Building Information Modelling for Building Control

Renato Filipe Vieira
renato.vieira@ist.utl.pt

Instituto Superior Técnico, Lisboa, Portugal

May 2015

Abstract

Despite the emergence of Building Information Modeling (BIM) standards, not all dimensions of construction are supported to the same extent. Such is the case of the Building Automation (BA) dimension, where only aspects related to mostly physical setup of devices are supported, which are largely insufficient to enable modeling automation scenarios. BA requires modeling a richer set of aspects, still very heterogeneous, without becoming bound to a specific technology upfront. BIM standards such as Industry Foundation Classes (IFC) position themselves as natural candidates for modeling and exchanging information regarding BA. However, the extent to which BIM supports automation aspects has never been rigorously analyzed.

This work explores the hypothesis of extending BIM to support BA concepts, proposing a new extension to the IFC model, based on an assessment of scientific and technical literature. This study elicits the information requirements of BA and performs a gap analysis with current BIM standards such as IFC, upholding the proposal of a set of new concepts, using model-driven techniques, which enable IFC to exchange automation scenario data. This approach is validated for completeness using model transformations.

Keywords: Building Information Modeling, Building Automation, Model-Driven Engineering

1. Introduction

Efficient Facility Management (FM) is becoming an increasingly important topic that relies on Information Technology (IT) to support the achievement of energy and managerial efficiency goals [23], supported by tools such as Computer-Aided Facility Management (CAFM) and Computer-Aided Maintenance Management (CAMM) as well as tools for Building Automation (BA) and Energy Management Systems (EMS). Ideally, the operation of these tools should be supported by a common representation as Building Information Modeling(BIM) [1, 10]. Although some progress has been accomplished in making CAFM and CAMM applications interoperable with BIM models [8, 34] BA and EMS still lack in terms of integration with BIM. The focus of this work is to establish such connection, proposing an extension that is able to express the requirements of heterogeneous BA, enabling distinct tools to share details regarding the configuration and setup of automation systems through BIM models.

A BAS consists of a network of sensors, actuators and controllers that collect data and manage the operation of a wide span of electric equipment, such as Heating, Ventilation, and Air Conditioning systems (HVAC), or lighting, according to a set of control procedures previously configured. BASs also enable monitoring of equipment and building performance as well as remotely managing electric loads to improve operation and energy efficiency [21, 24]. Information regarding structure and implementation of a BAS—network topology, device configuration, control loops—is very complex to grasp from the documentation delivered to the building manager when the system is installed [30]. Despite the fact that a part of this information is available, the implementation details are frequently not understood by the building owner nor by technical maintenance personnel. In fact, only the vendor can comprehend the entire BAS, making the building owners highly dependent on their contractors whenever they want to modify, commission, or tune the BAS [29]. Ideally, this information should be represented in a BAS- independent way, enabling facility managers to easily contract system alteration, hiring specialists, and using third- party tools or technologies.

BIM supports a more efficient management of facilities since all building information, from design to operation, is contained in a shared digital repository [5]. Thus, a BIM-supported FM is more effective, once the information is promptly available to the applications and more efficient, since it is shared across the building's peers avoiding losses, replication and incoherence of data [2, 26]. BIM focuses on integrating all the data concerning building's life cycle in one digital single model, providing a cross-entity representation of it, and making it easier for every stakeholder to get involved on each of the Architecture, Engineering, Construction and Operation (AEC/O) phases [12, pp.1]. Nevertheless, this approach is still deeply connected to the structural phases of the building life cycle, such as Architecture, Engineering and Construction in sharp contrast with several limitations of the Operational phase, specially in what concerns modeling and sharing information regarding BA elements.

The lack of connection between AEC and Operation dimensions is mostly due to two factors. The first has to do with the targeting of efforts to the design phases, where Computer Aided Design (CAD) software vendors play an important role on influencing research and development [17]. The second has to do with the automation information itself, that is of a more volatile nature than information regarding other constructive elements (characteristics and arrangement of automation elements are subject to change faster than the constructive elements). Although BIM exchange formats, such as Industry Foundation Classes 4 (IFC)[1], already feature constructs for the BA domain, these are only used for delivery of information such as installation instructions or device specification. Efforts are being made for integrating BAS data into BIM's exchange format, but the information supported by the current BIM standard is limited only to the system structure, like devices and wiring relations, disregarding the logical and operational dimensions, such as control loops, bindings or configuration management [8]. As will be clear later, data regarding BAS in BIM is still largely incomplete to enable interoperability with BAS tools, negatively impacting operation efficiency [20].

