ISO/IEC JTC 1/SC 25/WG 1 **N 844A**
WG 1 (Berlin, Wacks) 2
Date: Aug 3, 1999

ISO/IEC JTC 1 SC 25 WG 1
Interconnection of Information Technology Equipment
Home Electronic System

Title: TR 15067-1/PDAM 1: Objects and Messages for HES Building Control Systems, revision 5

Source: United States

Project: Working Group 1, 25.01.05.03

Requested Action: Review changes based on national body comments

Distribution: All WG 1 members

Notes:

This document was composed by Dr. Kenneth Wacks of the United States with inputs from the U.S. Technical Advisory Group. Suggestions and contributions were provided by Mr. Steven Bushby of the U.S. National Institute of Standards and Technology. Comments should be directed to Dr. Wacks at +1-781-662-6211, fax: +1-781-665-4311, E-mail: kenn@alum.mit.edu.

TABLE OF CONTENTS

FOREWORD

ISO (the International Organization for Standardization) and the IEC (the International Electrotechnical Commission) form the specialized system for world-wide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

This Working Draft of Technical Report 15067-1/PDAM 1 was prepared by the U.S. Technical Advisory Group to ISO/IEC JTC 1/SC 25/WG 1, *Interconnection of Information Technology Equipment* / *Home Electronic System*

INTRODUCTION

This document was prepared to fulfill the charter of Working Group 1 to specify HES for Command, Control, and Communications in Buildings (C^3B). At the June 1996 meeting of WG 1, the WG 1 C^3B Working Party agreed on an architecture based on proposals from France and the United States. This architecture models a building control system as a hierarchy of three levels: Management, Automation, and Field. The Field level is implemented with the HES Class 1 protocol.

Revision 5 of this document contains changes recommend by national bodies in SC 25 N 535 and by experts at a WG 1 meeting in January 1999. WG 1 would appreciate comments by manufacturers, trade associations, and standards organization involved in building automation regarding possible enhancements to this document.

1. Scope

This document proposes building control objects and messages appropriate at the automation and field levels. These objects and messages are based on selected features from the following four prominent communications protocols:

- BACnet™ Protocol, created by ASHRAE, the American Society of Heating, Refrigerating, and Air-Conditioning Engineers for building automation

- CAB Protocol, the Canadian Automated Building Protocol, created by Public Works and Government Services Canada for management of public buildings.

- CEBus® Protocol, created by the EIA, Electronic Industries Alliance (U.S.A.), and approved as an ANSI standard (American National Standards Institute) for home automation.

- The field level objects defined by CEN/TC247 in proposed European standard, ENV 13154-1. ENV-13154-1 includes an interface to the field level that has been adopted for the BatiBUS, European Home Systems (EHS), and European Installation Bus (EIB) protocols. This interface is part of the specification for a new system issued by EHBESA, the European Home and Building Electronic System Association. The working name for the merger of these three protocols is "Convergence." (A final name is expected soon.)

WG 1 has maintained liaison with members of ISO/TC 205/WG 3, and both groups are coordinating proposals. WG 1 is specifying interfaces at the Field Level, as explained in this document. Recommendations for enhancing the HES Class 1 protocol are presented. This set of objects and messages for C^3B is intended to augment those objects and messages already specified in proposed IS 10192, the *HES Application Model - Application Services and Protocol*.

2. The HES C^{3}B Model

2.1 HES C^{3}B Levels

The HES C^{3}B model represents an amalgamation of proposals submitted by French and United States experts. The respective models are included for information in Annex B.

Figure 1 illustrates the HES C^{3}B model features. This hierarchical description is provided for logical clarification of building control functions. It conforms to building automation practices that favor a hierarchical structure. Although Figure 1 is primarily a logical diagram, it also shows communications elements. The hierarchy consists of three logical layers:

- Field Level

  The lowest level in the hierarchy, the Field Level, is populated primarily by sensors and actuators. Control elements may be located at this level for fully distributed control. This is illustrated by the two Application Controllers on the right. Typical building automation applications include security, lighting, HVAC (heating, ventilating, and air-conditioning), vertical transport (elevators and escalators), and power distribution.

- Automation Level

  Control elements in building automation have traditionally been located at his level. This is especially the case when sensors and actuators do not include local intelligence. Thus, the two Application Controllers on the left side of Figure 1 are concentrators for less intelligent devices at the Field Level. Also, related functions located in different parts of the building may be coordinated at the Automation Level.

- Management Level

  This is where the building systems spanning many applications are managed and scheduled. Typically, operator stations are located here. However, autonomous management without operator intervention is possible, depending on the implementation.

