

# Merging Building Automation Network Design and IFC 2x Construction Projects

A. Karavan

*Institute for Applied Computer Science, Dresden University of Technology, Dresden, Germany*

M. Neugebauer

*Institute for Applied Computer Science, Dresden University of Technology, Dresden & SAP Research Group Dresden, Dresden, Germany*

K. Kabitzsch

*Institute for Applied Computer Science, Dresden University of Technology, Dresden, Germany*

**ABSTRACT:** Currently, different design tools and databases are used for building automation networks and construction projects. Effort (design time, tools) can be reduced if these two design spaces can be merged. In this paper an approach for the integration of building automation network design in IFC 2x is introduced. Based on the resulting model, existing tools are to enhance in order to accomplish a holistic design approach for construction projects.

## 1 INTRODUCTION

Today, building automation networks are designed using standardized tools and data bases. The model structure of a building automation network with all required components (i.e. sensors, controllers, actuators) is created with graphically supported tools (i.e. LonMaker, see Figure 1). All necessary connections between the components on different layers (application and physical layer) are carried out within these tools. Even deployment (configuration of real components according to model structure) and maintenance (network monitoring, deployment of additional functionalities and devices, etc.) are facilitated by these tools.

There are certain similarities between the approaches used for designing building automation networks and construction projects. The LNS Network Operating System (Echelon 1999) is the quasi standard data base for design and deployment of building automation networks used by different parties involved. Therein, the complete information about the structure of the network on different layers is contained. The aim of IAI's IFC 2x (IAI 2005) is about the same in a different application area: a common standard to exchange information among various parties involved in a construction projects (i.e. spatial structure, walls, building services elements).

So far, the two design approaches for building automation networks and construction projects are carried out in two different tool worlds. According to the data on which the structures rely on there is no connection, except a drawing delivered from the architect to the building automation network designer. There are no basic approaches for interoperability guidelines that could enable seamless integration of building automation network structures in construction projects.

In this article we show an approach for future design methodologies and tools to integrate building automation network design and therewith to enable a comprehensive building model including the automation facilities (Schach et al. 2002). Especially, the similarities in current tool domains (building automation networks and construction projects) are elaborated in order to enable merging of different design approaches. Hence, the existing model structures in building automation networks have to be re-used and integrated in IAIs IFC 2x.

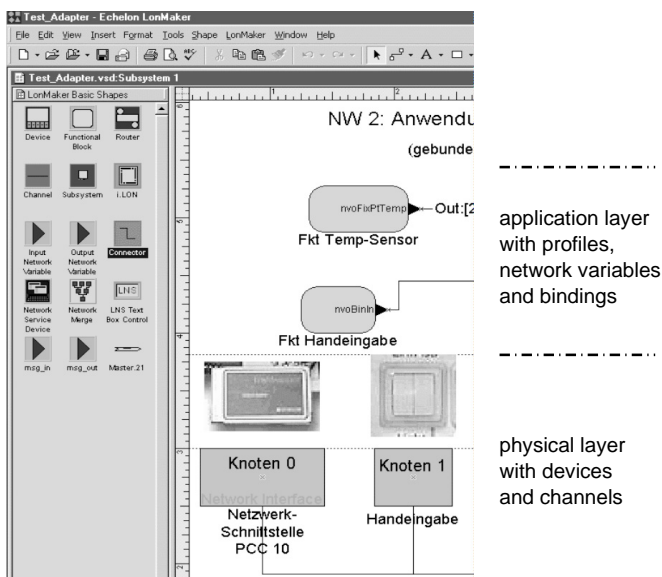


Figure 1. LonMaker for graphically supported design of building automation networks.



In Section 2 and 3 LonWorks technology and IAIs IFC 2x are introduced. Concepts for merging the two design methodologies are discussed in Section 4 and explained in more detail with an example in Section 5.

## 2 LONWORKS AS A BUILDING AUTOMATION NETWORK

Building automation is a fast growing market. It proposes now open and interoperable solutions (Dietrich et al. 2001). One of the leading networking standards in this shift is the LonMark standard, based on LonWorks control network technology.

