in figure 4.

Figure 4 shows how a measurement is related to a space, a sensor, and the features of the measurement itself. To make this integration possible, we store the data into a relational database (SQL), aiming to store row data and enable non-knowledge-based applications. The data coming from IoT contain metadata attached to the measured value, independently

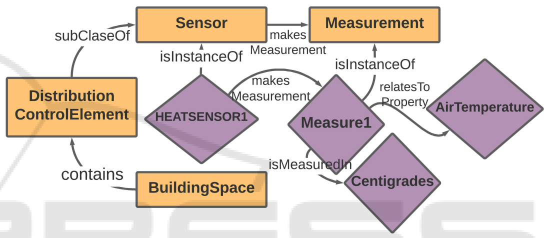

Figure 4: Part of T-Box and A-BOX from sensors and their measurements extracted form B^{2}rainKR.

from the communication protocol (MQTT, HTTP, and LoRWan[4]). This extra information (ID and TimeStamp) is also used to create many instances of sensors and measurements in the ontology. The figure 5 displays B^{2}rainKR representation in protege.

Figure 5: B^{2}rainKR in protege, left side the classes hierarchy of SAREF4Bldg. In the right part, the building space as instances of the class Building Space.

[3]https://wittym.com/

[4]https://lora-alliance.org/about-lorawan/

3.1.3 BOS Tools Module

The BOS tools module contains a set of tools to facilitate building applications resources to construct applications. One primary tool is the embedded reasoner. This semantic reasoner is available for inferring new data, which is helpful in the cases where the inference is required to explain the physical description of the relation between sensors and building elements. The reasoner translates knowledge and makes it easily understandable for humans. For example, a temperature sensor installed on a partition of a wall must be reflected by the temperature of the room (and not of the partition). It is the inference mechanism that finds the space associated with the partition. In this example, the inference would use the relation "saref4Bldg:contains" between the building space and "saref4Bldg:sensor". In our prototype, we embedded a Pellet-based reasoner (Openllet[5]).

3.2 Management Modules

The management modules focused on opening access to W-BOS to end-users. Therefore, easy access can be translated into better adaptability of the system.

3.2.1 IoT Management

The IoT management modules represent a subsystem that handles IoT connections, authentication, and communication. In our prototype, we have implemented a python-based client that holds a register of IoT device topics and identification codes. This module receives and then transfers data to the storage module. Additionally, it also re-transmit commands coming from the cloud applications. In our prototype, we have used experimental IoT devices connected to an MQTT cloud broker that transfers all IoT data to W-BOS.

3.2.2 API Server

The objective of the API server module is to break with silos of data by unifying information in B^2rainKR and providing access to it. As if it were a bus of communication between applications and data. This module is critically essential for W-BOS. Its primary function is to guarantee access and provide a consistent flux of information. In our prototype, we implemented a Swagger[6] API server that performs SPARQL and SQL queries over B^2rainKR by using RFDlib[7] libraries. Besides, this module provides information to the Wittym-BIM platform to display applications and IoT data. These APIs are also enabled for external applications that might require more processing skills. The API module has five functions described below:

- Add rules into B^2rainKR and extract inferred axioms
- Request the spatial measurement made by sensors (by sensor ID)
- Request spatial adjacency of measures or actuation (by BuildingSpace)
- Request positions of sensors (by sensor ID)
- Request status of the building element (by data properties)

3.2.3 Cloud Application Space

The cloud applications module is a specific space dedicated to the development of applications. In this way, end-user can adapt the system to solve new issues in new scenarios. This module profits from a Linux-based environment over which applications can be made using shell or high-level languages like python.

3.3 Field Devices and BIM Server

The prime function of IoT devices is to capture information from the physical environment. They compose the field layer. The hardware in our prototype is IoT devices composed of sensors and actuators. These devices count with Wi-Fi connection over which we implemented the MQTT protocol. MQTT is a topic-based protocol consisting of two main components: A broker aimed to administrate topics and subscribers. On the other hand clients can publish (transmit information) and subscribe (receive information) over some determined topics. Our IoT plays the role of clients and subscribers to transmit telemetry data and receive commands from W-BOS through a broker.

4 W-BOS USE CASE EXAMPLE

This section presents how an application can be developed in W-BOS. We create the Internet of Things Fire Detection Application (IOTFDAPP) as a "hello-world" example. The aim of showing this example is to demonstrate that W-BOS efficiently exploits the knowledge of the static elements and dynamic information. IOTFDAPP's objective is to determine the presence of fire in a room and the surrounding spaces.