Devices may be physically connected to a single controller, or may be connected via a Communications Link to the Automation Level where Application Controllers may reside. The Communications Link is a generic communications element. This element may be a router if the same upper layer protocols are used at the adjacent levels. If not, a gateway may be required. Similar technologies may be applied at the Automation and Field Levels to minimize complexity of the Communications Link. Where possible it is recommended that router functions rather than gateway functions be incorporated into the Communications Link between the two lower levels to enhance network integration.

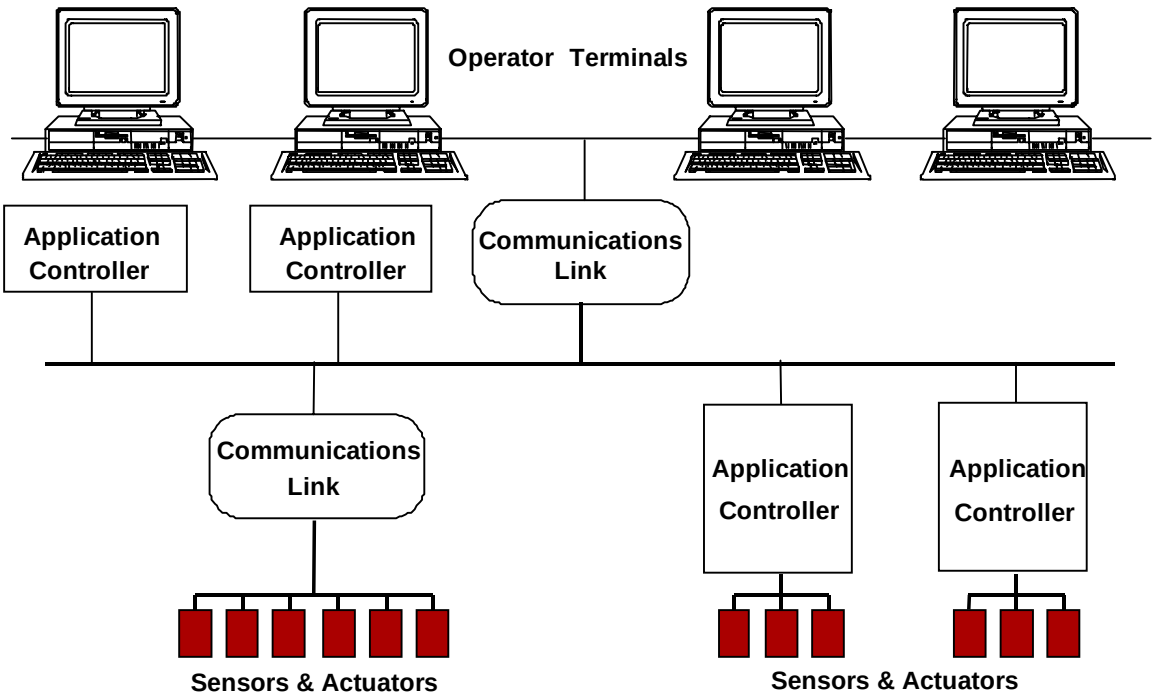

Figure 1 – HES Building Control Architecture

2.2 HES Field Level

Standards planned by ISO/TC 205/WG 3 will specify the Management and Automation Levels. HES standards specifies interfaces at the Field Level.

As noted, the Field Level is based on the HES Class 1 protocol. Therefore, sensors and actuators contain HES Interfaces with Type A or Type B conformance. Type A Conformance accommodates operation on a variety of control network provided a Universal Interfaces (UI) is embedded in the field level device.. Devices designed for a particular control network may be made HES Type B Conformance, thereby not requiring an explicit UI. Devices that do not support a UI and are not Type B Conformant may share a concentrator that functions as a single HES device for multiple low-cost sensors and actuators. This is illustrated in Figure 2.

