Computer-based Monitoring for Decision Support Systems and Disaster Preparedness in Buildings

Alan Vinh
Building Environment Division
Building and Fire Research Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899, U.S.A.

ABSTRACT

The operation of modern buildings can support a vast amount of static and real-time data. Static information such as building schematics is vital for security and rescue purposes. There is a need for building managers and for first responders to be notified of designated building alerts in real-time so that actions can be performed promptly. The capability to monitor building devices and to keep the first responder community updated with the latest building information during emergency situations, as well as the ability to remotely control certain building devices and processes, can be realized today.

This paper describes the various challenges encountered in the research area of building interoperability and proposes methods and insights for developing a standards framework to enable communication between building information systems and first responder information systems. Having a standards framework in place will assist in the development and deployment of commercial products in support of building interoperability.

Keywords: alarm; alert; authentication; authorization; emergency preparedness; information management; interoperability protocols; secure data exchange

INTRODUCTION

The operation of modern buildings can support a vast amount of static and real-time data; sensor devices can be used to monitor pertinent information for daily operation as well as emergency scenarios [1]. To date, there are many sensor systems using proprietary monitoring methods and there is no "standards" framework available to do remote monitoring of different types of buildings and their related sensors. Having a standards framework would allow authorized personnel such as building managers and emergency first responders to remotely monitor alerts and to do limited remote control of building devices or processes. Having standard interfaces to communicate with buildings would encourage software vendors to create products to potentially monitor and control any building for operational and emergency purposes.

The National Institute of Standards and Technology (NIST) is investigating alternative ways for communicating building information such as sensor data, alert data and floor plans to building managers and to first responders' operations centers and mobile units. The Building Information Services and Control System (BISACS) is developed and continues to be enhanced by NIST's Building and Fire Research Laboratory (BFRL) as a prototype standards system that focuses on resolving interoperability challenges in areas such as communication methodology, data security and data integrity, network scalability, data content and encapsulation standards, interface protocol standards, and presentation standards. This paper presents the mechanisms used in prototyping a standards framework for monitoring buildings and for managing various building resources. The challenges encountered, the lessons learned and future challenges in the area of building interoperability with remote systems are also presented.

1. THE BISACS ARCHITECTURAL OVERVIEW

The BISACS consists of a network of servers that monitor entities such as sensor devices and building processes. Software processes or devices such as building sensors send alerts to the BISACS servers, and the BISACS servers propagate the information to various nodes within its network. Client applications monitor only one of these nodes from the BISACS network of servers to obtain their aggregated set of alert information; appropriate personnel can respond to these alerts accordingly [2].

The two main components of the BISACS network are the BISACS Base Server (BBS) and the BISACS Proxy Server (BPS). The BISACS Base Server is the component (a.k.a. the BBS node) that accepts alert notifications from various devices or from Services Interfaces (SIs). The SI is a software component that controls one or more devices belonging to a network of monitored devices. The specifics of how various devices communicate within the network that the SI controls can be hidden; hence the SI behaves as the proxy for its own network to hide any proprietary and extraneous information. The SI takes care of managing all devices that it controls and sends any alerts from those devices to the BBS if configured to do so. The SI also handles all requests and commands destined for devices that it controls via its web services interfaces.

The BISACS Proxy Server is the component (a.k.a. the BPS node) that queries for alert notifications from various BISACS Base Servers or other BISACS Proxy Servers. Since the BISACS network of servers is composed of multiple web servers exposing their web services interfaces, the BPS operates as an active client; it polls for alerts from other nodes that it monitors within the BISACS network of servers. With proper authentication and authorization, this architecture allows for any authorized application or web browsers (with limited capability) to view alert information from any of the nodes within the BISACS network of servers. The BPS resides one or more levels above the BBS in the BISACS network of servers. Figure 1 shows the basic layout of how a BBS and BPS may interact.