A LonWorks network consists of intelligent devices that communicate with each other using a common protocol over one or more communication channels (Fig. 2). Each device includes one or more processors that provide its intelligence and implement the protocol. Each device also includes a component called a transceiver to provide its electrical interface to the communication channel. Transceivers are available for a variety of communication media including single twisted-pair cable, power line, radio frequency (RF), infrared, fiber optics, and coax cable (Echelon 1999).

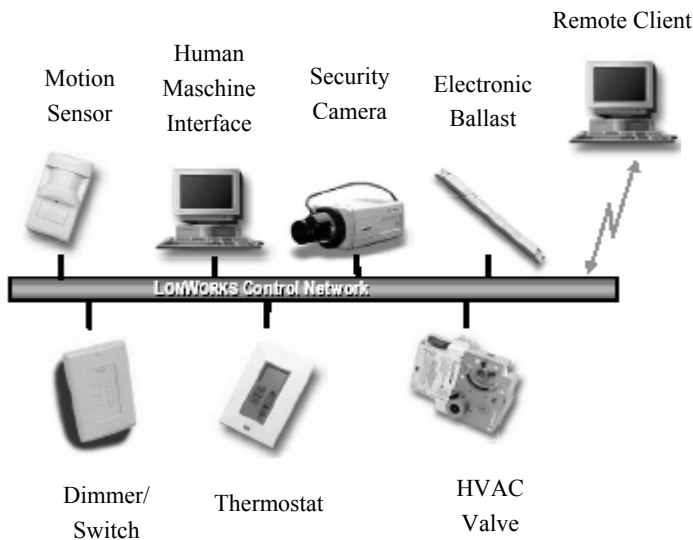


Figure 2. LonWorks network.

The LonWorks protocol, also known as the LonTalk protocol and the *ANSI/EIA 709.1 Control Networking Standard*, is the heart of the LonWorks system. The LonWorks protocol is a layered, packet-based, peer-to-peer communication protocol. The seven layers along with the corresponding services are provided by the LonWorks protocol.

The LonWorks protocol implements the concept of network variables. Network variables simplify the tasks of designing LonWorks application programs for interoperability with multivendor products and facilitate the design of information-based, rather than command-based, control systems.

Via a process that takes place during network design and installation called binding, the device firmware is configured to know logical addresses of the other devices or groups of devices in the network. The binding process creates logical connections between an output network variable in one device and an input network variable in another device or group of devices. Connections may be thought of as “virtual wires” (Fig. 10).

Every network variable has a type that defines the units, scaling and structure of the data contained within the network variable. A set of standard network variable types (SNVTs) is defined for commonly used types.

LonWorks Network Services (LNS) architecture is important for the LonWorks protocol. LNS provides an object-oriented programming model and brings the power of client-server architecture and component-based software design into control networks (Fig. 3). LNS enables to install, maintain, monitor and control LonWorks networks (Echelon 2004).

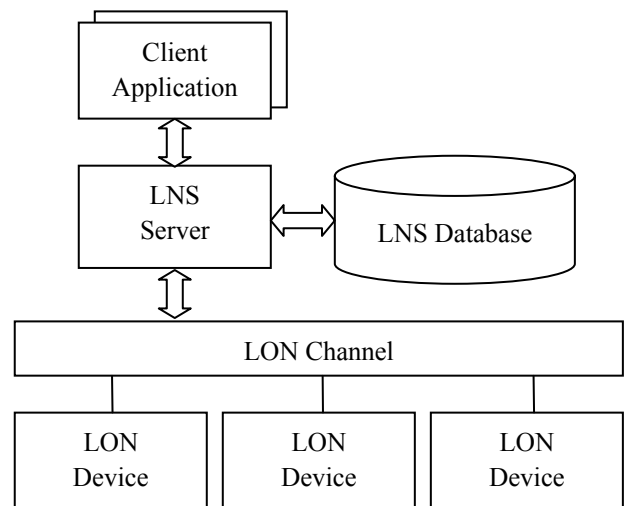


Figure 3. LNS architecture.

The LNS Object Server defines a set of objects, properties, methods and events that represent the physical attributes of networks and their configurations. LNS Object Server is a COM server that provides an interface, independent of any programming language, to the LNS Server and the LNS database.

The objects are grouped together in a hierarchical fashion (Fig. 4), such that the *ObjectServer* object is at the top of the hierarchy. The *ObjectServer* object contains a collection of *Network* objects, each of which represents a network defined in the database. Each *Network* object contains a *System* object representing the network’s 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.