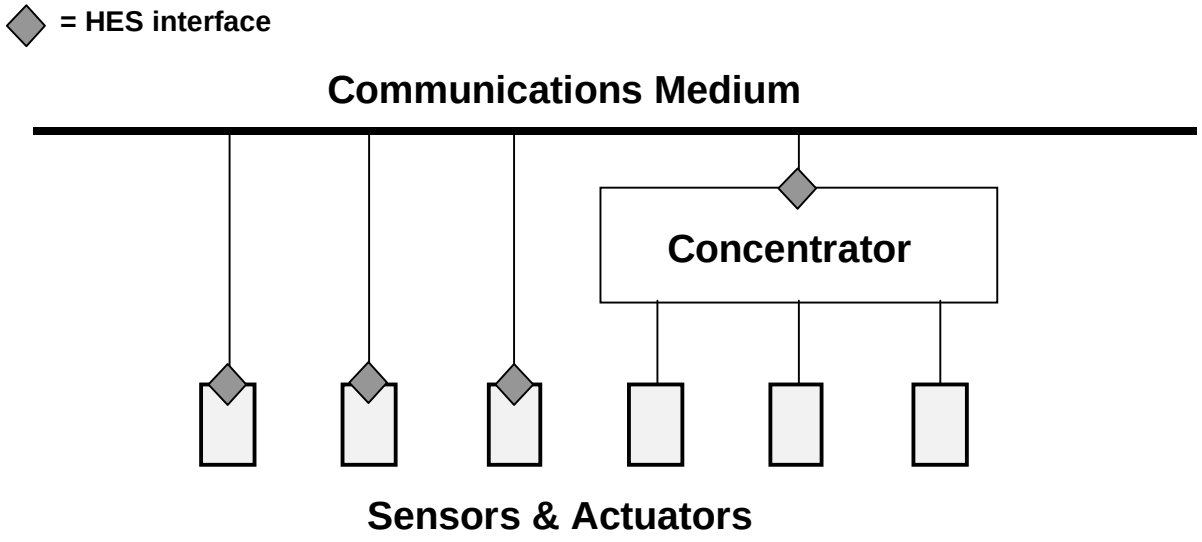

Figure 2 – Concentrator for Low-Cost Devices

3. HES C^{3}B Objects

3.1 Automation Level Objects

WG 1 expects that ISO/TC205/WG3 will incorporate the following Automation Level objects from the BACnet standard[1]. Issues at the Automation Level are out of scope for WG 1.

- Command: A list of commands, each of which causes specified object properties to assume specified values; acts like a macro; a parameter in this command specifies which list is to be executed

- Program: Externally visible parameters of an application program running on a device such as a controller

- Loop: An object in a device that forms a feedback control loop using the proportional, integral, and/or derivative (PID) control mode

- Multi-state Input: One of N states of a device, such as a controller; the states are determined by multiple binary or analog device inputs or by an internal algorithmic process

- Multi-state Output: One of N states desired for one or more physical outputs or for processes within a device, such as a controller; useful for a controller to communicate with multiple actuators

- Calendar: A list of dates relevant for building control operation

- Schedule: A list of values to be sent to specified object parameters on a scheduled basis; schedules are set by day over a range of dates with exceptions allowed

- Notification Class: A list of devices to receive notification of events; these devices must have identical requirements for event notification

- Group: A list of objects and properties to provide a logical group that is useful, for example, in updating a graphical representation on an operator terminal.

---

[1] ANSI (American National Standards Institute) / ASHRAE (American Society of Heating, Refrigerating, and Air-Conditioning Engineers), *Standard 135-1995: BACnet™ - A Data Communication Protocol for Building Automation and Control Networks*, available from ASHRAE Customer Service, 1791 Tullie Circle, NE, Atlanta, Georgia 30329, USA, 1995. Telephone: +1 404 636-8400; fax: +1 404 321-5478; http://www.ashrae.org. Approved by ASHRAE on June 29, 1995 and by ANSI on December 19, 1995.

– File: An object that describes properties of a data file accessed with BACnet file services (defined below)

3.2 Field Level Objects

3.2.1 Categories of Field Level Objects

The following Field Level objects for HES Class 1 protocol are specified. Note that the features of each object are not mandatory for HES devices.

The objects are grouped into the following categories:

- Management of objects by a device
- Input objects
- Output objects
- Internal device functions
- Event management functions

These objects have been selected from among the 18 BACnet objects, 25 CEBus objects, the six objects in the CAB Protocol that are not found in the BACnet protocol, and the seven field level objects defined by CEN/TC 247 in ENV 13154-1. Application-specific objects have been omitted. They will be reconsidered after the basic set of objects are specified. Please note that the exact object names may be adjusted to align with other protocols and conventions.

3.2.2 Management of Objects by a Device

– Node Control

Contains product information and status, such as power on or off, device on-line or off-line, manufacturer, model number, serial number, product class, product address, functionality, configuration status, keys for authentication, relevant software/firmware version data, subunits (logical or physical section of a real device) contained in device (subunits are called Contexts in CEBus)

– Subunit Control

List of objects in the device subunit

3.2.3 Input Objects

– Binary Sensor

A device that can report one of two states of an input. The state change in the device may be momentary or static.

– Multi-State Sensor

A device that can report one of N states of an input. The state change in the device may be momentary or static.

– Analog Sensor

A device that can report a value in defined increments within a specified range of values.

3.2.4 Output Objects

– Binary Switch

A device that can be set in one of two states, such as a relay. It may be momentary or toggled. Toggling a binary switch is not recommended because it depends on the previous state without an absolute determined value. Three types of binary switches are noted:

1) Absolute Switch: Set in one of two determined states.
2) Toggle Switch: Changed from one to the other state.
3) Momentary Switch: Changed from one to the other state temporarily

– Multi-Throw Switch

A device with one output that can be set in one of N states, such as an N-throw switch. It may be momentary or toggled.

