This work-in-progress paper was presented as part of the main technical program at IEEE ETFA'2011

Integrating CCTV Systems into BACnet

Christian Mauser, Wolfgang Granzer, Wolfgang Kastner
Vienna University of Technology
Automation Systems Group
Treitlstrasse 1-3, A-1040 Vienna, Austria
{cmauser,w,k}@auto.tuwien.ac.at

Abstract

Closed-Circuit Television (CCTV) systems are used in modern buildings for different purposes. By the use of smart cameras, dedicated safety or security-critical events that may lead to an alarm condition can be detected autonomously (e.g., safety events like fire or security events like glass break). Integrating CCTV systems into building automation systems provides the opportunity to react to critical events by, for example, informing the operator about the occurrence of such events, providing relevant video data, and most important concurrently initiating appropriate actions (e.g., smoke extraction in case of fire). This paper focuses on the integration of such smart camera systems into BACnet based networks. At first, a short introduction to BACnet and its interworking model is given. Afterwards, a possible way of integrating CCTV systems into BACnet is presented. The functionality of this model is shown by a proof-of-concept implementation which is also described in this paper.

1. Introduction

Today's Closed-Circuit Television (CCTV) systems are based on analog technologies. As in numerous other fields of communication technology a trend towards IP-based solutions can be identified. This is also true for monitoring and recording systems. These systems have to handle large amounts of video data, even though data is usually only relevant in case of some specified events (e.g., in the case of security or safety events). Therefore, another trend is to add intelligence i.e. computational power and image processing functions on-the-spot to smart cameras (cf., [7, 12, 10, 6]). As a result, the amount of data which has to be propagated over the network can significantly be reduced since video data is only transmitted if a critical or abnormal situation is detected by the smart camera. This enormous reduction of transmitted data provides the possibility to integrate CCTV systems into an existing Building Automation System (BAS) [8] where the used control networking technologies are inherently limited to handle only small amounts of data.

Integrating CCTV with smart cameras into a BAS leads to numerous benefits. For example, smart cameras are able to detect safety critical events like fire or water ingress as well as security critical events like glass breaking or suspicious motion. By integrating the CCTV system into a BAS, the propagation of relevant visual information to the control center can be done just in time to derive further actions by the operator. However, in addition to a simple notification to the system operator, the system can concurrently initiate immediate reactions to these events. For example, in the case of fire the Heating, Ventilation, and Air Conditioning (HVAC) system can perform a smoke extraction by starting the ventilation system. Furthermore, CCTV systems can be used to detect the presence of people which act as additional input parameter for the HVAC system (sensor fusion). For example, the amount of people within a room can be used for smart HVAC control. Moreover, it is possible to reduce the amount of sensors since already existing smart cameras can assume sensor functionality (sensor sharing).

Typical BAS are partitioned into field level, automation level and management level. For example, [11] shows a possible way to integrate CCTV systems on the field/automation level into the open BAS standard KNX. This paper focuses on the integration on the automation/management level into the open BAS standard BACnet [1, 2]. In the following, a short introduction to BACnet and the corresponding interworking model is given in the next section. Then a new extension of the current BACnet object model that provides the opportunity to integrate smart cameras into BACnet is described. The paper is concluded with an ongoing proof-of-concept implementation.

2. BACnet interworking model

The Building Automation and Control network (BACnet) protocol is an open data communication standard for BAS of all sizes and types. BACnet was developed by the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE). Since the first release BACnet is under continuous maintenance. The current valid standard is BACnet 2008 [1] and ISO 16484-

978-1-4577-0018-7/11/$26.00

5:2010 [2].

BACnet is based on a collapsed ISO/OSI-architecture consisting of a physical, data link, network, and application layer. Regarding physical and data link layer, multiple options are defined: Ethernet, ARCNET, LonTalk, ZigBee, Master-Slave/Token-Passing (MS/TP) based on EIA-485, Point-To-Point (PTP) based on EIA-232, and BACnet/IP. The BACnet network topology is very flexible due to the different network options (cf. Figure 1). BACnet devices are coupled to *physical segments*. One

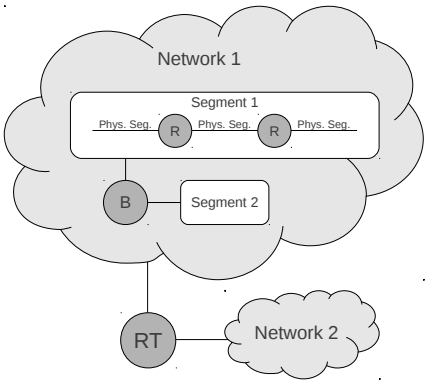

Figure 1. BACnet internetwork

or more physical segments connected by Repeaters (R) on the physical layer form a *segment*. Segments in turn can be interconnected by Bridges (B) on the physical and data link layer to build a *BACnet network*. Furthermore two or more networks can be connected by BACnet Routers (RT) to a so called *BACnet internetwork*.