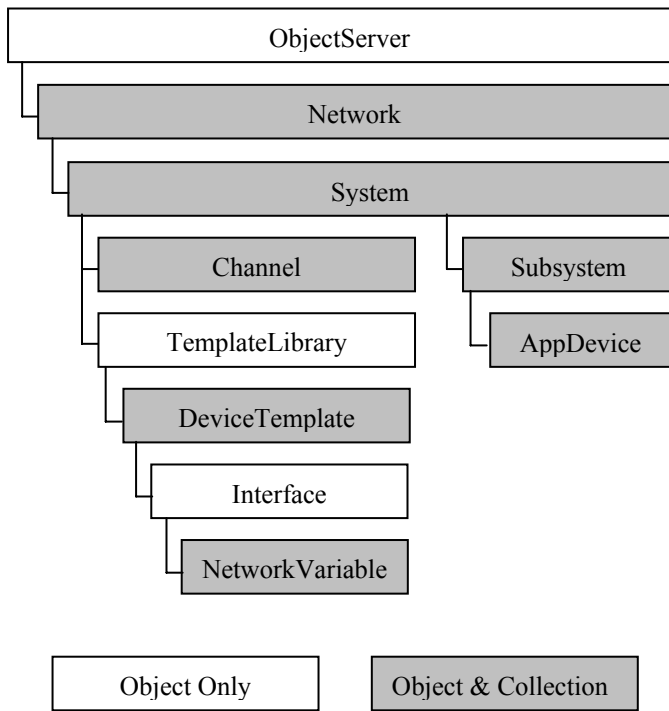


Figure 4. LNS Object Server hierarchy.

For example, the user must set up a network to control a building with three floors. The user can define three subsystems within the network, so that the user can logically group the devices on each floor separately. The user can also set up additional subsystems representing different rooms on each floor of the building.

As the subsystems are logical divisions of a network, and devices can belong to multiple logical divisions, the user can create multiple subsystems that cross-reference a network, and add individual devices to each of them. In this manner, devices can appear in multiple subsystems. For example, the user can create one subsystem to represent the physical layout of network, and another one to represent the functional layout (i.e. HVAC, lighting, etc.).

Table 1. Properties and methods of LNS objects.

AppDevice	Subsystem
- AttachmentStatus	- AppDevices
- Channel	- ClassId
- Handle	- Description
- Name	- Extension
- NeuronId	- Handle
- SubnetId	- Name
- Subnets	- RouterDevices
- Subsystems	- Subsystem
+ Load	
+ Reboot	
+ Replace	
+ Reset	
+ Test	

Each one of the object types defined in the LNS Object Server includes its own set of properties and methods (Table 1). Properties contain information defining the current configuration and operational

behaviour of an object. Methods provide a mechanism to perform various operations on each object.

### 3 IFC AS A SHARED PROJECT MODEL

IFC (Industry Foundation Classes) is the set of internationally standardized object definitions (ISO 2005) for use in the building industry developed by the International Alliance for Interoperability (IAI 2005).

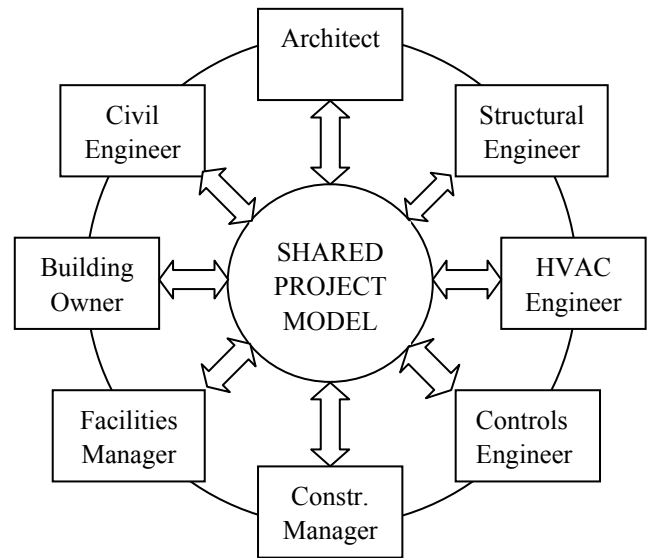


Figure 5. Shared project model for building.