– Matrix Switch

A device with N inputs, each of which can be connected to one of M outputs or to no output. A connection may be momentary or toggled.

– Multiplane Switch

This object defines a matrix switch with multiple gangs, each of which can be set separately or all at once.

– Analog Control

A device that can be set in defined increments within a specified range of values. The rate at which the device can be changed may be limited.

– Ganged Analog Control

This object defines a group of analog controls that can be set simultaneously, such as an equalizer.

3.2.5 Internal Device Functions

– Binary Value

ACTIVE or INACTIVE state of a binary parameter in the memory of a device

– Analog Value

Numeric value of a parameter in the memory of a device

– List Memory

A block of memory containing a list of items of a specified maximum length.

– Data Memory

Memory containing data organized into blocks or records.

– Counter

An up or down counter toward a specified target value. It may count events or time intervals.

– Clock

A real-time clock plus calendar with day of week, time zone, and seasonal offset (daylight savings in U.S. or summer time in Europe)

3.2.6 Event Management Functions

– Event Enrollment

Defines criteria for the occurrence of an event and a list of devices to be notified

Many building control devices operate by monitoring a physical parameter, such as temperature or pressure. They issue a message to a controller upon occurrence of an *event*. An event may be a pre-specified change in the value of a parameter in an object that causes a message to be sent to one or more designated recipients. The conditions for issuing an event message and the recipients are defined in the Event Enrollment object.

– Change-of-value reporting

  Notify controller (or another device) when a parameter changes by an amount greater than a specified increment

– Intrinsic reporting

  Notify a controller or another device when a standardized algorithm intrinsic to a fixed set of properties of an object generates an event notification

– Alarm

  When an event is defined by the Event Enrollment Object, it may be characterized as an *Alarm*. An Alarm event is a high priority event intended for notifying humans.

4. HES C^{3}B Methods

4.1 Expansion of HES Methods

Three methods and associated command language service elements had been defined for HES: Read, Write, and Report. WG 1 was concerned about mapping the higher-level service elements found in BACnet, CAB Protocol, or CEBus into multiple HES service elements and the impact on the timely performance of application functions. Therefore, HES includes additional methods especially for event and file management.

4.2 New HES Methods

The following service elements have been selected from among the 35 BACnet service elements, 24 CEBus methods, the 21 service elements in the CAB Protocol that are not found in the BACnet protocol, and the four methods of ENV 13154-1. The full set of BACnet, CEBus, and CAB Protocol service elements is listed in Annex A.

4.2.1 Reading and Writing

- read variable

  Read the value of a specified variable.

- read-multiple variables

  Read the values of specified variables.

- write variable

  Write a specified value into a specified variable.

- write-multiple variables

  Write multiple specified values into separately specified variables.

- read-file

  Read a specified amount of data from a specified file contained in a specified object.

- write-file

  Write a specified amount of data to a specified file contained in a specified object.

NOTE: Open and Close functions are not included because the read and write file functions are atomic.

4.2.2 Modifying Variables

NOTE: In using the following commands, the user should be tracking the absolute state of the variable being altered.

- increment

  Increment the value of a specified instance variable by a specified amount.

- decrement

  Decrement the value of a specified instance variable by a specified amount.

4.2.3 Reporting

- subscribe

  Request to be notified when a specified event occurs in a specified object.

- report

  Notify the subscribed objects that a pre-defined event has occurred.

- confirmed-report

  Notify the subscribed objects that a pre-defined event has occurred and request confirmation that the report was received.

- report-confirmation

  Confirm that a confirmed-report was received.

- subscription-list

  Request a list of all objects enrolled to receive a specified event report from a specified object.

- event-status

  Request that an object report the state of the variable that is being watched for issuing an event.

4.2.4 Security

– set-access

  Set access control in a specified object. (What category of messages are acceptable; this could imply which operators may originate messages.)

– clear-access

  Remove access control restrictions on a specified object. The message containing clear-access must have the proper access authority for this action.

– set-password

  Set a password in an object. Subsequent messages must contain this password to access the object. Also, the message must have the proper access authority.

– clear-password

  Remove a password to access an object. The message containing clear-password must have the proper access authority and password for this action.

– set-key

  Establish an authentication key between a sender and a receiver object.

– authenticate

  Request that the sender of a message be authenticated.

NOTE: The algorithms for access and password control and authentication are to be determined. They may be derived from a combination of BACnet, CAB Protocol, and CEBus.

4.2.5 Macros

– macro

  Define a variable to represent a frequently sent message.