[5]https://github.com/Galigator/openllet
[6]https://swagger.io/
[7]https://rdflib.readthedocs.io/en/stable/

The result of IOTFDAPP is the names of room spaces that surround a room with fire. In our case, we use the model of our laboratory building. This use case can be helpful for firefighters to evacuate people in a building. For this purpose, we employed 10 IoT physical devices installed in rooms of our laboratory. We enable the API module to exchange information between the Wittym-BIM platform and W-BOS to display IoT data. The idea is to represent the result of the IOTFDAPP application as a virtual sensor that contains the names of these dependencies(Building spaces like rooms). For this purpose, we follow the following five steps required to create an application in W-BOS:

- 1.Insert a rule, perform semantic reasoning and query the ontology with the inferred axioms. In this use case, the following rule describes our desire: The rooms surrounding a space that reaches a temperature higher than 50[Centigrades] must be evacuated. Table 1 depicts this rule. The result of the application is displayed in figure 6.

Table 1: Ontology rule in the API.

Ontology Rule
s4b:Sensor(?s)∧s4b:Measurements(?m)∧ s4b:makesMeasurement(?s,?m)∧ s4b:relatesToProperty(?m,AIRTEMPERATURE)∧ s4b:hasValue(?m,?v)∧bot: Space(space?)∧ s4b:conatins(?space,?s)∧ bot:adjacentZone(?space,?adjacent)∧ swrlb:greaterThan(?v, integer 50) → s4b:isluminated(?adjacent, true)

- 2.Request the result with the ”isIluminated” property that has a Boolean value of ”true”.
- 3.Creates a virtual sensor label to transmit the result into W-BOS.
- 4.Create a virtual sensor in WITTYM-BIM
- 5.Request the API with the ID of the sensor. The result of the application is displayed in figure 6.

5 DISCUSSION AND CONCLUSIONS

This paper introduced W-BOS, a BOS architecture doted with a knowledge representation that contains static and dynamic information of the building. W-BOS aims to solve the central issue of building automation by blending knowledge in a unique and open architecture. We built a cloud-based prototype to demonstrate W-BOS's advanced adaptability. Specifically, we have evaluated a ”hello-word” use case.

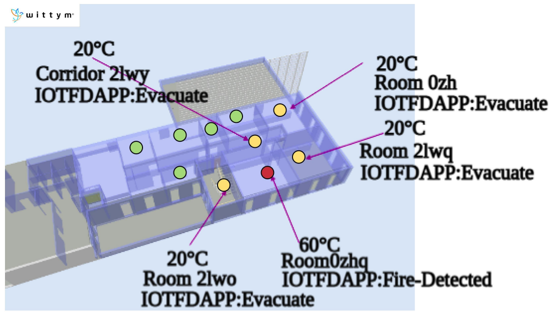

Figure 6: Result of IOTFDAPP displayed in WITTYM-BIM. The plan in 3D belongs to the rooms of our laboratory building floor.

We believe that our prototype covers essential functions to reach a high level of adaptability. Nevertheless, a cloud solution already provides easy access to information in geographical terms. Our proposed architecture structures the data and their flux. By using knowledge, we firmly believe that the system is more understandable and thereby easy adaptable. It is possible to conceive the underlying operating system (Linux) of W-BOS as the BOS itself. However, related work reveals that data arrangement and flux are more significant than the programming language or the underlying operating system.

Beyond this evidence shown in this paper, there is growing concern about the security of data that must be addressed. Targeting this issue, we presume that using knowledge can limit access to private user data by creating rules according to the user's privacy and security policy.

A relevant challenge is a paradigm of combining IoT and BIM information. To create B^2rainKR we transformed IFC2x3 to SAREF4Blds manually. Although this method is valid, the task of transforming IFC information manually into ontologies is inefficient. An automatic transformation can increase the scalability of the system and simplify this process. Despite this gap, we could describe most building dependencies, sensing, and actuating devices without creating new classes.

In the W-BOS prototype, we could quickly request the knowledge using the API module to develop IOTFDAPP. Following the five steps, this task becomes easy. We think that more functions can make this adaptation even easier. However, creating more functions without a standardized guideline can reduce the understandability of the system. Despite, this need we could observe that the API facilitates access to information by simplifying all complicated SPARQL queries to the ontology model.

Our work paves the way for developing applications based on knowledge rather than only information. We believe that by opening and facilitating the control of building systems, we can count on more appealing and efficient solutions.

ACKNOWLEDGEMENTS

This project is funded by the KID'S AI Company. We thank the members of this company involved in the WITTYM project and especially the engineer's team for providing insights.

REFERENCES

Anonymous (2019). Etude de cas: le futur campus d'emlyon business school - urban practices. https://urbanpractices.com/etude-de-cas-le-futur-campus-demlyon/. (Accessed on 05/05/2021).