Model-Driven Engineering (MDE) provides a non-traditional approach to system development. MDE conceives the domain model as the central concept for developing a system, separating the formal description of the domain from the concrete platforms on which it will be built[11]. In other words, MDE enables the independent specification of the functionality, abstracting from the concrete idiosyncrasies of the target platform on which the system is implemented. This allows the same model to be implemented on several infrastructures, through the specification of mappings between the model and target platforms. Since the model specification is independent from specific implementation, and the mappings handle the conversion between platforms, different platforms can interoperate more easily, sharing a common representation[28, 32]. In a practical sense, MDE eases system development due to, amongst others, the following aspects. Firstly, the correctness of the core domain (model) of the system is more easily evaluated since the functionality is isolated from the implementation. Secondly, the generation of implementations under several platforms is semi-automated, saving time. Thirdly, the integration of different platforms is more simple to achieve through the usage of a common, independent model representation. Finally, the domain is more easily maintainable, since changes are executed in the model and easily propagated through target platforms[22].

Despite being developed as a data exchange format, IFC can be perceived under the light of MDE[15]. Since IFC is an independent model for representing building data, model transformations can be specified, using MDE techniques, in order to map IFC constructs into different platforms such as BA protocols or commissioning tools instead of serving as input for applications which have to conform with its specification[19]. Using this approach, further IFC versions can evolve independently of the target platforms, as long as the model transformations remain coherent, thus favoring maintainability. Also, the target platforms can be fed by information contained in IFC models without having to implement IFC specifications.

The convergence between BIM and BA worlds using model-driven techniques opens an array of interesting possibilities regarding building efficient operation such as (i) reducing scattering and improving consistency of BA data, (ii) favoring interoperability between BA tools which reduces maintenance costs and finally, (iii) improving the independence of the representation of BA data, eventually leading to alleviating customer lock-in. Studies have suggested the development of extensions to BIM [25, 31, 34] and also of platforms that integrate information with it for use in applications related to buildings [3, 6, 35]. There are also, several initiatives relying on MDE for interacting with BIM [7, 15, 16, 16]. This underscores the importance of BIM as interoperable source of data. However, IFC version 4 elements concerning BAS do not fully cover the entire BA domain concerns. In addition, the only tools available for realizing, analyzing and managing BAS are proprietary, and do not support BIM, keeping FM inefficient and locked-in to tool vendors, that are the only ones that can understand the details of Automation Systems and manage their implementation [9]. Interoperability of BA with BIM would, therefore, contribute to more efficient building automation.

This work presents a new BIM-based standard for modeling BA requirements in a vendor-independent manner. Such approach will enable the develop-

[1]http://www.buildingsmart.org/standards/ifc

ment of open tools to manage, control, and analyze BA systems. Indeed, the results presented herein are far-reaching as they contribute to alleviating the customer dependency from the vendor that installed the systems. The model extension developed in this work will be validated using a model transformation mechanism, mapping it into the BA protocols studied, in order to prove the completeness and equivalence of the extension, facing the abovementioned BA protocols.

2. Background

This section presents a series of relevant definitions for understanding the problem discussed. Firstly, we are going to describe BIM, making a brief analysis to its motivations and features. Secondly, some concepts regarding BAS will be visited. Thirdly, this work explores the data that characterizes the BAS features. Finally, the most relevant aspects related to Model-Driven Engineering are visited.

2.1. Building Information Modelling

BIM is an open technology for digital building modeling. The National Institute of Building Sciences (NIBS)[2] describes BIM as *an improved planning, design, construction, operation, and maintenance process using a standardized machine-readable information model for each facility, new or old, which contains all appropriate information created or gathered about that facility in a format usable by all throughout its life cycle"* [27].

The motivation for BIM has sprung from the need to simplify the process of passing the information between activities in building AEC/O phases. Overall, BIM rests on three main aspects. First, BIM stores information in a digital representation, that reduces costs of paper support, while reducing the complexity inherent to the construction process and making it more agile. BIMs also serve as unified data sources that centralize all relevant building information. A computational model containing the facility's data enables integration of a wide variety of tools that modify and collect data on the model, enabling several kinds of analysis. One common example is energy simulation [4].

The second aspect of BIM is that digital models may be shared and updated concurrently. Distinct stakeholders can insert, update or delete information from it, contributing to the enrichment of the model. The recurring example is the architect adding sketches of the building, the structural engineer modifying it to comply with engineering requirements, the builder using it to plan the construction phase, and the facilities manager analyzing it to operate the facility properly [5, 18, 35].