IFC is an universal language to improve the communication throughout the design, construction, operation and maintenance life cycle. The IFC concept is based on the idea of objects brought together in an integrated model. These objects are defined to support the whole lifecycle of facility development from inception through design, then facility management and finally demolition (Fig. 5).

Objects in IFC in addition to the full geometric description in 3D have also location and relationships, as well as all the properties of each object.

IFC contains not only physical properties, but also spatial concepts such as floors and rooms (Liebich 2004).

The following three different concepts are subsumed under the *IfcSpatialStructureElement* (Fig. 6):

- Building
- Building Storey
- Space

These different entities contain each other such as they provide a clear hierarchical structure for the building project. The three subtypes *IfcBuilding*, *IfcBuildingStorey*, *IfcSpace* are used to represent the levels of the spatial structure. The spatial structure is created by using the fundamental decomposition relationship. The subtype *IfcRelAggregates* is used



to link the instances and establishes a hierarchical structure.

*IfcBuilding* is used to provide additional information about a building itself. The building includes the references to the building storeys belonging to that building or it can have building sections.

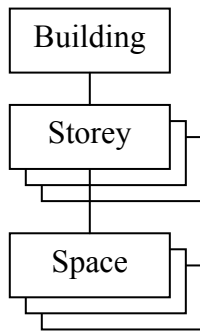


Figure 6. Spatial structure of a building.

*IfcBuildingStorey* is used to provide information about a building storey itself, a vertical structure normally used with building and construction. The building storey includes the references to the spaces belonging to it.

*IfcSpace* is used to provide information about the space as a functional area or volume with a spatial structure. Spaces normally contain all building services or interior design elements (such as distribution elements, electrical elements, furniture). The space boundaries define the relationship between the space and its bounding elements.

#### 4 INTEGRATION PLATFORM CONCEPT

As it has been mentioned in Section 4, the LNS model represents the structure of the LonWorks automation network, and IFC model - the structure of the building. These models are created by means of different tools saved in various databases and have different interfaces to data.

By creation of a new project of the building automation network, the user repeats operations which have already been done by the user of the CAD system and results are kept in IFC model (definition of subsystems: floors, rooms and systems of lighting, ventilation, heating). For the process acceleration of the building automation network design, it is expedient to integrate LNS model in IFC model, as more general and universal one. The holistic model also can be used for more detailed modelling of various processes in the building.

For the planning architecture of the building and its life-support systems various CAD systems are used. These systems support import and export of the IFC data model. These systems have also their own API with which it is possible to expand functionalities at the program level. Some CAD systems, such as ArchiCAD and ADT (Geiger 2001), support

import and export of IfcXML format that is necessary for integration.

For the integration of two models we propose the following approach:

1. integration on the basis of an existing automation network model;
2. integration on the basis of the creation of a new automation network model.

#### 4.1 Integration on the basis of an existing automation network model

This kind of the integration implies that the model is created by other tools (e.g. LonMaker) and stored in the LNS database. As mentioned above, CAD systems (e.g. ArchiCAD) support the import of IFC data. There are two basic methods for such data exchange: the IFC (STEP) file and the IfcXML file. For the construction of the IFC model on the basis of the LNS database, it is possible to use XML (Fig. 7).

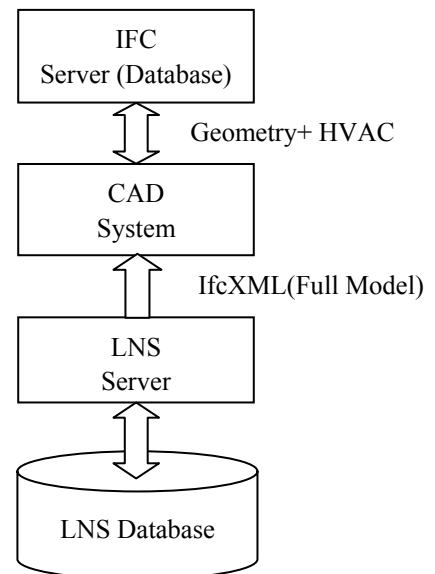


Figure 7. Integration on the basis of existing model.

