Automatic development of Building Automation Control Network (BACN) using IFC4-based BIM models.

R. Sanz, S. Álvarez & C. Valmaseda
Fundación CARTIF, Energy Division, Boecillo (Valladolid), Spain

D. V. Rovas
Institute for Environmental Design and Engineering, University College London, London, United Kingdom

ABSTRACT: Nowadays Building Information Modelling (BIM) tools are widely used in the process of design and construction works for new building and renewals. Architects and engineers usually design the architectural characteristics without paying attention to other facilities such as Building Automation Control Networks (BACN) which are normally defined, developed and deployed after the building is finished by a third party company. Then the model of the building does not match with the final status as it is built. Taking advantage of the fact that IFC standard tries to cover the problematic of interchanging information among building model design software (interoperability), the standard will be used to automatically design and deployment of BACN (facing problematics related to geometry and location of devices) and its possibility to act on control facilities.

1 BUILDING AUTOMATION CONTROL NETWORK AND BIM MODELLING

Building Information Modelling often encapsulate geometric and building services information, but rarely is information related to the Building Automation and Control Network (BACN) captured within these models. This is partly because design and commissioning of the BACN happen at the later design stages and there is little-perceived value in adding this information to the model. A second aspect is that often it is not possible to capture all relevant BACN parameters and configurations within existing open data models, e.g. IFC4 (BuildingSMART 2018) – there are other data models and ontologies, e.g. SAREF (ETSI 2015) or BRICK (BRICK Schema 2016) as well as the data models linked to specific automation protocols, that better capture all pertinent parameters and configurations of the building automation domain. Still, in line with lifecycle interoperability considerations, there is value and partial model support, in keeping the IFC-based building model updated with BACN information.

In this paper a methodology is presented whereby a BACN network can be configured semi-automatically and, once this process is complete, the IFC model of the building is updated to include information about the control network. Thus, this information becomes available and queryable for facility management or other operational purposes. Core to our approach is a product library of commercially available devices that includes sensor, controllers and actuators to be integrated into an automation network. We chose to work with devices that support the LonWorks protocol – but many of the ideas would be applicable, with necessary adaptations, to other standard building communication protocols.

The devices database stores various types of sensors, controllers (focusing in this paper on controllers that conform to Space Comfort Controller Profile), actuators, from different vendors and associated functional features, as well as, purchasing costs. In LonWorks functionalities are derived from the different Functional Blocks which a device is composed of and associated Network Variables. Once device selection is complete, these Blocks need to be connected between them for information exchange to be possible; this is translated to the ways the Network Variables of each Functional Block in devices are linked to others in other Functional Blocks. Once the automation network is completely developed, it is necessary to update the BIM model with the geometric and non-geometric characteristics of the devices included in the network. In the first case, of geometrical characteristics, devices have been modelled as rectangular prisms using a simplified geometric representation; in this case, characteristics related to the bounding box, location, dimensions and orientation are needed. A more detailed elaboration of the automation components' geometry is also possible, but this has no impact to our study.

Replicating expert knowledge, the geometrical placement of each device in the model is done following a set of rules: sensor, controller and actuator can be placed in 1) a wall in the same space as the HVAC system is placed in the model, or 2) in the ceiling in which the HVAC system is placed, or 3) sensors are placed in the same space as the HVAC system and the remaining devices in alternative locations such as a mechanical room or an electric cabinet. In all cases the BIM model is interrogated to identify the walls or ceilings where the devices can be potentially placed. Other non-geometrical characteristics regarding vendor, device type (temperature sensor, controller, actuator) unique identification number, electrical needs, price, and so forth, are also included in the model of the automation network when the devices are selected. As a proof of concept, a tool implementing the proposed methodology has been developed. The user through a simple interface can setup or change configurations of the control network, see historical data, and find the interconnected devices. Tests in building case studies demonstrate the relevance of both strands of the proposed work: (i) in the automated creation of the Building Automation Control Network – which significantly improves the commissioning process of the BACN network; and, (ii) in the automated updating of the BIM model, addressing the issue of not having an up-to-date model updated after the control network has been developed. By way of example, a case study is presented in this paper, whereby the automation network has been developed for temperature regulation of each space in a building, assuming an that building is serviced by individual split-type air conditioning systems. This can be expanded to other types of control, to include lighting and indoor air-quality aspects.