– alias

  Define a short variable to substitute for a character string.

5. Open Issues

- Support of dynamic object creation.
- Object management: Identifying and locating particular objects.
- Time service to synchronize timing.
- Device management: Initialize, start, stop a device.
- Specialized services for controllers and gateway (as in the CAB Protocol).
- Support of conditional and iterative methods.

6. Annex A – Service Elements from other Protocols

6.1 BACnet Protocol

The following 35 service elements are defined in BACnet to alter or report values of the parameters defined in the objects and to manage devices:

6.1.1 Reading and Writing Object Parameters

- ReadProperty: Read a specified property of a specified object
- ReadPropertyMultiple: Read a list of specified properties from a list of objects
- ReadPropertyConditional: Read a list of specified properties of all objects that meet a specified condition
- WriteProperty: Write a value to a specified property of a specified object
- WritePropertyMultiple: Write a list of values to a list of specified properties of specified objects

6.1.2 Event Management

- GetEnrollmentSummary: Request a list of all event enrollment objects that meet specified search criteria
- GetAlarmSummary: Request a list of all event enrollment objects and objects supporting intrinsic reporting of type Alarm that are not in the NORMAL state
- AcknowledgeAlarm: A response to an event notification that an event defined as an Alarm has been received by an operator
- ConfirmedEventNotification: Notify a device about an event and request confirmation that this notification was received
- UnconfirmedEventNotification: Notify all or a group of devices about an event
- SubscribeCOV: Request notification upon occurrence of a change-of-value of a specified parameter in a specified object; typically a controller issues this request
- ConfirmedCOVNotification: Notify a device that previously has requested SubscribeCOV about a change-of-value and request confirmation that this notification was received

– UnconfirmedCOVNotification: Notify a device that previously has requested SubscribeCOV about an change-of-value event

6.1.3 File Management

– AtomicReadFile: Open a specified file, read a specified number of bytes starting from a specified location or a specified number of records from a specified starting record, and close that file; no other file access is permitted during this operation

– AtomicWriteFile: Open a specified file, write a set of bytes into a specified location at the specified starting location or a specified number of records from a specified starting record, and close that file; no other file access is permitted during this operation

6.1.4 Managing Objects

– CreateObject: Create a specified object in a specified device

– DeleteObject: Delete a specified object from a specified device (unless it is protected from deletion)

– AddListElement: Add one or more list elements to a specified property of a specified object, provided that property is a list; can be used to add members to a group

– RemoveListElement: Remove one or more elements from a specified property of a specified object, provided that property is a list

6.1.5 Managing Devices

– ConfirmedPrivateTransfer: Initiate a proprietary service in a remote device and request a confirmation that this service was performed

– UnconfirmedPrivateTransfer: Initiate a proprietary service in a remote device

– ConfirmedTextMessage: Send a text message to another device and request a response that the text was received

– UnconfirmedTextMessage: Send a text message to one or more devices

– Time Synchronization: Send a message containing the current time to one or more devices

- Who-Is; I-Am: Companion services for identifying Device Objects that are on the same network within a specified range of device addresses; these services can also be used in an internetworking environment linked by routers.

- Who-Has; I-Have: Companion services for locating a particular object by name or identifier within a specified range of addresses; these services can also be used in an internetworking environment linked by routers.

- ReinitializeDevice: Instruct a device to restart; *coldstart* as if powered up, or *warmstart*, from a vendor-defined initial state; used for diagnostic purposes

- DeviceCommunicationControl: Instruct a remote device to stop sending or receiving commands, except for this command and for ReinitializeDevice; used for diagnostic purposes

6.1.6 Virtual Terminal

An additional set of service elements is defined for Virtual Terminal services. These services are intended for linking application programs in different devices. The devices communicate by exchanging character-oriented data. The model for a virtual terminal allows one application program to act as a "terminal emulator" that interacts with an "operator interface" in another application program. The service elements include:

- VT-Open: Open a Virtual Terminal session between applications

- VT-Data: Exchange data between applications opened

- VT-Close: Terminate an opened Virtual Terminal Session

6.1.7 Security Services

BACnet supports peer and data origin authentication by defining one device as a key server. The BACnet standard references the 56-bit Data Encryption Standard (DES), designated ANSI X3.92 by the American National Standards Institute. Each device participating in security is assigned a key and must support the following services:

- RequestKey: Request that a Session Key be established between two devices; this request is enciphered with a Private Key using the DES

- Authenticate: Authenticate a peer device, an executing device, or a human operator using the Private Key or Session Key

6.2 CAB Protocol

The CAB Protocol adds the following unique services:

6.2.1 Access Authority

There are 16 categories of access control for users, services, and points. Each category is represented by a bit in a 16-bit mask. Access is allowed when the following evaluates true ($\cap$ is the logical AND function[2]):

$$[(\text{service mask} \cap \text{operator mask}) \cap \text{point attribute mask}] <> 0$$

– CAB-Confirmed-SignOn

An operator at an OWS requests access to a network node device (a CAB-Gateway or a controller programmed with the CAB Protocol).