Integration process proceeds as follows. Necessary data are read out from the LNS database, an ifcXML file is created dynamically and then it is imported in ArchiCAD. By means of a special tool, a user of ArchiCAD links physical objects (sensors, controller) with objects of the automation network model.

#### 4.2 Integration on the basis of the creation of a new automation network model

That kind of integration implies that the model is created in a CAD system. On the basis of an already existing building model (geometrical model, HVAC, lighting, etc.) the user of CAD system creates the generalized automation network model (creates such objects as sensors, actuators, controllers and makes their logic connections). The final generalized automation network model is exported into the



ifcXML file. This model is a template for the full automation network model. A user of LonMaker imports the ifcXML file and uses the template as the basis for the construction of full model adding the data peculiar for LonWorks technology (Fig. 8).

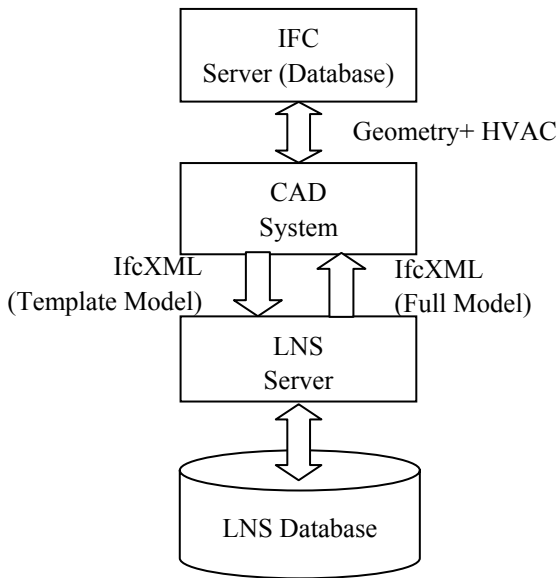


Figure 8. Integration on the basis of the creation of new model.

The template helps to create subsystems of the automation network (i.e. heating, lighting, ventilation, etc.) with territorial division (i.e. floor, room, etc.). By means of the template, logical connections between network variables can be automatically created.

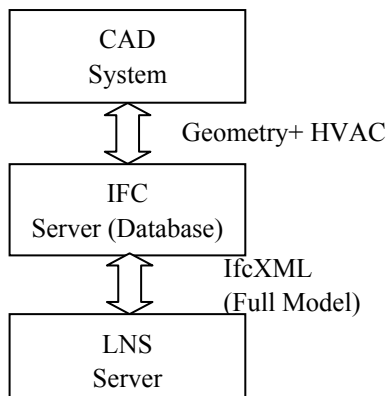


Figure 9. Integration on the basis of the creation of new model without LNS Database.

The technology described above can be used on the basis of already existing program systems (ArchiCAD and LNS Server). A variant when instead of LNS Database will be used IFC Database, could be of interest. Thus the full process of development of the automation network will be carried out in one place, and its results will be stored as IFC model. It means that LNS Server will use directly the information from the IFC model (Fig. 9). Thus it is possible to avoid a duplication of the data and to concentrate the building information in one place.

### 4.3 IfcXML in the integration

In the basis of the mechanism for definition of object properties lies EXPRESS definition. With the help of XML specification, EXPRESS schema can be transformed into the XML Document Type Definition (DTD) or XML Schema Definition (XSD). This process can be valid both for all IFC models and for their certain parts (Nisbet & Liebich 2004).

Table 2 shows a XML Schema Definition of the class *IfcDistributionControlElement* as an output from the general definition in IFC2x2.

Table 2. EXPRESS and ifcXML specifications of *IfcDistributionControlElement*.