Finally, the third aspect is that BIM organizes information in a structured way. This contributes to making processes more consistent and also to preserving the information coherence, preventing data quality issues. Moreover, structuring automates the analysis of the models, favoring error traceability. For example, it is possible to query the model to check for collisions between different components or construction errors, like an air duct colliding with the building roof [12, 33, pp. 21-22, 24].

[2]http://www.nibs.org/

2.2. Building Automation

A BAS connects electrical and mechanical components in a facility enabling them to interact. Typically a BAS is responsible for managing HVAC, lighting, among other aspects and possibly interacting with other systems, such as access control and fire safety [21]. A large number of concepts have to be analyzed in order to support BA aspects in BIM.

With respect to the conceptualization of a BAS, it is important to define upfront the difference between the concepts of *equipment* and *control device*. This distinction is as follows:

Equipment stands for a part of, or an entire electrical or mechanical system, that interacts with the building environment, namely HVAC systems, electrical blinds, or luminaries. Equipement can be atomic, or composed of simpler equipment. This composition is depicted in the Figure 1.

Control devices consist of equipment with logical behavior in a BAS control network. Sensors, actuators, meters, controllers, or gateways are examples of control devices. They can sense, act and execute logic on the environment directly or on another equipment, for example, a sensor detecting movement on a given room or a dimmer controlling the output of a luminary.

Control devices and equipment in general can be arranged in several ways. For example, there are

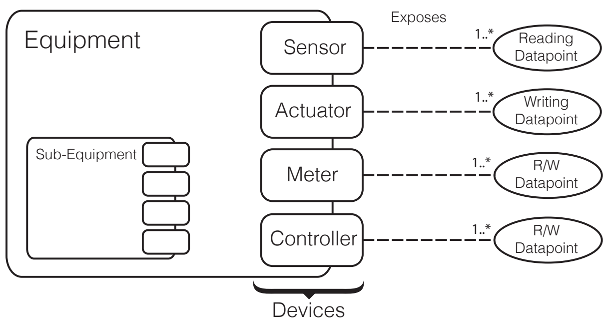

Figure 1: Decomposition of an equipment in smaller equipment or control devices. Each control device exposes one or more datapoints in the BAS network.

devices that connect externally to another equipment, as is the case of external on/off actuators, while others are part of a complex equipment, such as a fan controller inside an HVAC module. Figure 2 illustrates this idea.

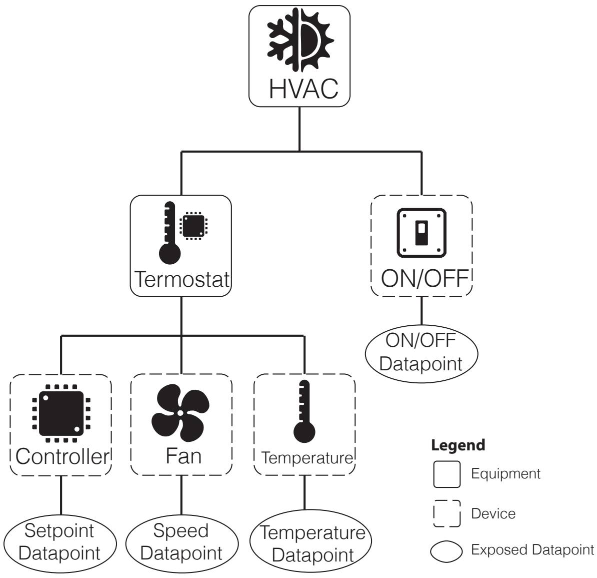

Figure 2: Example of the equipment tree of a simple HVAC split system. It decomposes into a thermostat sub-equipment and an on/off switch control device. The thermostat divides three control devices. All the control devices expose one datapoint.

One BAS can be organized according to three levels: the Field level, the Automation level, and the Management level. The field level consists of equipment that interacts with the building environment, as well as the control devices. Automation level consists of controllers responsible for orchestrating other control devices, according to predefined control-loops, and collecting the data produced. Management Level receives information from controllers and manages their configuration. Management Level also unifies an interface for manual management tasks such as system state monitoring, event logging, manual variable configuration, access to historical data, or alarm notifications. [13, 14, 21].

2.3. Domain Model

This section systematizes the information requirements from the BAS aspects. For each aspect, this subsection defines the information that needs to be captured in order to express it. This information is summarized in Table 1.