2 IFC4 AND LONWORKS.

2.1 Building Controls Domain in IFC4

The Industry Foundation Classes data schema have been in development for more than 20 years and this ambitious project under the oversight of BuildingSmart has expended and improved over the various iterations of the schema. Now an ISO standard (ISO 16739) IFC is widely accepted as the de facto openBIM standard. In view of the above, the use of IFC is particularly appealing due to its open nature. Moreover, as stated in the standard the IFC development aspires to capture aspects along the building life cycle to include beyond design, operation and maintenance. Building services and facility management are explicitly within the scope of the definition, and particularly relevant for our purposes.

During this evolution of IFC, the schema has expanded to include definitions of new elements (entities), systems (domains). The latest version of IFC, IFC4 Add2 includes timely definitions related to the representation of building control systems. In the latest version new entities have been added or expanded in the Buildings Control Domain (IfcBuildingControlDomain schema with forms part of the Domain Layer of the IFC model) to capture information on pertinent elements (sensors, controllers and actuators) using the IfcSensor, IfcController and IfcActuator entities. The concepts of building automation, control, instrumentation and alarm are defined to support ideas including types and occurrences of: actuator, alarm, controller, sensor, flow instrument, unitary control element. Components that physically perform the control action such as valves and dampers are subtypes of distribution flow elements located within the IfcHvacDomain and IfcElectricalDomain schemas.

2.2 LonWorks control networks

LonWorks networks are based on the use of the LonTalk protocol which is an open standard developed for LonMark mainly used for Building Automation. LonWorks is based on a hierarchical level architecture, see Figure 1. (Echelon Corporation 1997a, 1997b, 1999)

The Object Server object contains a collection of Network objects, each of which represents a defined network. Each Network object contains a System object representing the network system, and each System object contains a set of Subsystem objects that represent logical or physical partitions of that particular network. Each Subsystem object contains a collection of AppDevice objects which represent the application devices defined in that subsystem. Thus, LonTalk protocol provides a common applications framework that ensures interoperability using powerful concepts called network variables and Standard Network Variable Types (SNVTs). Functional device models have been developed by the Lonmark Interoperability Association to assure plug and play compatibility.

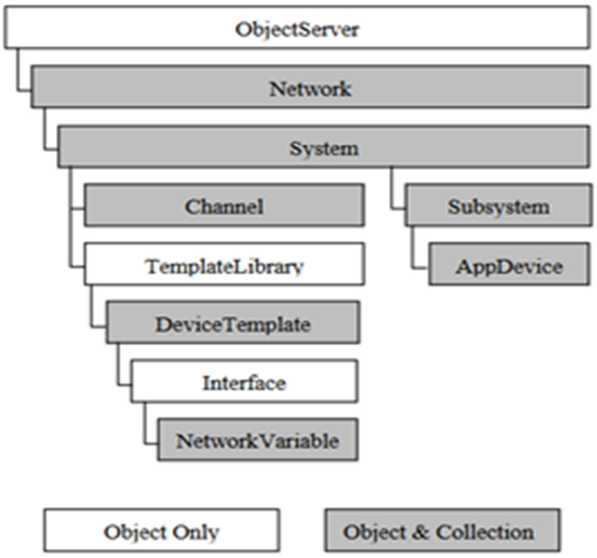


Figure 1. Hierarchy of the LonWorks.

Communication between nodes on a network takes place using the network variables that are defined in each node. The product developer defines the network variables when the application program is created as part of the Application layer of the protocol. Network variables are shared by multiple nodes. Some nodes may send a network variable while others may receive. By only allowing links between inputs and outputs of the same type, network variables enforce an object-oriented approach to product development. This greatly simplifies the process of developing and managing distributed systems. In addition, Network Variables are part of Functional Blocks which are used as the device interface including also configuration properties that are used to perform a determined task.

3 MAPPING BETWEEN IFC AND LONWORKS DATA MODEL

3.1 IFC and LonWorks matching

Connection among control devices in IFC4 is based on the idea that the output of a Sensor is connected to an input of a Controller, which can have outputs connected to other controller or connected to an actuator as it is shown in Figure 2. IFC4 also defines several types of sensors, depending the physical characteristics of the measured signal, controllers, depending on whether they are programmable, proportional, etc and actuator such as electric, hydraulic, pneumatic, etc.