To describe the logical structure of BACnet control data, BACnet provides an object-oriented approach. Each BACnet device consists of a set of *BACnet objects* which correspond to the data points of the control application. Each object in turn is described by a set of *BACnet properties*. The standard provides predefined types for both objects and properties. For example, Table 1 lists a few properties of the `Analog Input Object Type`, which represents a sensor input, together with the corresponding property data types.

Property	Data type
`Object_Identifier`	`BACnetObjectIdentifier`
`Object_Name`	`CharacterString`
`Object_Type`	`BACnetObjectType`
`Present_Value`	`REAL`
`Description`	`CharacterString`
⋮	⋮

Table 1. `Analog Input Object` **properties**

An important special type of object is the `Device Object` which represents the externally visible characteristics of a BACnet device. Therefore, every BACnet device has to contain exactly one `Device Object`. Furthermore, there are some properties which uniquely describe a BACnet object and hence must be part of every object type definition. These properties are `Object_Identifier`, `Object_Name`, and `Object_Type` (cf. the first three properties in Table 1).

The predefined object types in the earlier versions of the standard are generic types (e.g., `Analog Input Object Type`, `Binary Output Object Type`) whereas nowadays a trend to application specific object types can be identified. Typical examples are the `Access Door Object Type` (already part of the standard) and the `Lighting Output Object Type` (defined in BACnet Addendum i). In addition to the predefined standard object and property types, proprietary object types as well as proprietary property types may be defined if needed. Information exchange and data manipulation in BACnet follows a client/server approach. The communication between the devices is done by calling services which are grouped into five categories. *Alarm and Event Services* deal with the management of events i.e. significant state or value changes in BACnet objects. *File Access Services* are used to access and manipulate files i.e. collections of octets with arbitrary meaning contained in BACnet devices. *Object Access Services* are used to access and manipulate BACnet objects and their properties. *Remote Device Management Services* allow remote controlling of BACnet devices and requests about other existing BACnet devices and their objects in the (inter)network. Finally the *Virtual Terminal Services* can be used to emulate terminal connections between BACnet devices.

3. BACnet object model for CCTV systems

This section presents the developed approach that can be used to integrate CCTV systems into BACnet (inter)networks. Suppose a CCTV system consists of one or more smart cameras. These cameras shall be integrated into a BACnet (inter)network where BACnet clients can retrieve event notifications and information about detected events for further processing. Each camera is controlled by a *Camera Application Program (CAP)* which recognizes specified events (e.g., fire or glass break) by using image processing functions. To provide the relevant video data and to notify BACnet clients via the (inter)network in case of a detected event, the device that is interconnected to the camera has to act as a BACnet server. Figure 2 illustrates the interaction between cameras, application programs, and the BACnet (inter)network. Note that currently neither the CAP nor the interface between CAP and the BACnet server are part of the BACnet standard. Figure 2 also shows the object model that is implemented by the BACnet server. The only proprietary object type in this model is called `Camera Event Object Type`. For each smart camera that is connected to the BACnet device, a particular object instance of this type is defined. These `Camera Event Objects` encapsulate all relevant information about the camera events that can be accessed by BACnet clients like an operator or a logging device. Table

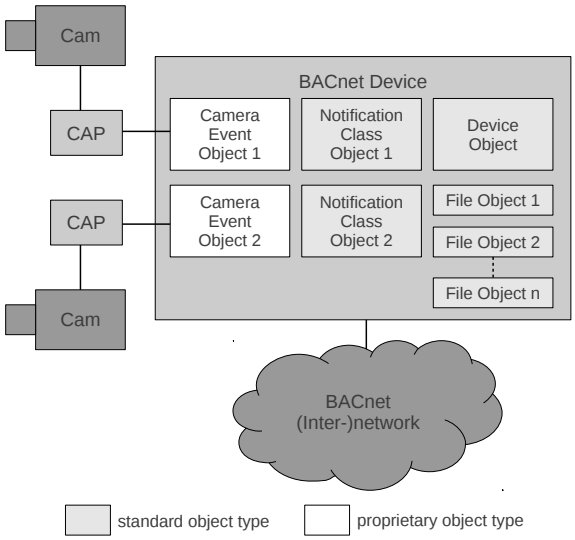

Figure 2. BACnet camera object model

2 lists the properties of this object type.