Level	*Concept*	*Information required*
Field	Wiring	Device Connection Device Connection Type
	Equipment Specification	Specific Device Information
	Location	Spacial Element Position
	Influence Zone	Influence Area Influence Type
	Influenced Devices	Influence Devices Influence Type
Automation	Bindings	List of bindings
	Datapoints	List of datapoints
	Device/Datapoint Relation	List of Datapoints Relation with devices
Management	Setpoints	List of setpoints Unit of measure Relation with datapoints
	Scenarios	List of scenarios Relation with trigger conditions Relation with datapoints
	Scheduling	List of Schedules Relation with datapoints Relation with date and time
	Groups and Zoning	List of Groups and Zones Relation with datapoints Relation with building spaces
	Alarms and Events	List of events and alarms Relation with datapoints Action to trigger

Table 1: Information requirements for the distinct automation components, organized according to the corresponding BAS level.

<u>Field Level</u>

Wiring relations are described through a list of *device connections*, which represents every link between devices. It is also necessary to specify the *device connection type* of each relation, for example RS232 or RJ45 cable.

Equipment Specification requires detailing the characteristics of each device, namely *specific device information* provided by manufacturers.

Location is supported by data about the *spatial element* where the device is contained, as well as the *position* of the device within that element.

Influence Zone supports the knowledge about the spaces influenced by devices, namely the *influence area*. It is also necessary to know the *type of influence* employed, for example, whether it is of actuation or monitoring.

Influence Devices similarly to influence zones, depend on information about the *device connections*. They are also defined by the *type of influence* exerted.

<u>Automation Level</u>

Datapoints information is captured through the *list of datapoints* that compose the BAS, and its metadata.

Bindings require the specification of a *list of logical connections* between datapoints.

Control Device/Datapoint Relation encodes a mapping between control devices and datapoints. Hence, to express this concept it is necessary to know both the *datapoint list* and the *control device list*, as well as the *mapping between control devices and datapoints*.

Management Level

Setpoints are supported by a *list of setpoints*. For each setpoint it is also necessary to know the *unit of measure* and the datapoint it refers to.

Scenarios are defined by the *list of scenarios*, which define the presets for each datapoint they affect.

Scheduling is based on a *list of schedules*, as well as the datapoints they affect.

Groups and Zoning are defined by the list of group and zone objects and the datapoints they contain. For zoning it is also necessary to capture the spatial elements related to the zone object.

Alarms and Events are described by the list of occurrences, the datapoints related to those occurrences and the action they trigger.

3. Implementation

IFC4Automation is an extension the IFC4 information model, that covers the remaining relevant BAS aspects from automation and management levels, such as bindings, setpoints, schedules, or scenarios, that are not met by the actual specification.

The model defines a set of new objects, relations, and resources that comply with IFC, enabling the reuse of constructs already available, such as equipment details, wiring, or location. In other words, it is designed to make possible the interaction between existing IFC elements and the new ones.

EXPRESS IFC4Automation is developed using EXPRESS, the modeling language used for specifying IFC. Writing an extension using EXPRESS provides a powerful mechanism for expressing all the elements, required without ambiguities, and guarantees the consistency needed in a BIM extension, in the sense that there is no need for translating between different modeling approaches.

MDE IFC4Automation model defines a set of abstract constructs that capture the core information of each BA aspects in order to keep the model independent from the several implementations, detaching the core specification of a BAS from the several technologies in which it could be realized.

EMF IFC4Automation is also represented using the Ecore metamodeling language. Ecore is part of the EMF platform where, in contrast with EXPRESS which is less established in the MDE community, a multitude of open-source tools for handling models and metamodels are already developed and well supported. Once represented as an Ecore model, it is possible for IFC4Automation to interact with a larger variety of MDE tools.

IFC Compliance The structure of the IFC4Automation extension follows the same design principles as the rest of the IFC specification. It consists of a set of entities, relationships and data types, and also predicts the definition of property sets for characterizing each element.

Abstraction Constructs developed in IFC4Automation express each feature in a sufficiently high level manner, enabling the creation of generic BAS domains that are independent from the underlying technology.

3.1. Element Realms

IFC4Automation is arranged into three main realms, that address the three most general concerns of this work, which are the device topology, the structural configuration and the behavioral configuration. This division corresponds to the three main functions that BAS models cover, namely topology, structure, and behavior. The entities that are contained in each of these realms are detailed below.

Device topology realm contains the elements aiming at specifying control devices in terms of their physical and logical structure. The elements that belong to this realm define which datapoints are part of a given control device, how the datapoints are bound to each other, or which physical elements (IfcActuator or IfcController, for example) are related to which control devices.

Structural Configuration realm organizes the constructs that capture the way datapoints can be organized to form a more complex structure. Its elements enable the composition of datapoints in free groups, such as all the HVAC-related datapoints, as well as zoned groups, which have a correspondence with a given physical space.

Behavioral Configuration realm encloses the elements that address the logical behavior of datapoints. These elements enable the definition of configuration setpoints, as well as action sets, which are predefined actuations that can be triggered by human users, environment constraints or even other datapoints.