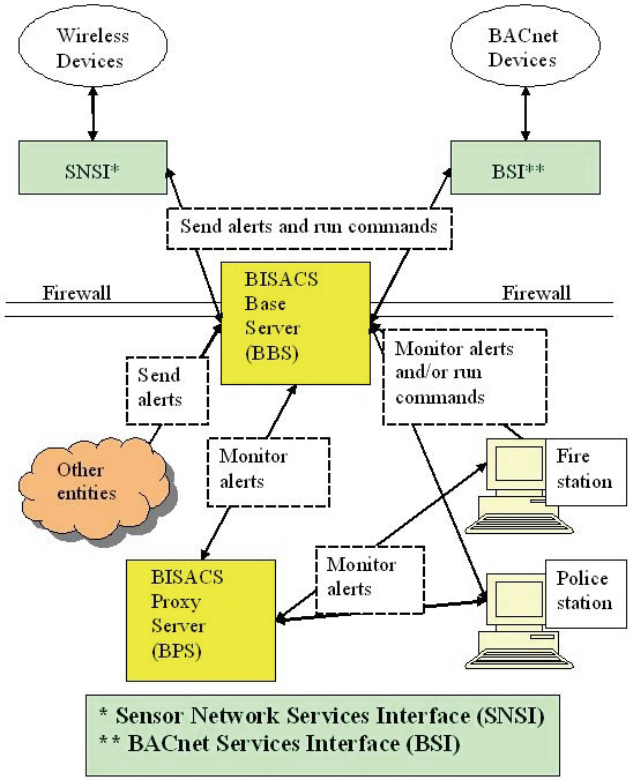

Figure 1 BISACS Architectural Overview

2. COMMUNICATION METHODOLOGY, DATA SECURITY AND INTEGRITY

In order to have a framework for communicating information, the communication methodology, information elements and descriptions, and the network architecture must be in place; the most readily accessible network available today is the Internet and the most prevalent servers available today are the web servers. The BISACS project makes use of the web services technology by implementing its BBS and BPS as web servers using the Services Oriented Architecture (SOA). Communication methodology is done using the secure Hypertext Transfer Protocol (HTTPS) running over the Transport Layer Security (TLS) protocol [3] [4]. Using HTTPS allows the data transmitted over the public Internet to be encrypted and secured while the Transmission Control Protocol (TCP) used by the Internet guarantees data integrity [5].

3. NETWORK SCALABILITY FOR MONITORING ALERTS

Using web services and the SOA, the BPS is designed to poll all of the nodes that it monitors; hence a network of BPS can form a hierarchy of servers to support a wide range of coverage. At the lowest level in the hierarchy lives the BBS. The BBS is the server that monitors one or more buildings for alerts, and it also manages any incoming requests or commands destined for those devices that it controls. All outgoing alerts coming from buildings must first go through a BBS; the BPS sits at least one level above the BBS and it can monitor multiple BBS and/or multiple BPS by polling them for alerts. The chain of alert polling forms a hierarchy where the higher nodes in the network would contain all alerts coming from below. An authorized remote monitoring system can connect to any of the BISACS servers and access all alerts that particular server contains; this network hierarchy maps nicely to cover local buildings, then up to city coverage, then to county coverage, then to state coverage and finally to nationwide coverage. Figure 2 shows an example of how the BISACS network of servers can form a hierarchy of alert information so that at the lowest level are the BBS nodes that monitor one or more buildings; first responder monitoring systems such as emergency communication centers and police stations may reside in the middle and communicate with one of the mid level BPS; at the highest level is a nationwide monitoring system such as the Federal Emergency Management Agency that monitors one of the top level BPS in the hierarchy of servers. Building support and management systems can communicate with their local BBS to monitor their network of sensors to enhance their decision making processes.