<u>EXPRESS specification</u>
<p><b>ENTITY</b> IfcDistributionControlElement  <b>SUBTYPE OF</b> (IfcDistributionElement);            ControlElementId : <b>OPTIONAL</b> IfcIdentifier;  <b>INVERSE</b>            AssignedToFlowElement : <b>SET</b> [0:1] <b>OF</b>            IfcRelFlowControlElements <b>FOR</b>            RelatedControlElements;  <b>END_ENTITY</b>;</p>
<u>IfcXML specification</u>
<pre>&lt;xs:element name="IfcDistributionControlElement" type="ifc:IfcDistributionControlElement" substitution- Group="ifc:IfcDistributionElement" nillable="true"/&gt;   &lt;xs:complexType name="IfcDistributionControlElement"&gt;   &lt;xs:complexContent&gt;   &lt;xs:extension base="ifc:IfcDistributionElement"&gt;   &lt;xs:sequence&gt;   &lt;xs:element name="ControlElementId" type="ifc:IfcIdentifier" nillable="true" minOccurs="0"/&gt;   &lt;/xs:sequence&gt;   &lt;/xs:extension&gt;   &lt;/xs:complexContent&gt;   &lt;/xs:complexType&gt;</pre>

## 5 MERGING BUILDING AUTOMATION NETWORK IN IFC MODEL

In the previous section, CAD system has been chosen as an integration platform for LNS and IFC models. The general integration scenarios also have been described. For the realization of these scripts, it is necessary to select IFC subtypes and relationships which would represent an essence of objects and their communications in LNS model.

Communications between objects of the automation network model are presented by two levels: physical and logical (Fig. 10). The IFC2x2 provides the mechanism of connectivity management between objects (Liebich 2004).



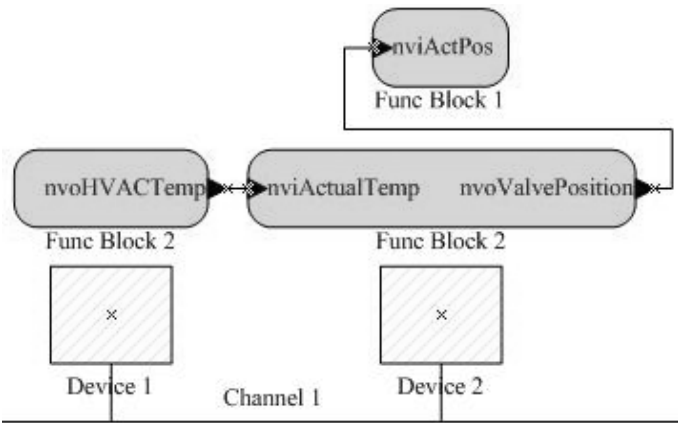


Figure 10. Objects communications in the automation network.

### 5.1 Physical layer of connections in the network

In LonWorks networks a physical connection exists between network nodes and data channels (i.e. cable, powerline, radio channel), and between network nodes and elements of control systems (sensors, actuators).

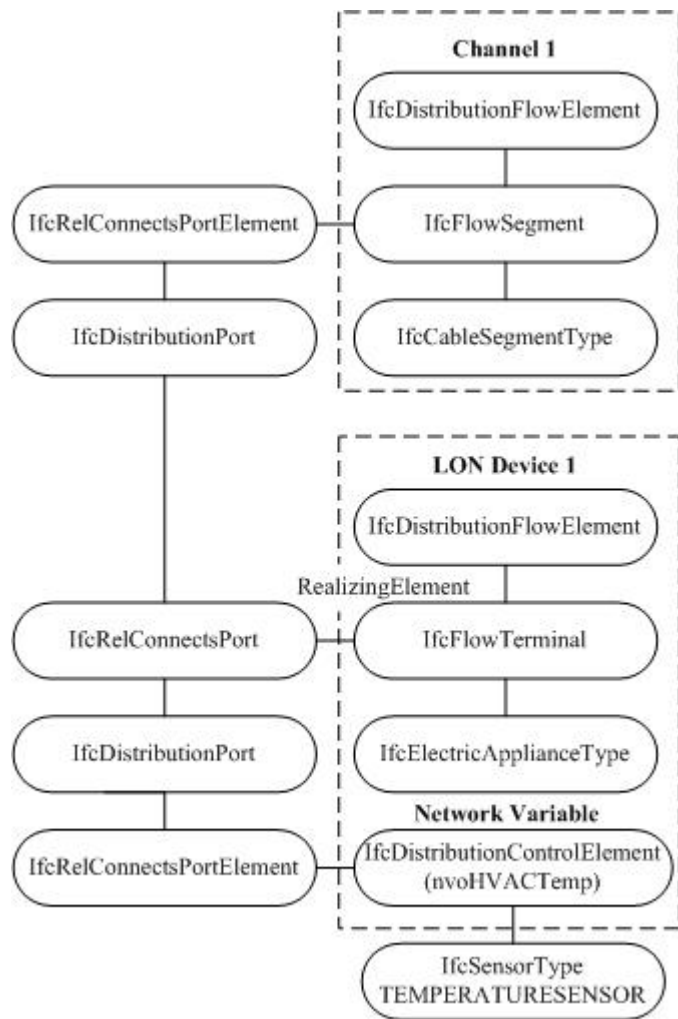


Figure 11. Definition of physical connections with ports.