Table 2 summarizes the elements that constitute the model extension.

Component	*Entity*
(none)	IfcAutomationObject IfcAutomationObjectType
Device Topology	IfcAutomationDatapoint IfcAutomationInputDatapoint IfcAutomationOutputDatapoint IfcAutomationControlDevice
Structural Configuration	IfcAutomationDatapointSet IfcAutomationGroup IfcAutomationZone
Behavioral Configuration	IfcAutomationDatapointWrite IfcAutomationActuation IfcAutomationSetpoint IfcAutomationDevicePreset IfcAutomationEvent IfcAutomationAlarm IfcAutomationScenario IfcAutomationSchedule
Relationships	IfcRelBelongsToDevice IfcRelBelongsToControlElement IfcRelGroupsDatapoint IfcRelTriggersEvent IfcRelInfluencesSpace IfcRelBindsToDatapoint IfcRelBindsByDatapoint IfcRelBindsByDatapointWrite IfcRelLocatesInSpace IfcRelDefinesSchedule IfcRelDefinesDatapointWrite

Table 2: Summary of the elements defined in IFC4Automation organized by type of element and with references for the component in which they are contained.

3.2. Non-functional Requirements

The development of IFC4Automation encompasses the satisfaction of three main non-functional requirements which are (i) flexibility, in order to render the model less sensitive to change and evolution, (ii) complexity, since a simpler model is more easily readable, therefore, more easily used, and finally (iii) interoperability, which is of extreme importance, since this extension aims at being a common representation between several automation technologies. The design decisions that fulfill the requirements above are detailed as follows.

Flexibility is addressed through the organization of the model in realms, which are represented by abstract entities . This separation of concepts makes the development of new entities inside each realm easier, since part of their definition is reused.

Complexity is addressed by circumscribing the inherent complexity of IFC. Firstly, entities are organized in small hierarchies. Another aspect is the definition of a small, but cohesive set of entities to express all the information requirements. Reducing the depth of the inheritance tree and increasing the cohesiveness of the entity specification reduces the effort required by the final user to discern the structure of the model.

Interoperability is achieved in the first place by extending the IFC specification *per se*. Moreover, each IFC4Automation element is designed to contain the attributes that are relevant for describing each aspect without depending on any technology.

4. Results

Since IFC4Automation extends IFC standard specification, it includes the complexity inherent to IFC. Despite is necessary to slightly increase the complexity of the IFC4 by adding 55 new classes, IFC4Automation remains with the same inheritance tree depth. It is also worth noting that this new 55 classes cover the remaining 9 BAS aspects.

In spite of directing efforts to make IFC4Automation a flexible model, this work had also to cope with complexity concerns, which would be greatly impacted if a trade-off would not be considered. However, the measures taken in order to make it a flexible model succeeded, since they make it easier to modify.

Finally, as mentioned earlier, IFC4Automation is the only model that covers all the aspects of BASs, and therefore, is the only model that is suited for modeling BA scenarios on field, automation and management level. Therefore, it can be said that IFC4Automartion covers the whole set of aspects of BAS.

5. Case Study: Sustainability Systems Room

In order to further validate that IFC4Automation is adequate to express automation scenarios that are convertible to BAS installation descriptions (conforming to information models of particular automation systems), this work encompasses the realization of a case study that consists of modeling an automation scenario installed in room 1.58 at IST-T and comparing it with the real installation. This case study evaluates whether the model generated from an IFC4Automation description is able to express the same logic as the currently deployed KNX configuration. This case study uses the Room

1.58, on the first floor of IST-T to test the model. This room is equipped with a KNX installation, and exposes datapoints for controlling mostly lighting, HVAC and blind control.