– CAB-Confirmed-SignOff

An operator cancels access control for a network node device.

– CAB-Confirmed-Inq-SignOns[3]

The sender requests the data about each operator signed-on. Among these data are the operator name, temporary number assigned for the present operating session, and the network node device name and address.

6.2.2 Operator Access Control Utilities

– CAB-Confirmed-Inq-Pass

Request operator password for a specified operator (or all operators). The password is encrypted using a key passed in the argument.

– CAB-Confirmed-Set-Pass

Set the password for a specified operator.

---

[2] The logical AND function is: 1 AND 1→1, 1 AND 0→0, 0 AND 1→0, 0 AND 0→0.

[3] *Inq* in the command means Inquire.

– CAB-Confirmed-Del-Pass

Delete the operator access data for a specified operator.

6.2.3 Event and Exception Management

– CAB-Confirmed-Ack-Notify

A device notifies all subscribers to a critical alarm event which OWS has acknowledged the alarm. NOTE: The BACnet notification service offers similar functionality.

6.2.4 Device Management

– CAB-Confirmed-Inq-Controllers

Request a list of controllers connected to a network node device.

– CAB-Confirmed-Inq-Priv

Request a list of access control authority masks for the CAB application protocol services.

– CAB-Confirmed-Inq-Systems

Request a list of systems defined on a specified network node device.

6.2.5 Data Collection

– CAB-Confirmed-Set-Collection

Define a data collection entry of a specified collection type for a specified point and attribute (attribute ID is null for Control Loop Plots).

– CAB-Confirmed-Inq-Collection

Request information from a specified device about a data collection type. Information includes the operator requesting the data collection, the OWS where the request was made, and the data file ID for the accumulated data.

– CAB-Confirmed-Can-Collection

Cancel data collection of specified collection type and delete any accumulated data.

– CAB-Confirmed-CLP-Notify

Notify a device that control loop plot data is available.

Note that BACnet does not contain similar explicit services for data collection. However, BACnet allows Program objects which could contain code to implement data collection. Also, BACnet is debating adding a trend object.

6.2.6 Memory Management

– CAB-Confirmed-Save-Controller

Save in a specified file the memory contents of a specified controller.

– CAB-Confirmed-Restore-Controller

Restore the memory of a specified controller from a specified file.

– CAB-Confirmed-Inq-Controller-Save

Request the following information about the most recent memory save: controller ID, product type, product text description, name of the operator requesting the save, file name where data were saved, date and time the save was completed, and the save status: successful, failed, not performed, in progress.

– CAB-Confirmed-Inq-Controller-Restore

Request the following information about the most recent memory restore: controller ID, product type, product text description, name of the operator requesting the restore, file name where data were saved, date and time the restore was completed, and the restore status: successful, failed, not performed, in progress.

– CAB-Confirmed-Inq-Gateway-Save-Resp

Gather information for a complete or partial image of a CAB-Gateway. The image is saved using the file transfer services.

– CAB-Confirmed-Gateway-Restore-Info

Coordinate a full or partial CAB-Gateway restoration (to be performed using file management services).

– CAB-Confirmed-Gateway-Save-Rest-Done

Notify a CAB-Gateway that a save or restore is completed.

6.3 CEBus Protocol

6.3.1 Reading and Writing

– SetOFF, SetON

  Used to set a Boolean instance variable to either 1 (setON) or 0 (setOFF)

– GetValue

  Used to retrieve the contents of an instance variable from an object.

– GetArray

  Used to return a specified number of elements from an instance variable starting a specified number of elements from the start of the instance variable.

– SetValue

  Used to store an argument value into an object instance variable.

– SetArray

  Used to store one or more elements into an instance variable starting at specified number of elements from the start of the instance variable

– compare

  Compare the value of a specified instance variable to a specified argument.

– report

  Used to initiate a reporting message from an object, regardless of any predefined conditions in the instance variable controlling event reporting.

– nop

  The nop method performs no operation on an object. This method stimulates an object to return a result packet.

6.3.2 Instance Variable Modifications

– add

  Add the values of specified instance variables.

– subtract

  Subtract the values of specified instance variables.

– increment

  Increment the value of a specified instance variable by a specified amount.

– decrement

  Decrement the value of a specified instance variable by a specified amount.

– swap

  Swap the values of two specified instance variables.

– copyValue

  Write an argument or the value of a specified instance variable into a specified instance variable.