Class *IfcDistributionFlowElement* (with subclasses *IfcFlowTerminal* and *IfcElectricApplianceType*) corresponds to the physical network node, class *IfcDistributionFlowElement* (with subclasses

*IfcFlowSegment* and *IfcCableSegmentType*) - to the data channel, class *IfcDistributionControlElement* (with subclasses *IfcSensorType*, *IfcControllerType* and *IfcActuatorType*) - to the network variable and the element of control systems.

In the IFC2x2 it is possible to operate physical connectivity through ports which are presented by classes *IfcDistributionPort*, *IfcConnectsPort* and *IfcRelConnectsPortToElement*. Here the port has no physical representation and is a virtual object for other object communications (Fig. 11).

All physical objects of the model can be connected with *IfcDistributionPort* by means of the relationship *IfcConnectsPort*. *IfcDistributionPort*'s can be connected among themselves through *IfcRelConnectsPortToElement*. Presence of the attribute *IfcConnectsPort.RealizingElement* means that connection is realized by means of some other element. Thus elements of control systems are physically connected to the data channel through LonWorks network nodes.

### 5.2 Logical layer of connections in the network

In LonWorks networks a logical connection is present at the level of communications between network variables (*IfcDistributionControlElement*). At the logical connection it is possible to do without use of ports that can be reached by introduction of the relationship *IfcRelConnectsElements* which will connect directly *IfcDistributionControlElement* objects (Fig. 12).

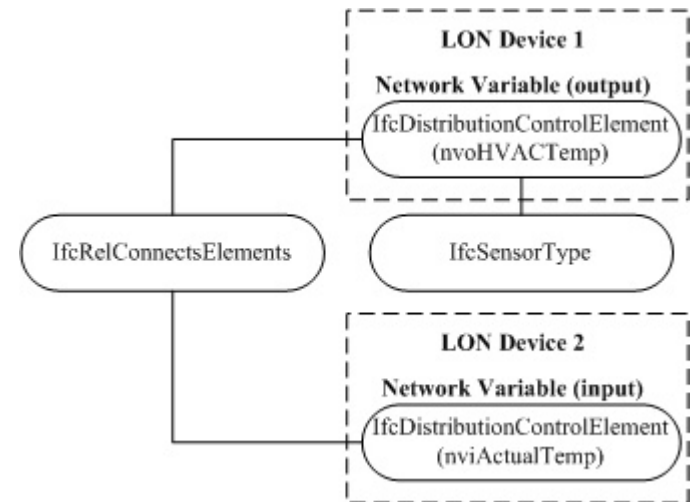


Figure 12. Definition of logical connections.

### 5.3 Measurement of network variables

Network variables are used for the measurement of process conditions in a building and for the control of these processes. For this purpose, the object *IfcMeasureWithUnit* can be attached to a network variable (object *IfcDistributionControlElement*) with the help of relationship *IfcRelAssignsToProcess*. Object *IfcMeasureWithUnit* has connectivity with

object *IfcValue* which contains the numerical value of a network variable (Fig. 13).