5.1. Room 1.58 IFC4A model

With respect to the modeling of the automation scenario, the following approach was followed. Each device was modeled as an *IfcAutomationControlDevice*. The name, description and OID were set according with each one of the devices, as Figure 4 depicts. Each datapoint exposed by the each device was modeled, according to the type of datapoint, as an *IfcAutomationInputDatapoint* or an *IfcAutomationOutputDatapoint*, where the attributes OID, name, description, dataType, unit and address were set according to the datapoints characteristics. The dataType attribute is always set to *USERDEFINED* since the list of available datatypes is still an open topic for future work. The individual group addresses for each datapoint were specified in modeling time, since the the installation documentation already specified them. Otherwise they could be generated upon transformation, however, in this case this would led to discrepancy between the addresses from the documentation, and the ones generated by the transformation. The containment relation between each device and its exposed datapoints was modeled recurring to an *IfcRelBelongsToDatapoint* per device. The device was inserted in the device end of the relation and the set of datapoints it exposed were inserted in the set of relating datapoints. For modeling each group of datapoints, such as *stopAllBlinds*, an *IfcAutomationGroup* was instantiated with the name and description attributes set. Similarly to device, each group is related with the datapoints it groups through one relation object, this time *IfcRelGroupsDatapoint*. The group was inserted in the one end of the relation, and the datapoints that compose it were added to the set that figures on the other end of the relation. An excerpt of the resulting model is depcted in Figure 3.

```
#358534= IFCAUTOMATIONCONTROLDEVICE('2080D1F834770559A1F0C6F881AC1BC5',$,'LEDsTV','LED lights around the Television',$,$);
#177761= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.TRUE.,#9376);
#177762= IFCFACE(());
#177763= IFCCARTESIANPOINT((18772.4376267873,18285.8000029244,699.996366118196));
#177764= IFCCARTESIANPOINT((18576.4792073837,17807.1896131715,0.));
#177765= IFCCARTESIANPOINT((2039.11564395107,367.387193323944,681.139998049677));
#177766= IFCCARTESIANPOINT((4083.61112094128,7738.34006281904,75.3299238562959));
#177767= IFCCARTESIANPOINT((15990.2731914418,15660.4230934082,89.3057441356647));
#177768= IFCFACE(());
#337643= IFCCOMPOSITECURVE((),.F.);
```

Figure 3: Excerpt of the resulting IFC4Automation model for room 1.58. The first line depicts the definition of a IfcAutomationControlDevice.

5.2. Generated Installation

As mentioned in the previous section, one of the room 1.58 IFC4Automation representations serialized by the populate script intends to be fed into the *TransformToKNX* transformation. Basically,

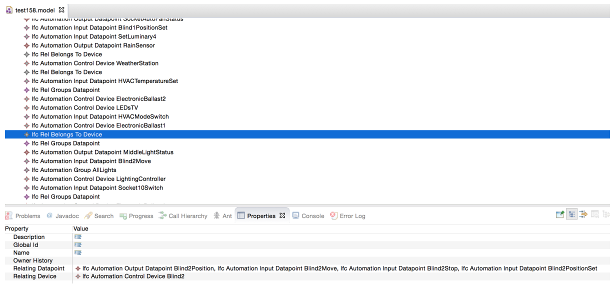

Figure 4: Excerpt of the resulting EMF model for room 1.58. The highlighted construct represents a IfcRelBelongsToDevice, which properties are detailed in the properties tab on the lower section.

this file keeps representing the constructs conforming to the IFC4Automation schema, however, following a XML-based representation, as depicted in Figure 4, where each construct is represented as an element, and each relation as a reference attribute. For taming the complexity of reading and parsing all the constructive elements already represented in the IST-T model, the EMF representation only contains the BA-related elements, that are the ones that are relevant in the following steps.

After executing the transformation, the inputted elements were converted to the following constructs of the KNX domain, as illustrated by Figure 5. Each *IfcAutomationControlDevice* was transformed into a KNX *FunctionalBlock*. The name attribute of each device is mapped into the name of the functional block. Though the documentation of the base installation did not mention their existence, the generated model includes functional blocks as logical aggregations of datapoints, that compose one device. Therefore, each functional block has a reference to each datapoint that it contains. Each *IfcAutomationDatapoint*, whether input or output, generated a KNX *Datapoint* and the respective *GroupObject*. The group object contains relevant network information regarding datapoints, hence, they are created at the same time. The OID, name, description, dataType and unit attributes were directly mapped form each IFC datapoint into each KNX datapoint. IFC input and output datapoints were converted to KNX datapoints with the FlowType attribute set to "INPUT" and "OUTPUT", respectively. Each datapoint has a reference attribute to its respective group object and functional block. Each *IfcAutomationGroup* was converted into a KNX *GroupAddress*. The IFC group name and description were not mapped, since group addresses do not define these attributes. The group address was generated by the transformation, according to an algorithm that increments the group identifier each time a group is converted. Each group address have references to the group objects that represent the datapoints contained in it. It is also worth noting that each group object contains a reference to its main group address, the one that it uses when is necessary to send a package, and a reference to a list of group addresses from which the datapoint listens. In terms of generated objects, the transformation produced a total of 218, where 92 represent BAS aspects such as datapoints, group addresses and functional blocks, and 126 represent auxiliary objects such as group objects and datapoint group addresses.