6.3.3 Macro Management

– alias

  Used to equate a commonly used character string to a one byte alias ID. The alias ID is used in subsequent messages to avoid transmitting the entire character string.

– build

  Used to create a shorthand identifier for a message frequently sent to an object.

6.3.4 Network Resource Management

– inherit

  Used to assign a resource value (Area Address, Unit Address, Group address, or data channel band number) to an object.

– disinherit

  Used to request an object to disinherit a resource (Group address or Data Channel.)

6.3.5 Complex Methods

– if

  Initiate a conditional execution of messages.

– do

  Initiate a conditional iteration of messages.

– while

  Initiate a conditional iteration of messages. Test the exit condition before the first execution.

– repeat

  Repeat the execution of a message a specified number of times.

– exit

  Terminate the execution of a loop.

6.4 Field Level Objects in ENV 13154-1

6.4.1 Object Structure

The object model defined in ENV 13154-1 specifies the system neutral exchange of data at the field level. Data contained in objects are organized in *properties*. An object may contain three types of properties:

– Mandatory properties must be defined and accessible in a standardised manner through the network

– Optional properties may be defined and, if defined, may be accessible in a standardised manner through the network

– Engineerable properties may be defined, but are never accessible in a standardised manner through the network

6.4.2 Methods

The following methods are defined:

- ReadProperty

  This service reads a property of an object-instance. The read service is used by management communications for one-time reading of any property of any object. It is also used to poll for properties in a repetitive manner. The read service may also be used by engineering communication to build-up or update the engineering database.

- WriteProperty

  This service writes a property of an object instance. The write service is used by the management communication to command objects and by engineering communication to set-up or change parameter-values in a server device.

- COVNotification

  This service unsolicitedly offers the property value of a server object to a peer or client. The notification service is used by a server to inform its peer process-devices or superordinate clients about changes of values or states of certain properties of an object.

- AlarmNotification

  This service offers alarm-state related information to clients without solicitation. It is used to inform managment clients about an alarm condition in the process of a server.

7. Annex B – Alternative C^{3}B Models Considered

Figure B-1 shows the United States proposal for the HES C^{3}B architecture, while Figure B-2 shows the French proposal. A combination, accepted by WG 1 as the standard, is presented in Figure 1. In Figure B-1, the logical control of groups of sensors and actuators is based on physical links to specific controllers. The French proposal places the controllers at the Automation Level while the sensors and actuators remain at the Field Level. The association between controllers and devices is determined by the logical configuration of the network. The dotted lines in Figure B-2 illustrate that Controller I manages Devices A, B, and C, while Controller II manages an application that involves only Devices A and B. Note that that it is possible for two controllers to manage a single device when distinct applications share devices. Such controllers should be programmed to avoid device conflict