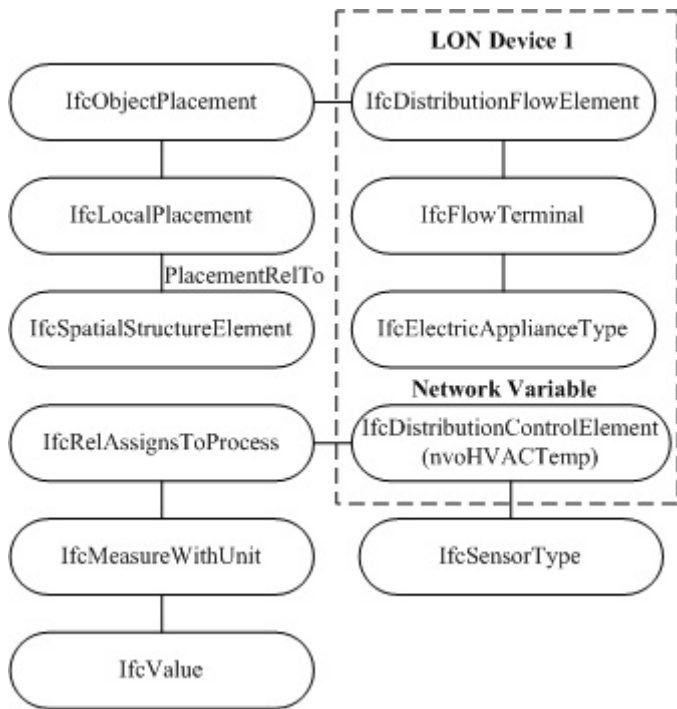


Figure 13. Definition of network variable value and device placement.

#### 5.4 LonWorks objects placement

All physical objects of an automation network (i.e. nodes, cable, sensor) have their certain placement in the building. They are defined by *IfcLocalPlacement* (a subtype of *IfcObjectPlacement*) which can determine an absolute placement, relative placement, or grid reference, with each defining the local coordinate system referenced by all geometric representations. The attribute *PlacementRelTo* of *IfcLocalPlacement* points to the *IfcSpatialStructureElement* (Storey, Space) (Fig. 13).

## 6 CONCLUSION

This paper has presented an approach for the integration of building automation network structures in construction projects. The integration concept is to merge the LNS model into the IFC model on the basis of the software environment of the CAD system. Two integration scenarios were discussed: the integration on the basis of an existing automation network model and the integration on the basis of the creation of a new automation network model. IFC subtypes and relationships are selected which represent an essence of objects and their communications in the LNS model. That opens up new possibilities for the interoperability through the building shared object model.

## REFERENCES

- Bozany, A. 2003. Integration of Building Automation Systems and Facility Information Systems. *Hungarian Electronic Journal of Sciences, HU ISSN 1418-7108*. <http://heja.szif.hu/INF/INF-030530-A/inf030530a.pdf>
- Dietrich, D. 2003. Paradigm Shifts in Automation-Building Automation, Part I. *Instrument Standardization & Metrology* 2: 12.
- Dietrich, D. 2003. Paradigm Shifts in Automation-Building Automation, Part II. *Instrument Standardization & Metrology* 3: 25.
- Dietrich, D.; Kastner, W.; Sauter, T. 2000. *EIB Gebäudebus-system*. Heidelberg: Hüthig.
- Dietrich, D.; Loy, D.; Schweinzer, H. 2001. *Open Control Networks*. Boston/Dordrecht/London: Kluwer Academic Publishers.
- Echelon Corporation 1999. *Introduction to the LonWorks System*. Palo Alto: Echelon Corporation.
- Echelon Corporation 2004. *LNS Programmer's Guide*. Palo Alto: Echelon Corporation.
- Geiger, A. 2001. *Produktdatenmodelle im Bauwesen. IFC im Praxistest*. Fachhochschule Karlsruhe. <http://www.iai-ev.de/download/pdf/Praktikumsarbeit.pdf>
- IAI 2005. International Alliance for Interoperability (IAI). <http://www.iai-international.org/>
- ISO/PRE PAS 16739 2005. IFC 2.x Platform specification.
- Liebich, T. 2004. IFC 2x Edition 2 Model Implementation Guide. International Alliance for Interoperability.
- Neuberg, F.; Rank, E.; Ekkerlein, C.; Faulstich, M. 2002. Internet based simulation of the resource requirement of buildings. *European Conference of Product and Process Modeling - eWork and eBusiness in AEC; Proc. ECPPM, Portoroz, 2002*
- Nisbet, N. & Liebich, T. 2005. ifcXML Implementation Guide. International Alliance for Interoperability.
- Schach, R.; Kabitzsch, K.; Knabe, G.; Hörschele, V. 2002. *Brücken zwischen Gebäudeautomation und Facility Management*. ISBN 3-86005-354-X: TU Dresden.