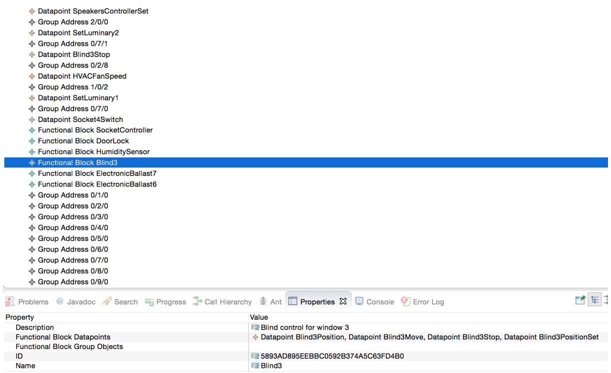

Figure 5: Excerpt of the resulting KNX model for room 1.58. The highlighted construct represents a FunctionalBlock for blind 3, which properties are detailed in the properties tab on the lower section. This object is equivalent to a device.

5.3. Discussion

The analysis of this case study makes possible to draw a set of conclusions regarding the transformation performance. In what concerns to converted constructs, the transformation succeeded in converting all the 92 aspects (devices, datapoints and groups) from the input model into constructs from the output model, hence, it can be concluded that, for this case study, the IFC4Automation extension can cover 100% of the automation scenario of the room 1.58. Besides, the transformation was able to convert all the constructs from the input model into equivalent constructs from the output model, which states that no ambiguities were found in rule execution. It is worth noting that all the possible test cases could not be executed, since the KNX information model does not cover the whole spectrum of BAS aspects, and that the automation scenario from room 1.58 is also unable to exercise all the use cases of the KNX information model. However, evaluating the whole set of use cases that cover the entire KNX or IFC4A information models implies an unpractical effort, out of the scope of this work. In addition, the aspects concerning data type testing and data type validation were not regarded since they are part of a future work, also out of the scope of this research.

6. Conclusions

The emergence of BIM as a central repository for all the building-related data makes possible to manage most aspects of buildings in a more efficient, consistent, and coherent way. However, the coverage of IFC, the standard model specification for BIM, regarding the logical aspects of BA is almost inexistent. Also, there is no model that covers the full spectrum of BAS concepts natively, accounting for platform heterogeneity and vendor independence.

This work explores the connection amid the BIM and BA realms, and proposes IFC4Automation, an extension to IFC that (i) covers the logical components of BAS configurations, abstracting from technological specificities, (ii) enables the definition of interoperable BAS models, that can be refined to installations conforming to specific automation platforms, and (iii) provides a centralized repository for the BAS-related data of facilities.

The proposed extension is evaluated towards completeness, complexity, flexibility, and interoperability, and its practical applicability is stated through a model transformation.

The impact of this work is far-reaching as it draws a path for solving the problems of (i) consistency and coherence of BAS-related data, (ii) interoperability between different BA tools and protocols, and (iii) vendor dependency of the BAS configuration. Interoperability between BA models has been stated as a solution for preventing customer lock-in, and for enabling the development of a multitude of applications that make use of the BAS information for improving, among other FM software.

Acknowledgements

The author would like to thank Professor Paulo Carreira, for all the effort and good will demonstrated in supervising this work. Moreover, a work of gratitude to Instituto Superior Tcnico, which tries everyday to provide the students with valuable knowledge, with rigor and exigence.

References

[1] A. Akcamete, B. Akinci, and J. H. Garrett. Potential utilization of building information models for planning maintenance activities. 2010.

[2] M. Alahmad, W. Nader, A. Brumbaugh, Y. Cho, S. Ci, H. Sharif, J. Shi, and J. Neal. The BIM's 4D+ dimension: Real time energy monitoring. *2011 IEEE GCC Conference and Exhibition (GCC)*, pages 589–592, Feb. 2011.

[3] M. Alahmad, W. Nader, J. Neal, J. Shi, C. Berryman, Y. Cho, S.-K. Lau, H. Li, A. Schwer, Z. Shen, J. Stansbury, and

T. Zhang. Real Time Power Monitoring & integration with BIM. *IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society*, pages 2454–2458, Nov. 2010.

[4] B. alliance. National BIM Standard - United States Version 2. 2012.

[5] S. Azhar, M. Hein, and B. Sketo. Building Information Modeling (BIM): Benefits , Risks and Challenges. 2007.

[6] V. Bazjanac. IFC BIM-based methodology for semi-automated building energy performance simulation. 2008.

[7] J. Beetz and L. V. Berlo. BIMSERVER.ORG An open source IFC model schema. (Weise 2006):16–18, 2010.

[8] C. Bogen, D. Ph, M. Rashid, E. W. East, and F. Asce. A framework for building information fusion. pages 26–28, 2011.