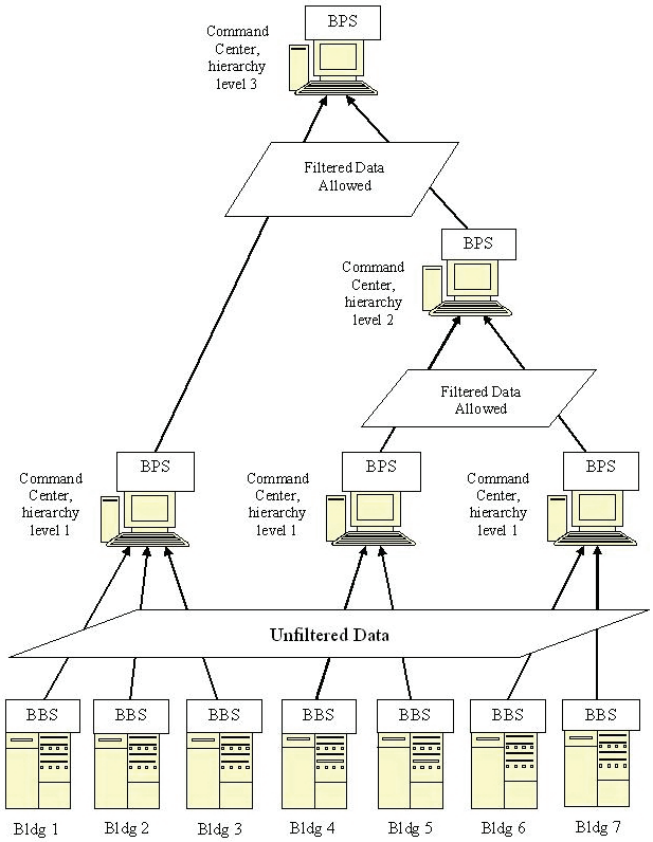

Figure 2 BISACS Network Hierarchy

4. DATA CONTENT AND ENCAPSULATION STANDARDS

What are alerts, and how are they represented? In the context of this research, alerts are indicators such as signals sent from sensors to their controllers. The indicators can be normal status information or alarm notifications such as the temperature of a room during normal conditions versus under fire conditions.

In the BISACS architecture, the Services Interface (SI) monitors its network of sensors and forwards alerts to the BBS; one BBS can manage multiple buildings and multiple SI such as shown with the green boxes in Figure 1. The SI converts sensor signals to Common Alerting Protocol (CAP) messages [6] and uses XML [7] to encapsulate the data. The Organization for the Advancement of Structured Information Standards (OASIS) developed the Common Alerting Protocol as a standard in 2005; the CAP is "a simple but general format for exchanging all-hazard emergency alerts and public warnings over all kinds of networks". There are various kinds of information coming from numerous building devices that can be sent to the BBS, the CAP is the data encapsulation standard that the SI uses to encapsulate its alert information. Figure 3 shows a sample CAP message that is sent between the SI and the BBS. When the BPS polls the BBS, alerts are sent from the BBS to the BPS by using CAP messages.

```
<?xml version="1.0" encoding="UTF-8"?>
<alert xmlns="urn:oasis:names:tc:emergency:cap:1.1">
    <identifier>1179353147004</identifier>
    <sender>https://p623572.campus.nist.gov:8443/bisacs</sender>
    <sent>2007-12-16T18:05:47-04:00</sent>
    <status>Exercise</status>
    <msgType>Alert</msgType>
    <source>alarm1bundle.sensor01</source>
    <scope>Public</scope>
    <info>
        <category>Fire</category>
        <category>Safety</category>
        <category>Security</category>
        <event>Smoke</event>
        <urgency>Immediate</urgency>
        <severity>Extreme</severity>
        <certainty>Observed</certainty>
        <expires>2007-12-16T18:06:47-04:00</expires>
        <description>Smoke detector, building 226, 3rd floor, room B346.
        </description>
    </info>
</alert>
```

Figure 3 Sample Common Alerting Protocol Message

What is learned by using the CAP? In-house user comprehension tests have shown that the amount of alert information available from the BBS can be overwhelming. The CAP needs to support the ability to filter on alerts of interest, so that users are not overwhelmed by the number of alerts that can be displayed. BFRL is developing recommendations for improvements in the CAP to be submitted to the OASIS. These recommendations will involve creating one or more elements in the CAP message structure to support specific event types for filtering purposes.