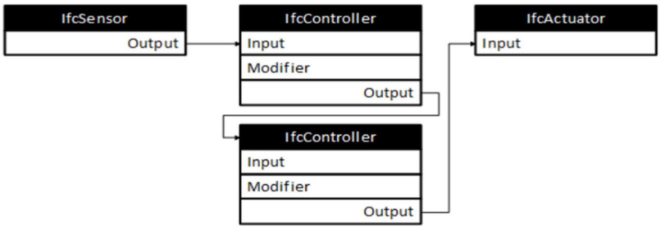


Figure 2. Control linking paradigm in ifc.

Although there are several physical connection topologies for control networks using the LonWorks protocol (ring, bus, star, mixed) the fundamental idea of connecting between devices (for logical connection and interchange of information in the network) follows the sensor-controller-actuator paradigm even though the link does not necessarily have to be one-to-one.

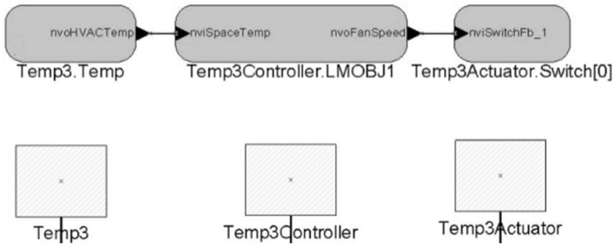


Figure 3. Basic LonWorks control connection.

As shown, both data models, IFC and LonWorks are pretty similar, allowing the user to make the assumption of using LonWorks Network Variables as Ifc Outputs and Ifc Inputs and LonWorks Devices through Functional Blocks as IFC Devices.

4 BUILDING AUTOMATION CONTROL NETWORK GENERATION AND IFC UPDATING

4.1 Automated generation of the Building Automation Control Network.

As explained in the introduction, a software tool/application has been developed to use on this work, also other previous developed tools such as an exporter from LonWorks to SQL database, and several SQL databases which support the storage of historical (dynamic database) data and architecture of LonWorks Control Network (static database). The static database is an exact copy of the structure of the LonWorks network including all devices and connections between them. This database is deployed to have a possibility to access this configuration from any other software module, and it is performed once the Building Automation Control Network is totally functional and deployed. A software module reads information from a xml file (exported using LONMAKER software) and automatically creates the overall structure of the database, relationships between tables, etc. On these tables are also stored all the connection between Functional Blocks and Network Variables of the control network Devices.

In case of automatic development of the Building Automation Control Network the application follows the sequence that is shown in the following figure.

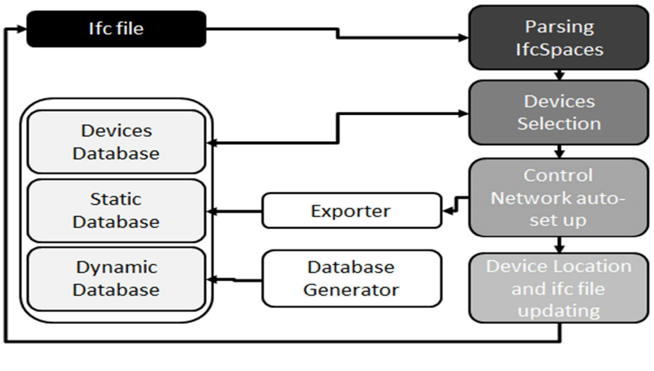


Figure 4. Flowchart on generation of BACN.