[9] S. T. Bushby. BACnet: a standard communication infrastructure for intelligent buildings. 6:529–540, 1997.

[10] E. Curry, J. ODonnell, E. Corry, S. Hasan, M. Keane, and S. ORiain. Linking building data in the cloud: Integrating cross-domain building data using linked data. *Advanced Engineering Informatics*, 27(2):206–219, Apr. 2013.

[11] V. Devedzic, D. Djuric, and D. Gasevic. *Model Driven Engineering and Ontology Development.* 2009.

[12] C. Eastman, P. Teicholz, R. Sacks, and K. Liston. *BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors.* Wiley. com, 2011.

[13] W. Granzer and W. Kastner. Information modeling in heterogeneous Building Automation Systems. *2012 9th IEEE International Workshop on Factory Communication Systems*, pages 291–300, May 2012.

[14] W. Granzer, W. Kastner, G. Neugschwandtner, and F. Praus. A modular architecture for building automation systems. *2006 IEEE International Workshop on Factory Communication Systems*, pages 99–102, 2006.

[15] a. Grilo. Shifting the construction interoperability paradigm, in the advent of Service Oriented and Model Driven Architectures. *Conference on Information Technology in Construction*, 304:6, 2005.

[16] A. Grilo and R. Jardim-Goncalves. Challenging electronic procurement in the AEC sector: A BIM-based integrated perspective. *Automation in Construction*, 20(2):107–114, 2011.

[17] R. Howard and B.-C. Björk. Building information modelling Experts views on standardisation and industry deployment. *Advanced Engineering Informatics*, 22(2):271–280, Apr. 2008.

[18] U. Isikdag, G. Aouad, J. Underwood, and S. Wu. Building Information Models: A Review on Storage and Echange Mechanisms. 1999.

[19] U. Isikdag and J. Underwood. Two design patterns for facilitating Building Information Model-based synchronous collaboration. *Automation in Construction*, 19(5):544–553, 2010.

[20] A. Karavan and K. Kabitzsch. Merging Building Automation Network Design and IFC 2x Construction Projects. 2005.

[21] W. Kastner, G. Neugschwandtner, S. Soucek, and H. M. Newman. Communication Systems for Building Automation and Control. 93(6), 2005.

[22] S. Kent. Model Driven Engineering. *Integrated Formal Methods, Third International Conference, {IFM}*, 2335:286–298, 2002.

[23] J. Lee, Y. Jeong, Y.-S. Oh, J.-C. Lee, N. Ahn, J. Lee, and S.-H. Yoon. An integrated approach to intelligent urban facilities management for real-time emergency response. *Automation in Construction*, 30:256–264, Mar. 2013.

[24] V. Marinakis, H. Doukas, C. Karakosta, and J. Psarras. An integrated system for buildings energy-efficient automation: Application in the tertiary sector. *Applied Energy*, 101:6–14, Jan. 2013.

[25] P. Meadati. BIM Extension into Later Stages of Project Life Cycle. 2009.

[26] I. Motawa and K. Carter. Sustainable BIM-based Evaluation of Buildings. *Procedia - Social and Behavioral Sciences*, 74:419–428, Mar. 2013.

[27] National Institute of Building Sciences. National Building Information Modeling Standard. Technical report, 2007.

[28] OMG. Object Management Group, Model Driven Architecture (MDA). (June):1–15, 2003.

[29] S. Runde and A. Fay. A data exchange format for the engineering of building automation systems. *2008 IEEE International Conference on Emerging Technologies and Factory Automation*, pages 303–310, Sept. 2008.

[30] J. Schein. An information model for building automation systems. *Automation in Construction*, 16(2):125–139, Mar. 2007.

[31] A. Schlueter and F. Thesseling. Building information model based energy/exergy performance assessment in early design stages. *Automation in Construction*, 18(2):153–163, Mar. 2009.

[32] D. C. Schmidt. Model-Driven Engineering. *Computer*, 39(February):25–31, 2006.

[33] W. Shen, Q. Hao, H. Mak, J. Neelamkavil, H. Xie, J. Dickinson, R. Thomas, A. Pardasani, and H. Xue. Systems integration and collaboration in architecture, engineering, construction, and facilities management: A review. *Advanced Engineering Informatics*, 24(2):196–207, Apr. 2010.

[34] R. Vanlande, C. Nicolle, and C. Cruz. IFC and building lifecycle management. *Automation in Construction*, 18(1):70–78, Dec. 2008.

[35] W. Yan, M. Clayton, J. Haberl, W. Jeong, J. B. Kim, A. Texas, and C. Station. Interfacing BIM with Building Thermal and Daylighting Modelling. (Bazjanac 2001), 2007.