5. INTERFACE PROTOCOL STANDARDS

How do applications communicate with the BISACS? The BBS and BPS are implemented as web servers hence web services interfaces are used for communication purposes. For the BISACS servers, a specific set of web services interfaces is created to support outgoing data as well as incoming requests and commands [2]. All BISACS web services interfaces are documented using the Web Services Description Language (a.k.a. "the WSDL") [8] and are described in the NISTIR 7466 [2]. The communication between the BBS and the Services Interface is also done via web services interfaces (see Figure 1). The SI is the proxy for communicating with building resources and the BACnet Web Services standard (BWS) [9] is used for the SI web services interfaces. The BWS is designed to support communication with any building and the SI is designed to support any network of devices so they are well matched. Research is being done by BFRL to determine the exact information that will be passed through the BWS interface in order to support all types of devices. The WSDL for the BWS is available from the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) [9].

Outgoing building alerts from the SI destined for the BBS are encapsulated using the CAP standard so that all applications that support the CAP standard will be able to process the alert information contained in the BBS or the BPS. The current implementation of web services interfaces

for the BISACS servers are generic in design to support any type of text information [2]. This design supports future research such as commands and requests that are destined for the SI from remote monitor and/or control systems. These remote systems communicate with the protected SI by using the BBS as the proxy.

6. AUTHENTICATION AND AUTHORIZATION

How is access security supported in the BISACS architecture? Building information such as alerts can be sensitive data, so measures are taken to ensure that only authorized access is allowed to the BISACS servers. HTTPS is used to encrypt the data channels, but this does not keep out unauthorized users. The access mechanism implemented for the BISACS involves multiple levels of security using user IDs and passwords along with certificates of authenticity [10], Personal Identify Verification (PIV) cards [11] [12] and encrypted security keys [2]. The use of PIV cards along with certificates of authenticity allows only recognized terminals or devices to access the BISACS servers. A user must have a PIV card to access a BISACS terminal device (TD).

Once access is allowed to a TD, the TD becomes an access point on the BISACS network of servers. The TD first contacts the certificate validator server to validate its own certificate of authenticity, i.e., the TD's certificate of authenticity public signature must exist in the BISACS database. Once the certificate of authenticity is validated, an encrypted certificate key is obtained from the certificate validator server. The encrypted certificate key, the user ID and password is then checked to authenticate the user and to give the user proper access control to the system. Based on the "Access Control in BACnet" document [13], the process to obtain the proper access control is referred to as the "authorization process" [2] and this process is still being researched and defined. Once the user is properly authenticated and authorized, then a session key is returned to the Application Specific Client (ASC) so that the session key can be used with all of its commands and requests, i.e., all commands and requests destined for a BISACS server that does not have a valid session key will not be serviced. Session keys become invalid after a configurable time period if the client application is inactive. Once the session key has expired, the authentication and authorization process must be repeated in order to obtain a new session key. Although the mechanisms used for accessing the BISACS servers are described above, the ASC mentioned above is currently under development.

7. PRESENTATION INTERFACE STANDARDS

How will the alert information be displayed? Currently, a web client is available to view alerts using any of the nodes within the BISACS network of servers [2]. The web client only checks the user ID and password; hence it is not a very secure interface. Figure 4 shows the login screen for the web client and Figure 5 shows the web client's "Alert Status Screen" containing some alerts.