Starting from a building model (IFC4 file), the tool uses information contained in the model to identify spaces inside the building. Once the division into spaces is done, next step is Device Selection, this is carried out using a Device Database (a product catalogue) in which many different devices from several vendors, applications, and types are stored. This catalogue is also used to store price data for each device, as well as, the number and type of functional blocks. The selection is then performed using different criteria to cover the needs of the heating system (in this approach based on individual split-type air conditioning system for each space). At this point the control network set-up is automatically done, linking Functional Blocks of devices through Network Variables (SNVT's).). An example of this linking can be seen in Figure 3. In this Figure an example is shown of the Temperature sensor (Temp3), temperature controller (Temp3Controller) and actuator for the split system (Temp3Actuator) which in this case is an ON/OFF switch. Once selection of devices and the links between them have been established the LonWorks Building Automation Control Network definition can be exported to a formatted xml file. This can in turn be imported to the commissioning and integration tool LonMaker for deployment of the configurations to the physical devices. In the first stage, the information about unique identification number (NeuronID in Lonworks) is fulfilled with a "dummy" number due to this number is not known until the devices are purchased, then the Lonworks control network needs to be updated (by hand) with this information to be entirely operative.

4.1.1 Device location and ifc file updating.

In IFC4 the placement of objects is achieved using IfcObjectPlacement which is an abstract super type of placement subtypes that define the location of an item, or an entire shape representation, and provide its orientation. Any IfcObject has an IfcLocalPlacement that can be related to other / others IfcLocalPlacement and a RelativePlacement which configures the local coordinates system using IfcAxisToPlacement3D. IfcAxis2Placement3D provides location and orientations to place items in a three-dimensional space. The attribute Axis defines the Z direction, RefDirection the X direction. The Y direction is derived.

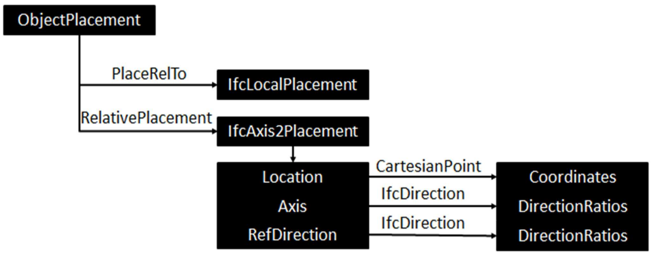


Figure 5. IFC4 placement for an Object.

Using a coordinate transformation matrix, each position of any of the IfcObject is calculated using "Bounding Box" properties. As result coordinates for (X_{min}, Y_{min}, Z_{min}) and (X_{max}, Y_{max}, Z_{max}) of each component (IfcSpatialStructure, IfcWall, IfcObject, etc.) can be obtained. Depending on the results of the mapping between the local and global coordinate systems four different cases are possible regarding numerical value of X_{min} and Y_{min}. It is supposed to have the origin reference the same as the global origin reference.

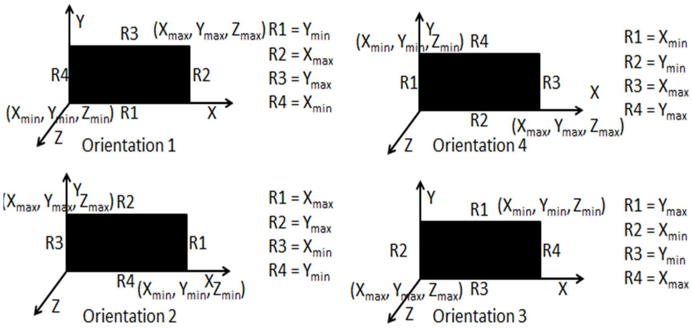


Figure 6. Orientation depending on coordinates.

To select the correct placement for the devices, the relative location and orientation between spaces and walls needs to be checked due to several possibilities existing depending upon the orientation of space and walls. Knowing if a wall is part of a space the easiest method is comparing if any of the lines R1, R2, R3 and R4 are common between wall and space. Implicit to this assumption is that we have a good quality model, so model checking, and proper guidelines are an essential pre-requisite.

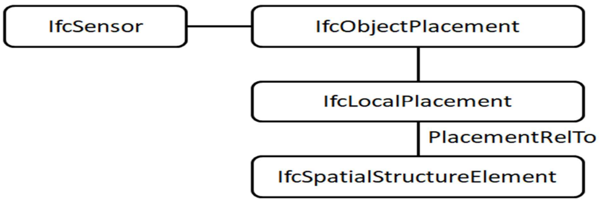


Figure 7. Relation between ifc control device and ifc Spatial Structures.