Property identifier	Property data type
Object_Identifier	BACnetObjectIdentifier
Object_Name	CharacterString
Object_Type	BACnetObjectType
Description	CharacterString
Event_State	BACnetEventState
Event_Type_List	List of BACnetCameraEventType
Event_Deadline_List	List of BACnetDateTime
Event_File_ID_List	List of BACnetObjectIdentifier
Notification_Class	Unsigned
Event_Enable	BACnetEventTransitionBits
Acked_Transitions	BACnetEventTransitionBits
Notify_Type	BACnetNotifyType

Table 2. `Camera Event Object` **properties**

The crucial information about a camera event is the relevant video data, the time of event occurrence, and the type of the event. The video data (e.g., a freeze image or a short video sequence) is stored in a `File Object` (cf. Figure 2) together with a timestamp according to the event occurrence. This means that there exists exactly one `File Object` for every detected camera event. The type of the event is contained in the `Event_Type_List` property of the `Camera Event Object`. This property represents a list of data elements that are of the proprietary data type `BACnetCameraEventType`. This data type represents an enumeration which defines the sort of event (e.g., fire or suspicious motion). The `Event_Deadline_List` property contains a time instant for each camera event after which the event information is assumed to be outdated and therefore will be lost. Thus, the corresponding file object as well as the information about this event within the `Camera Event Object` will be deleted. The `Event_File_ID_List` property contains the object IDs of the corresponding `File Object`. Using these object IDs the camera events are linked to the `File Objects` that contain the video data. Note that the ordering of properties in these three lists is important to distinguish between several events of the same camera. For example, the first event type in the `Event_Type_List`, the first deadline in the `Event_Deadline_List`, and the first object identifier in the `Event_File_ID_List` all belong to the same event. Due to existence of deadlines that are used by different network devices it is necessary to synchronize time between the BACnet servers and the clients. Although a granularity usually in the range of seconds is sufficient for this case, it has to be assured that the clocks do not drift apart too much. If the BACnet (inter)network has access to an NTP-server, NTP can be used to achieve a sufficient synchronization. Otherwise BACnet provides an appropriate service which can be used to distribute the time of the primary server to other servers and the clients.

To inform the BACnet clients about the occurrence of a camera event, intrinsic reporting with `Notification Class Objects` is used (cf. Figure 2). Again there has to be exactly one `Notification Class Object` per camera and per `Camera Event Object`, respectively. This notification class manages the recipient list i.e. a list of BACnet clients that are notified in case of an event. Clients are able to subscribe themselves to the recipient list of each camera they are interested in separately. If a client receives a notification about a new camera event, it has to send back a confirmation message. After having sent the confirmation the client reads the relevant properties of the corresponding `Camera Event Object` from the server (e.g., event type, deadline, and object identifier of the related `File Object`). If the client is interested in the according video data and the deadline of the event is not yet reached, then it may request it by using the file access services of BACnet. In case of a request after the deadline occurred, the client would get an error message because the object identifier of the desired file object does not exist anymore[1]. To avoid that a camera event is deleted before a human operator has been noticed, an additional alarm acknowledgement can be demanded from the operator. Figure 3 illustrates a typical event notification between the server and two clients as timeline diagram.

4. Proof-of-concept implementation

To evaluate the BACnet camera object model and to demonstrate the interworking within the new extended model, an implementation of a proof-of-concept is currently underway. The BACnet server will be implemented on a single-board computer of type *SBC-i.MX51* from Bluetechnix [5] whereas the clients will be implemented on PCs. Furthermore, an ordinary USB Webcam will be used to capture the desired video data. The BACnet network will be based on the BACnet/IP network option.

[1] It has to be assured that the time interval between two camera events which get the same file object id is sufficiently long. In this case sufficiency depends on the rate of event occurrences and on the time span between event occurrence and deadline for deletion, i.e. it is highly application dependent. It must not occur that a BACnet client wants to read a recently outdated event and gets a new one back erroneously.