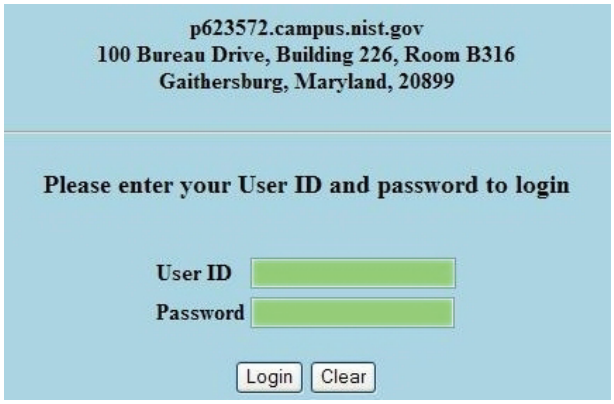

Figure 4 Login Screen

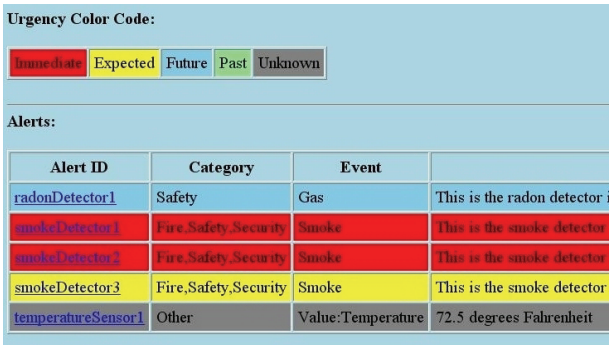

Figure 5 Alert Status Screen

The amount of alerts sent to the user screen can be overwhelming; filter mechanisms must be used in order to display only alerts of interest. Figure 6 shows the web client's ability to filter on alert categories, event text and other alert attributes in order to focus only on alerts of interest.

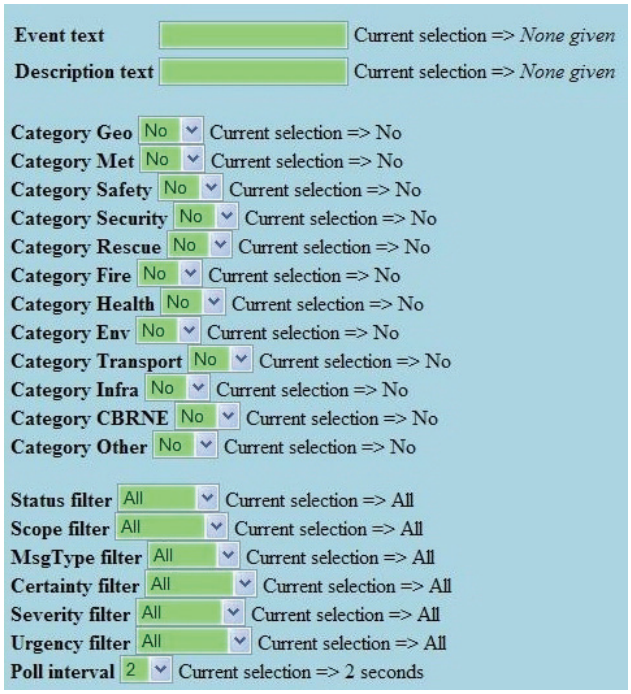

Figure 6 Alert Filter Screen

Figure 7 shows the web client's ability to drill down to specific alert information.

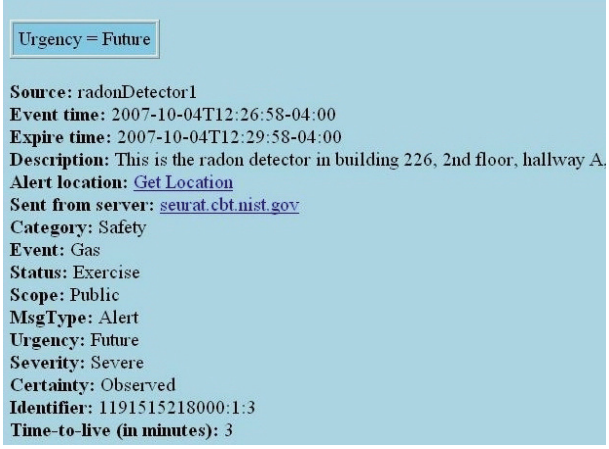

Figure 7 Alert Information Screen

Figure 8 shows the web client's ability to display a floor plan with the location of a sensor device by using simple JPEG images [15]. More elaborate interactive floor plan display mechanisms are still being researched.