The schematic above shows how in IFC4 spatial containment is generated to include the devices of the control network in spatial contexts. According to IFC, structural elements such as walls may not be directly related to spaces. This relationship can be inferred using geometric properties and coordinates in space. To identify whether a wall surrounds or is part of a space, in addition to using the criterion explained above in which some of the lines part of their "Bounding Box" are common it is necessary to know if the wall is part of the space and not part of the prolongation of it. Wall and space are projected on the vertical and horizontal axes and then the tool decides if the wall is surrounding the space by comparing the coordinates of both.

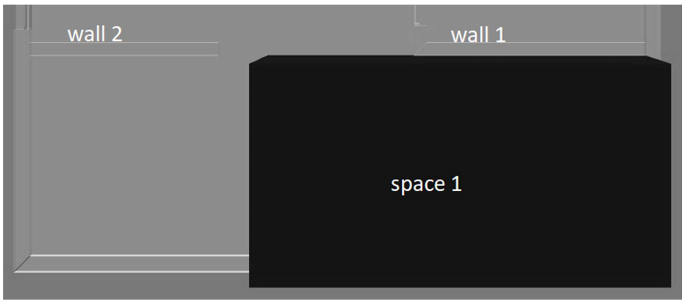


Figure 8. Example of Spaces surrounded by Walls.

For the example of the previous figure when projecting in X axes; Space1 X_{min} < Wall 1 X_{min} and Space 1 X_{max} >= Wall 1 X_{max} and Wall 1 X_{min}, then Wall 1 surrounds space 1. Wall 2 X_{min} and Wall 2 X_{max} < (Space 1 X_{min} and Space 1 X_{max}). In this case wall 2 does not surround Space 1.In this case wall 2 does not surround Space 1. As result of the projection and using the common lines criteria, the application makes a list including all possible walls where the devices of the automation control network can be placed. It is necessary to note here that this work is developed for square-based layout buildings on which spaces are defined as rectangular areas, in case the layout is other, the tool needs to be updated to include new capabilities. Now is necessary to know if the wall has openings (doors or windows) and, in this case, delete them from the list of possible locations. When the selection of the location is finished (right now the tool selects the first wall of the non-opening containing walls) the model of the building is updated including the information of the selected devices such as name, type of sensor, controller, actuator, graphical characteristics, etc. (in this work, both, sensors, controllers, actuators have been modelled as rectangular polygons, due to the graphical characteristics not being considered as critical at this stage).

Tests have been done using the next building model (IFC4 file). This model has several spaces, considered as individual zones, shown in Figure 10 below.

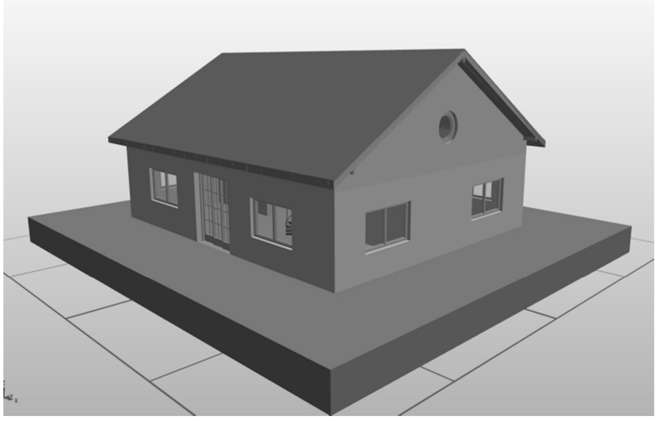


Figure 9. Building model used for tests.

Following application of the tool developed and described in this paper, an enhanced IFC4 file is generated whereby the control network characteristics are included in the model and devices are located in a proper position (white squares are surrounding devices).

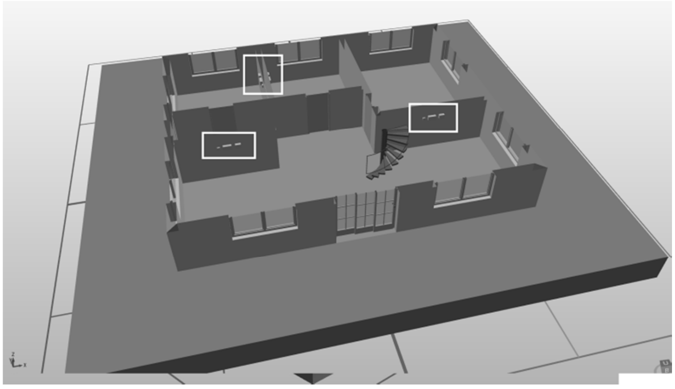


Figure 10. IFC4 placement for sensor-controller-actuator.