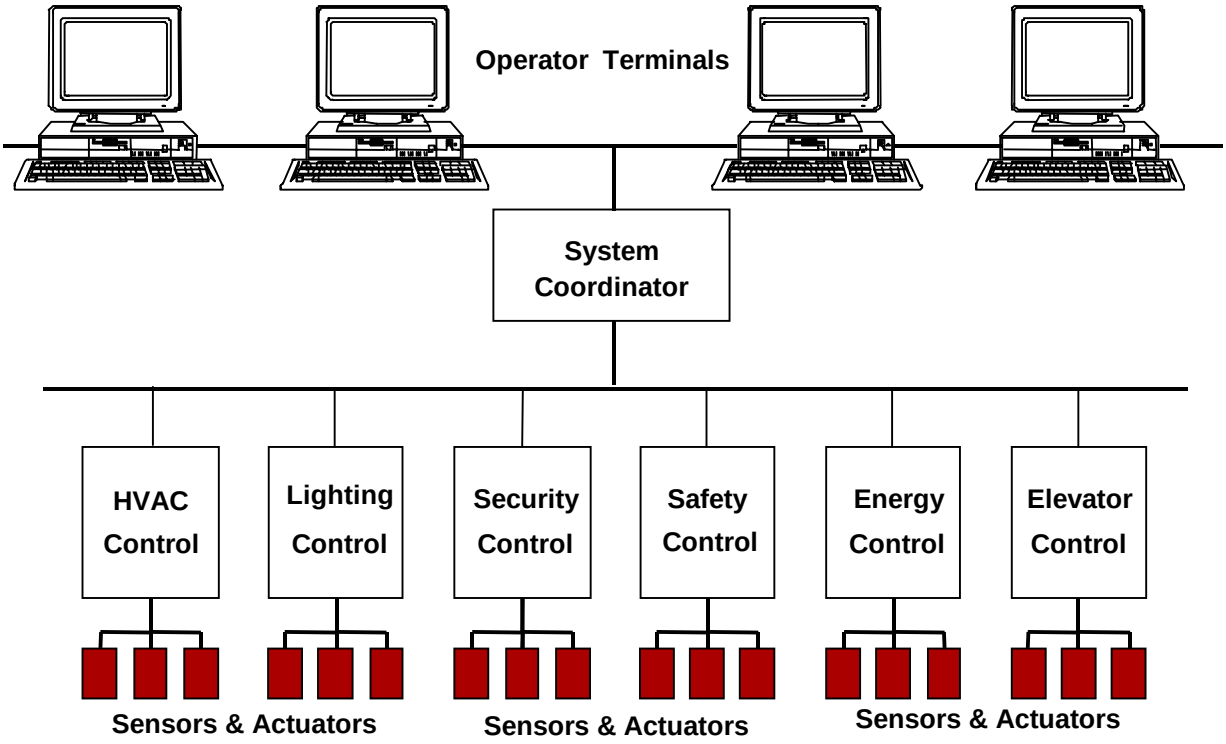

Figure B-1 – United States Proposal for Building Control Architecture

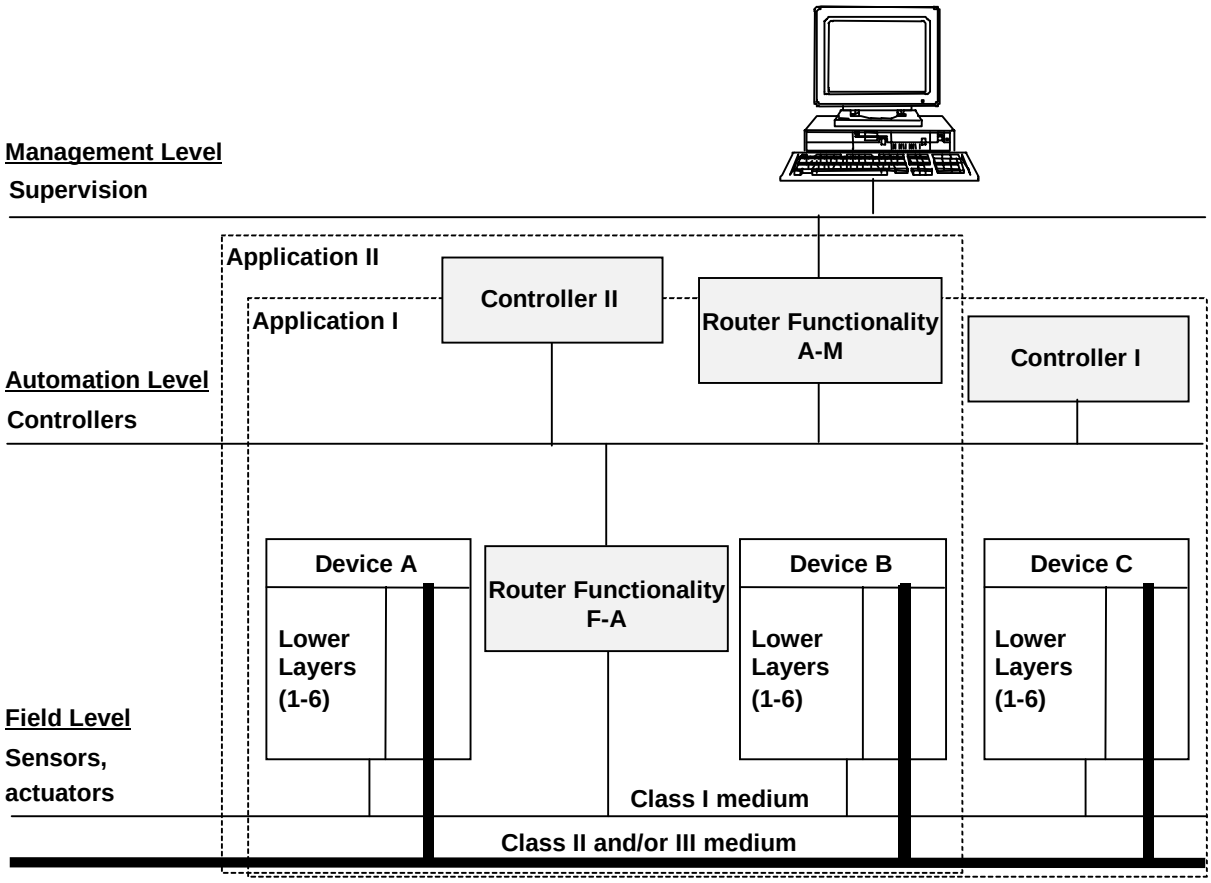

Figure B-2 – French Proposal for Building Control Architecture