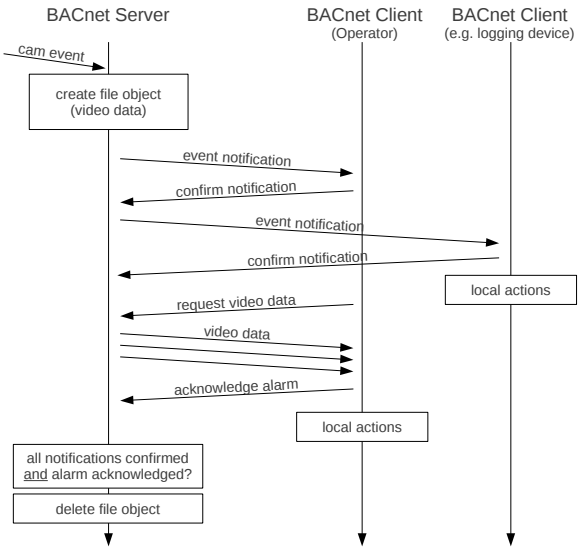

Figure 3. Event notification

On the software side the embedded Linux distribution *Angstrom* [3] is used as operating system for the single-board computer. To detect events and capture images from the Webcam, the *Open Source Computer Vision (OpenCV)* library [9] will be used. Furthermore an open source BACnet protocol stack [4] is used to implement the camera object model and to connect both SBC and PCs to the BACnet network. Figure 4 illustrates the proof-of-concept implementation.

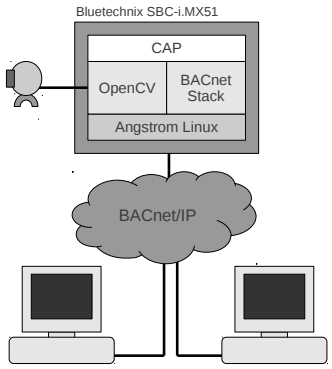

Figure 4. Proof-of-concept implementation

Since the BACnet stack does not implement all services according to the BACnet standard yet, it has to be extended. In particular, this concerns two missing features. First, the BACnet stack does not implement intrinsic reporting – it only supports Change-Of-Value (COV) reporting. For the same reason alarm acknowledgement to ensure that a human operator has been noticed is missing, too. Second, the BACnet stack does not support creation and deletion of objects during runtime. So, it is not possible to create or delete file objects dynamically as it is intended by the proposed model. These missing features will be implemented during the proof-of-concept implementation.

5. Conclusion and work-in-progress

This paper shows the first results of the ongoing work on integrating CCTV systems into BAS. Up to now, an extension to the BACnet object model was specified. Using this extension, all relevant information of specified smart camera events can be modeled in BACnet. Using standard BACnet services the encapsulated information can be propagated to BACnet clients (e.g., an operator workstation or a logging device) that further process the occurrence of detected (critical) events. As a next step, the realization of the intended proof-of-concept implementation as described in the former section is currently underway. This proof-of-concept will be used to show the feasibility of the defined model fully integrated within a CCTV system. Further ongoing work will deal with the integration of CCTV systems into other BAS technologies. While a similar concept was successfully tested for KNX systems [11], extending the application model of ZigBee is also aimed at.

Acknowledgment

This work was funded by FFG (Austrian Research Promotion Agency) under the Kiras project “Networked miniSPOT” P824777.

References

[1] BACnet – A Data Communication Protocol for Building Automation and Control Networks. ANSI/ASHRAE 135, 2008.

[2] Building Automation and Control Systems (BACS) – Part 5: Data Communication Protocol. ISO 16484-5, 2010.

[3] Angstrom. `www.angstrom-distribution.org`.

[4] BACnet Stack. `bacnet.sourceforge.net`.

[5] Bluetechnix GmbH. `www.bluetechnix.com`.

[6] M. Bramberger, A. Doblander, A. Maier, B. Rinner, and H. Schwabach. Distributed embedded smart cameras for surveillance applications. *Computer*, 39(2):68–75, 2006.

[7] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 34(3):334–352, 2004.

[8] W. Kastner, G. Neugschwandtner, S. Soucek, and H. M. Newman. Communication systems for building automation and control. *Proceedings of the IEEE*, 93(6):1178–1203, 2005.

[9] OpenCV. `http://opencv.willowgarage.com`.

[10] B. Rinner and W. Wolf. An introduction to distributed smart cameras. *Proceedings of the IEEE*, 96(10):1565 – 1575, 2008.

[11] F. Schuster, L. Krammer, W. Granzer, and W. Kastner. Integrating surveillance systems into KNX. In *Konnex Scientific Conference*, Nov. 2010.

[12] M. Valera and S. Velastin. Intelligent distributed surveillance systems: a review. *Vision, Image and Signal Processing, IEE Proceedings*, 152(2):192–204, 2005.