It should be admitted that the spatial placement of these devices might have little correspondence with their placement in reality. The procedure just described encapsulates expert knowledge regarding the placement of the sensors at various locations. If more precision on the placement in the model is required, then using and IFC editor (capable of handling placement of IfcSensor objects) can be used. But the model thus generated has value in relation to queries like number and types of sensors in a specific space, sensors connected to a specific building service (e.g. split-type air conditioning), controller linked to a conditioning of a space, and so on. All this information is particularly relevant in the context of hard facility management and therefore there is great value to the information even if the placement is approximate. Obviously, as mentioned before, manual and automated ways to establish a correct sensor placement can be established, but this is beyond the scope of this work.

5 IFC4 FOR BUILDING CONTROL.

When the Building Automation Control Network has been defined, included in the IFC4 file and physically deployed we can use the application to monitor and control the building facilities - the flowchart in Figure 11 illustrates how this is accomplished.

First the spaces contained in the building are obtained from the model and, for each space, sensors contained are identified (in this case temperature). The remaining devices of the automation and control network and their connectivity (in terms of communication flow) are queried from the “Static Database” which maintains an accurate representation of the control network, information about devices, network variables, Functional Block and how all these are inter-connected (called “Targets” in LonWorks terms).

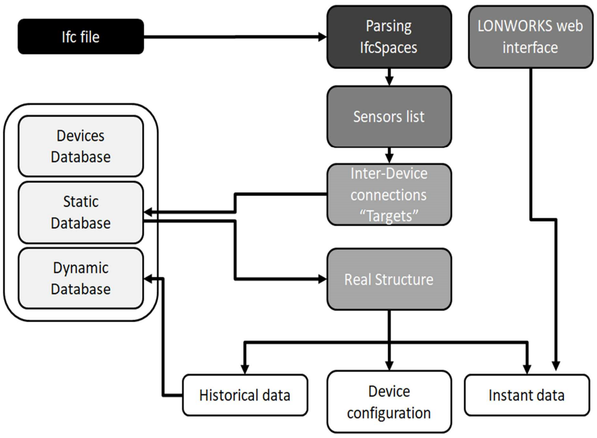
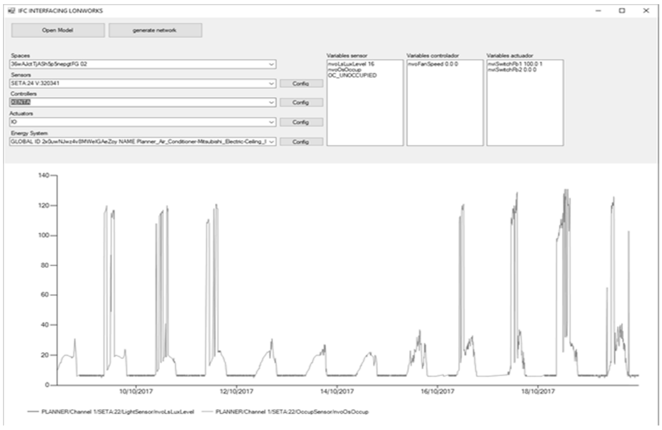


Figure 11. Flowchart for direct control from Ifc4 files.

When this information has been retrieved the user can access historical data for any of the variables in the network (values are stored in the dynamic database which stores time-series data), change reference values, switch on/off devices, etc. To access these data and store them, the application uses a web – services based interface, which is deployed in a device of the control network actuating as server and network interface, then the client retrieves data and store them in the dynamic database which is composed by three tables which include information on name of signal, value, and date time log between others. Thus, the user can control the facilities of the building directly using a simple interface (application) and the BIM model of the building through its IFC4 file.