Balaji, B., Bhattacharya, A., Fierro, G., Gao, J., Gluck, J., Hong, D., Johansen, A., Koh, J., Ploennigs, J., Agarwal, Y., et al. (2016). Brick: Towards a unified metadata schema for buildings. In *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments*, pages 41–50.

Beetz, J., Van Leeuwen, J., and De Vries, B. (2009). Ifcowl: A case of transforming express schemas into ontologies. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AI EDAM*, 23(1):89.

Buckman, A. H., Mayfield, M., and Beck, S. B. (2014). What is a smart building? *Smart and Sustainable Built Environment*.

Compton, M., Barnaghi, P., Bermudez, L., Garcia-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., et al. (2012). The ssn ontology of the w3c semantic sensor network incubator group. *Journal of Web Semantics*, 17:25–32.

Curry, E., O'Donnell, J., Corry, E., Hasan, S., Keane, M., and O'Riain, S. (2013). Linking building data in the cloud: Integrating cross-domain building data using linked data. *Advanced Engineering Informatics*, 27(2):206–219.

Daniele, L., den Hartog, F., and Roes, J. (2015). Created in close interaction with the industry: the smart appliances reference (saref) ontology. In *International Workshop Formal Ontologies Meet Industries*, pages 100–112. Springer.

Dave, B., Buda, A., Nurminen, A., and Främling, K. (2018). A framework for integrating bim and iot through open standards. *Automation in Construction*, 95:35–45.

Dawson-Haggerty, S., Krioukov, A., Taneja, J., Karandikar, S., Fierro, G., Kitaev, N., and Culler, D. (2013). {BOSS}: Building operating system services. In *10th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 13)*, pages 443–457.

de Farias, T. M., Roxin, A., and Nicolle, C. (2015). Ifcwod, semantically adapting ifc model relations into owl properties. *arXiv preprint arXiv:1511.03897*.

Dibley, M., Li, H., Rezgui, Y., and Miles, J. (2012). An ontology framework for intelligent sensor-based building monitoring. *Automation in Construction*, 28:1–14.

Döllner, J. and Hagedorn, B. (2007). Integrating urban gis, cad, and bim data by servicebased virtual 3d city models. *Urban and regional data management-annual*, pages 157–160.

Esnaola-Gonzalez, I., Bermúdez, J., Fernandez, I., and Arnaiz, A. (2016). Eepsa as a core ontology for energy efficiency and thermal comfort in buildings. *Semantic Web*, 1.

Fierro, G. and Culler, D. E. (2015). Xbos: An extensible building operating system. In *Proceedings of the 2nd acm international conference on embedded systems for energy-efficient built environments*, pages 119–120.

ISO9241-210 (2019). Iso - iso 9241-210:2019 - ergonomics of human-system interaction — part 210: Human-centred design for interactive systems. https://www.iso.org/standard/77520.html.

Kciuk, M. (2014). Openwrt operating system based controllers for mobile robot and building automation system students projects realization. In *15th International Workshop on Research and Education in Mechatronics (REM)*, pages 1–4. IEEE.

Martirano, L. and Mitolo, M. (2020). Building automation and control systems (bacs): a review. In *2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe)*, pages 1–8. IEEE.

Pauwels, P., Zhang, S., and Lee, Y.-C. (2017). Semantic web technologies in aec industry: A literature overview. *Automation in Construction*, 73:145–165.

Rasmussen, M. H., Frausing, C. A., and Karlshøj, C. A. H. J. (2018). Integrating building information modeling and sensor observations using semantic web. In *SSN@ ISWC*, pages 48–55.

Rosen, N., Sattar, R., Lindeman, R. W., Simha, R., and Narahari, B. (2004). Homeos: Context-aware home connectivity. In *International Conference on Wireless Networks*, pages 739–744.

Schneider, G. F. (2017). Towards aligning domain ontologies with the building topology ontology. In *Proceedings of the 5th Linked Data in Architecture and Construction Workshop (LDAC 2017)*.

Sleman, A. and Moeller, R. (2011). Soa distributed operating system for managing embedded devices in home and building automation. *IEEE Transactions on Consumer Electronics*, 57(2):945–952.

Vermorel, L. (2020). What is a building operating system? https://blog.wattsense.com/building-management/what-is-building-operating-system/#:~:text=A%20Building%20Operating%20System%20is,data%20from%20the%20building's%20equipment. (Accessed on 04/29/2021).

Wenjiang, L., Nanping, D., and Tongshun, F. (2009). Design of the embedded remote monitor system for building automation system based on the vxworks. In *2009 Asia-Pacific Conference on Computational Intelligence and Industrial Applications (PACIIA)*, volume 1, pages 436–438. IEEE.