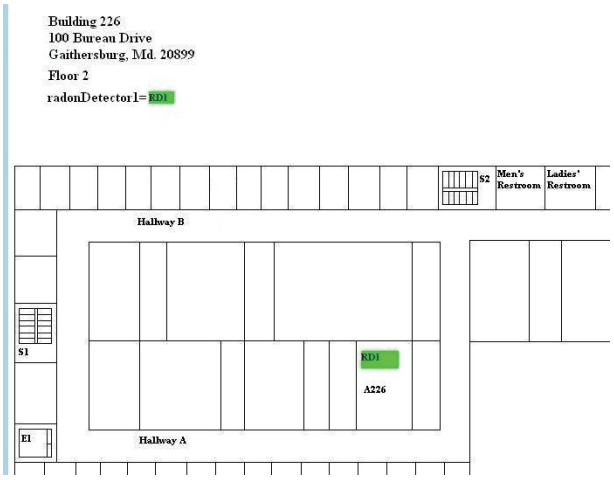

Figure 8 Alert Location Screen

The prototype for the more secured application specific client (ASC) that displays alert notifications, gives access to building floor plans as well as access to querying and controlling various building devices and processes is currently under development. The ASC will make use of encrypted data channels, certificates of authenticity, PIV cards and security keys for interacting with the BISACS servers.

The National Electrical Manufacturers Association (NEMA) is developing a standards document (SB30) for the emergency first responder user interface. The SB30 document is included in "NFPA 72, National Fire Alarm Code" [14] and describes in detail how the user interface should look and behave. For prototyping purposes, the ASC will base its user interface on the description provided by the SB30 document. Figure 9 shows a template of what this user interface may look like.

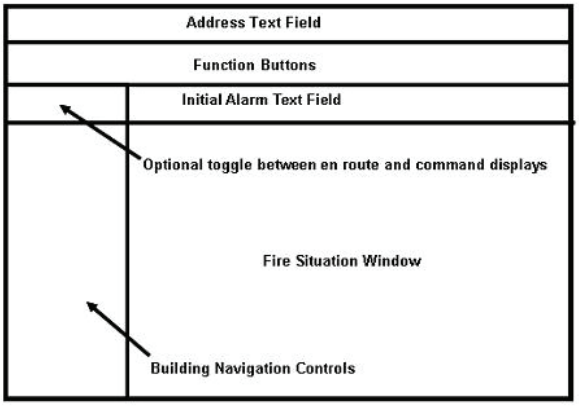

Figure 9 SB30 User Interface Screen

How do alerts get mapped to the screen? Currently, the SB30 document describes the screen layout for emergency responders but there is no technical description for how CAP messages are mapped to the user screen. BFRL is researching the requirements needed to map CAP messages to icons on an interactive floor plan of a building. This research will most likely result in changes to the SB30 document as well as the OASIS CAP standard.

Using authorized access mechanisms, building decision support systems can also make use of the BBS to monitor devices and processes to support management decisions where appropriate. All alerts destined for emergency first responder systems can also be monitored by building decision support systems. The research and tests for controlling building devices via the BISACS is currently being conducted and will be realized in the near future.

8. SUMMARY AND CONCLUSIONS

Modern buildings can offer a wealth of information to support daily operations and to support emergency scenarios. To date, there is no standards system for doing remote monitoring and controlling of buildings. BFRL has developed the BISACS to prototype a standardized system and has demonstrated solutions for overcoming various interoperability challenges. Solutions for communication methodology, data security and data integrity, network scalability, data content and encapsulation standards, interface protocol standards, and presentation standard have been presented in this paper. Using a web client, BFRL has demonstrated how alerts can be initiated by sensor devices, go through a hierarchy of servers and end up on the user terminal to support emergency first responders.

The amount of alerts presented to a user can be overwhelming; using a web client, BFRL has demonstrated how a filtering mechanism can be used to display only alerts of interest. The lesson learned from using the CAP standard is that an addition of one or more elements is required in the CAP message to support the filtering of

specific alert event types. The alert filtering capability allows for better management of the information presented to the user.