Spaces: 36wAJctTjASh5p5nepgtFG 02

Sensors: SETA:24 V:320341

Controllers: XENTA

Actuators: IO

Energy System: GLOBAL ID 2x0uwNJwz4v8MWeIGAeZoy NAME Planner_Air_Conditioner-Mitsubishi_Electric-Ceiling_I

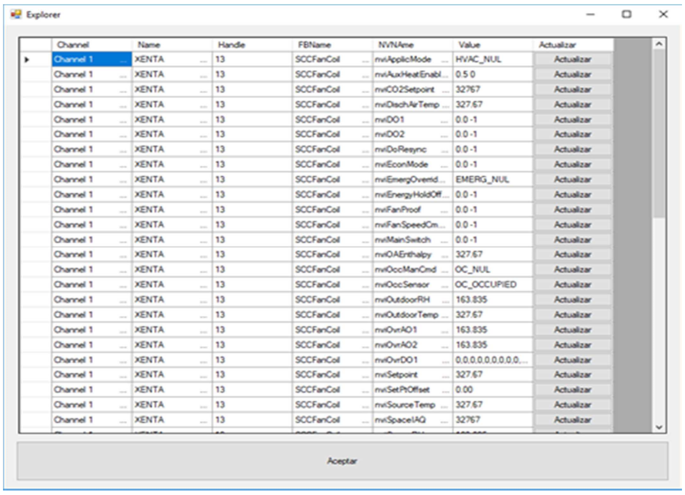
Variables sensor: nvoLsLuxLevel 16
nvoOsOccup
OC_UNOCCUPIED

Variables controlador: nvoFanSpeed 0.0 0

Variables actuador: nviSwitchFb1 100.0 1
nviSwitchFb2 0.0 0

Figure 12. Historical data of a variable.

By selecting one of the devices (sensor, controller, actuator) we can access to other window which allows the user to change any reference on the network variables.



Channel	Name	Handle	FBName	NVNAme	Value	Actualizar
Channel 1	XENTA	13	SCCFanCoil	nviApplicMode	HVAC_NUL	Actualizar
Channel 1	XENTA	13	SCCFanCoil	nviAuxHeatEnabl...	0.5 0	Actualizar
Channel 1	XENTA	13	SCCFanCoil	nviCO2Setpoint	32767	Actualizar
Channel 1	XENTA	13	SCCFanCoil	nviDischAirTemp	327.67	Actualizar
Channel 1	XENTA	13	SCCFanCoil	nviDO1	0.0 -1	Actualizar
Channel 1	XENTA	13	SCCFanCoil	nviDO2	0.0 -1	Actualizar
Channel 1	XENTA	13	SCCFanCoil	nviDoResync	0.0 -1	Actualizar
Channel 1	XENTA	13	SCCFanCoil	nviEconMode	0.0 -1	Actualizar
Channel 1	XENTA	13	SCCFanCoil	nviEmergOverrid...	EMERG_NUL	Actualizar
Channel 1	XENTA	13	SCCFanCoil	nviEnergyHoldOff...	0.0 -1	Actualizar
Channel 1	XENTA	13	SCCFanCoil	nviFanProof	0.0 -1	Actualizar
Channel 1	XENTA	13	SCCFanCoil	nviFanSpeedCm...	0.0 -1	Actualizar
Channel 1	XENTA	13	SCCFanCoil	nviMainSwitch	0.0 -1	Actualizar
Channel 1	XENTA	13	SCCFanCoil	nviOAEnthalpy	327.67	Actualizar
Channel 1	XENTA	13	SCCFanCoil	nviOccManCmd	OC_NUL	Actualizar
Channel 1	XENTA	13	SCCFanCoil	nviOccSensor	OC_OCCUPIED	Actualizar
Channel 1	XENTA	13	SCCFanCoil	nviOutdoorRH	163.835	Actualizar
Channel 1	XENTA	13	SCCFanCoil	nviOutdoorTemp	327.67	Actualizar
Channel 1	XENTA	13	SCCFanCoil	nviOvrAO1	163.835	Actualizar
Channel 1	XENTA	13	SCCFanCoil	nviOvrAO2	163.835	Actualizar
Channel 1	XENTA	13	SCCFanCoil	nviOvrDO1	0,0,0,0,0,0,0,0,0,...	Actualizar
Channel 1	XENTA	13	SCCFanCoil	nviSetpoint	327.67	Actualizar
Channel 1	XENTA	13	SCCFanCoil	nviSetPtOffset	0.00	Actualizar
Channel 1	XENTA	13	SCCFanCoil	nviSourceTemp	327.67	Actualizar
Channel 1	XENTA	13	SCCFanCoil	nviSpaceIAQ	32767	Actualizar

Figure 13. Variables changing window.