BFRL plans to have the Services Interface use the BACnet Web Services standard (BWS) for communicating with its devices, more research is required to specify the details for the information that will go through the BWS to communicate with all types of devices, i.e., all devices that may or may not be BACnet compatible. Other challenges such as obtaining and translating alerts from existing buildings' fire panels into CAP messages are still unresolved.

The secured Application Specific Client being developed by BFRL will reference NEMA's SB30 document to implement the user interface. The SB30 document currently does not give details for how to map an alert to a floor plan on the emergency first responder screen; BFRL is currently conducting research to help standardize a mechanism for mapping alerts to the SB30 user screen. The result of this research will most likely propose changes to the SB30 document and to the OASIS Common Alerting Protocol standard.

The BISACS prototype provides insights on the research and standards issues that must be addressed in order to enable the development and deployment of commercial products that will allow the interoperability between modern buildings and remote monitoring systems such as emergency first responder information systems. As interoperability challenges get resolved, the possibility of having a standards framework for integrating buildings with remote information systems for protecting properties and occupants better can be realized.

9. DISCLAIMER AND COPYRIGHT NOTICE

Certain trade names, documents or organizations are mentioned in this paper to specify adequately the resources used for developing and supporting the Building Information Services and Control System (BISACS). In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the resources used are the best available for the purpose.

10. REFERENCES

[1] Holmberg, David G., Davis, William D., Treado, Stephen J., Reed, Kent A., **Building Tactical Information System for Public Safety Officials, Intelligent Building Response (iBR), NIST Internal Report 7314**, January, 2006.

[2] Vinh, Alan B., **Building Information Services and Control System (BISACS): Technical Documentation, Revision 1.0, NIST Internal Report 7466**, September, 2007.

[3] **HTTP Over TLS**, available at http://tools.ietf.org/html/rfc2818, May, 2000.

[4] **The Transport Layer Security (TLS) Protocol Version 1.1**, available at http://tools.ietf.org/html/rfc4346, April, 2006.

[5] **Transmission Control Protocol, DARPA Internet Program, Protocol Specification**, available at ftp://ftp.rfc-editor.org/in-notes/rfc793.txt, September, 1981.

[6] **Common Alerting Protocol, v. 1.1, OASIS Standard CAP-V1.1**, available at http://www.oasis-open.org/committees/download.php/15135/emergency-CAPv1.1-Corrected_DOM.pdf, October, 2005.

[7] **Extensible Markup Language (XML) 1.0 (Fourth Edition)**, available at http://www.w3.org/TR/xml/, September, 2006.

[8] **Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language**, available at http://www.w3.org/TR/wsdl20/, June, 2007.

[9] **ASHRAE, ANSI/ASHRAE Standard 135-2004, Addendum C**, BACnet - A Data Communication Protocol for Building Automation and Control Networks, American Society of Heating, Refrigerating and Air-Conditioning Engineers, October, 2006.

[10] **Web Services Security, X.509 Certificate Token Profile, OASIS Standard 200401**, available at http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0.pdf, March, 2004.

[11] Dray, J., Guthery, S., Schwarzhoff, T., **NIST Special Publication 800-73-1 Interfaces for Personal Identity Verification**, National Institute of Standards and Technology, Gaithersburg, MD 20899, March, 2006.

[12] **FIPS 201-1, Personal Identity Verification (PIV) of Federal Employees and Contractors**, National Institute of Standards and Technology, Gaithersburg, MD 20899, March, 2006.

[13] Ritter, D., Mundt, H., Isler, B., Treado, S., **Access Control in BACnet, ASHRAE Journal Article**, American Society of Heating, Refrigerating and Air-Conditioning Engineers, Atlanta, GA, 2006.

[14] **NFPA 72, National Fire Alarm Code**, available at http://www.nfpa.org/freecodes/free_access_agreement.asp?id=7207, August, 2006.

[15] **ISO/IEC 10918-1: Information technology – Digital compression and coding of continuous-stone still images: Requirements and guidelines**, available at http://www.w3.org/Graphics/JPEG/itu-t81.pdf, September, 1992.