These read and write values of variables on the Network and their instant values are accessed using the capability of the network of communicating with other systems using Web Services (Client/Server based).

6 FUTURE WORK

Once the proof of concept of the methodology, the first step in the development of the tool is clearly the addition of some capabilities that increase the utility such as automatic parsing of devices (modeled in IFC4), extracting geometrical and structural characteristics, in order to have a library of components more extended including the exact graphical representation, avoiding rectangular prism and having as result an exact model of the building and facilities.

Integrate the information of the Building Automation Control Network in a 3D viewer with adding to it the capabilities of control, historical data representation, alarming, etc. making more attractive to customers and systems integrators to implement this solution in their products as an interface to the BEMS. This needs to be completed by extending the work to other Building Automation Control Network protocols (BACNET, etc) to increase the interoperability and standardization of the tool.

Expand the use cases to other more complex automatic systems for heating and cooling and other facilities as well as make the tool smarter in the process of device selection and design of the network, adding other criteria i.e. length of cabling, number and type of Functional Blocks (in case there are devices with several Functional Blocks for different operations), etc.

Improve the placement of devices procedure, adding possibilities such as technical rooms, electrical cabinets, etc. and different morphologies for building spaces (Manhattan-World-type structures).

7 CONCLUSIONS

The integration of Building Automation Control Networks in the BIM through IFC4 is possible in an automated way firstly by developing the network in a BACN-specific data model and secondly by including devices and their geometric location in the IFC4 file.

The possibility to control from the BIM model once IFC4 format includes elements relating to the network is quite clear.

Relations between devices can be found automatically, thus knowing from a user-friendly interface, regardless of where and how the control network has been developed.

Control of the facilities without knowing the network topology of the automation network, which offers the possibility to error detection of devices and alarming.

As long as all the information of the Building Automation Control Network can be accessed, devices can be set. It is also possible the representation of historical and trends, analysis of the behaviour of the building, etc.

Benefit from both the architectural part and the part of engineering for the control, since that is not two separate developments in which both have that design and integrate the network in their corresponding platform, but it saves much time on these tasks since that occurs automatically and completely transparent.

Building model (IFC4 based) is updated after selecting devices including the Building Automation Control Network information.

This work is based on the use of LonWorks control networks but it is expandable to other protocols and architectures such as BACNET, MODBUS, etc.

8 ACKNOWLEDGEMENTS

Part of the work presented in this paper is based on a research developed in an internship at the Institute of Environmental Design and Engineering, University College London (UCL), London. This internship has received funding from the Agencia de Innovación, Financiación e Internacionalización Empresarial de Castilla y León.

9 REFERENCES

BRICK Schema. 2016. Brick: A Uniform Metadata Schema for Buildings. Available online: https://brickschema.org/ (accessed: May 2018)

BuildingSMART. 2018. IFC4 Release. Available online: http://www.buildingsmart-tech.org/specifications/ifc-releases/ifc4-release (accessed: January 2018).

Echelon Corporation, Copyright ©1997-2011. Lonworks Network XML Programmer's Guide.

Echelon Corporation, Copyright ©1997-2009. i.LON ® SmartServer 2.0 Programming Tool User's Guide.

Echelon Corporation, Copyright © 1999. Introduction to the Lonworks System.

ETSI. 2015. SAREF: the Smart Appliances REFerence ontology. Available online: http://ontology.tno.nl/saref/ (accessed: May 2018).

Karavan A., Neugebauer M, Kabitzsch K. Mergin Building Automation Network Design and IFC2x Construction Projects. ITC Digital Library 2003.

Karavan A. Neugebauer M, Kabitzsch K, Integration of Building Automation Network Design and 3D Construction Tools by IFC Standard. Proc. FeT2005 6th IFAC International Conference on Fieldbus Systems and their applications, Puebla, Mexico, November 2005, pp247-254.