A Test Shell for Developing Automated Commissioning Tools for BACnet Systems

Natascha S. Castro, Michael A. Galler, Steven T. Bushby
National Institute of Standards & Technology

Synopsis

Despite advances in building automation technology, many building control systems do not work as intended. Thorough commissioning of control systems can significantly improve operation but is manually intensive and thus expensive and often neglected. Research in automated commissioning tools and processes has the potential to significantly improve this situation. However, it can be difficult to conduct this research in actual buildings because of the need to maintain comfortable and safe conditions for the building occupants.

This report describes an enabling tool designed to advance research efforts to develop and test automated commissioning tools for HVAC systems. The Commissioning Test Shell enables the side-by-side testing and comparison of two or more commissioning tools, as well as the integration of information from multiple tools. The Commissioning Test Shell interacts with building controllers through a BACnet communication interface [1]. It provides an Open Database Connectivity (ODBC) interface for commissioning tools not specifically designed to use the BACnet communication protocol. This permits researchers to develop commissioning tools without the added complexity of communication with the building control system. The Commissioning Test Shell can provide data from the control system to passive commissioning tools. It also supports a scripting capability that can be used by the commissioning tool to actively interact with the control system. The Commissioning Test Shell has been linked to the Virtual Cybernetic Building Testbed (VCBT), which is a whole-building emulator that combines simulations of a building and the mechanical system with real state-of-the-art BACnet speaking control systems to provide a hybrid software/hardware testbed.

This paper describes the architecture and capabilities of the Commissioning Test Shell. General opportunities to automate the commissioning of air-handling units (AHUs) are discussed. Results from preliminary tests of the Commissioning Test Shell to verify that scripting capabilities and data requirements are met for real-time, active testing of typical air-handling units (AHU) are presented and plans for the future use of the testing tool are described.

About the Authors

Natascha S. Castro is a mechanical engineer in the Mechanical Systems & Controls Group at NIST and leads the US Team of IEA Annex 40.

Michael A. Galler is a mechanical engineer in the Mechanical Systems & Controls Group at NIST.

Steven T. Bushby is leader of the Mechanical Systems and Controls Group at NIST and chair of the ASHRAE committee that maintains the BACnet standard.

Introduction

Buildings seldom operate at their intended design performance levels. Although this is attributed to a variety of causes, occupant comfort and the energy performance of buildings are clearly coupled to the way buildings are commissioned. Several studies have reported significant energy savings obtained through the retro-commissioning of existing buildings. In 1994, Claridge [2] showed up to 20 % energy savings on a recommissioned building while Haasl et al. [3] showed savings that ranged from 2 % to 17.6 % based on the scope of the commissioning services provided and had a short-term (1 year or less) payback in all cases. The primary obstacle to making commissioning a routine process for all buildings is the perceived cost. For this reason, tools for automating commissioning are being developed to decrease the cost of delivering commissioning and/or demonstrate the benefit obtained by performing commissioning.

Automated commissioning is a feature that building control system manufacturers and service providers are likely to offer in the near future; however, a considerable amount of development and testing of commissioning tools remains to be performed before this happens. In an effort to bridge this gap, the Energy Conservation in Building and Community Systems program of the International Energy Agency (IEA) has sponsored Annex 40 [4], a research project aimed at developing, validating, and documenting tools for commissioning buildings and building services. In addition to establishing guidelines on commissioning procedures and recommendations for improving the commissioning process, the Annex is developing prototype software that can be implemented in stand-alone tools or embedded in energy management and control systems.

One of the challenges to be overcome by an automated commissioning tool is obtaining the data needed from the building control system and, for active commissioning tools, commanding the control system into specific modes of operation. The use of a standard communication protocol such as BACnet[1] greatly simplifies this problem because the commissioning tool can be used with controllers from any manufacturer that supports the communication standard. Another challenge is that it is difficult to test prototype commissioning tools in real buildings without interfering with the operation of the building and potentially inconveniencing building occupants.

As a contribution to IEA Annex 40, NIST has developed a new tool called the Commissioning Test Shell that is being used by researchers to test automated commissioning approaches. The Commissioning Test Shell communicates with BACnet controllers to extract data needed by the commissioning tool. It also supports a scripting capability that can be used by the commissioning tool to command the control system into specific modes of operation. The commissioning tool interfaces to the Commissioning Test Shell through an Open Database Connectivity (ODBC)[2] interface. This provides a way for the researchers to focus on the commissioning process and avoid the data communication complexities of communicating directly with the controllers.

[1] BACnet is a registered trademark of the American Society of Heating, Refrigerating, and Air-Conditioning Engineers, Inc. Use of trademark names does not imply recommendation of any commercial products by the National Institute of Standards and Technology.

[2] ODBC is a Microsoft standard. Reference to this does not imply recommendation of any commercial products by the National Institute of Standards and Technology.

In principle, the Commissioning Test Shell could be coupled with any BACnet control system. For the Annex 40 research, the Test Shell has been coupled to the NIST Virtual Cybernetic Building Testbed (VCBT)[5]. The VCBT is a whole-building emulator that combines simulations of a building and the mechanical system with real state-of-the-art BACnet-speaking control systems to provide a hybrid software/hardware testbed. The overall effect is that the controllers see data that look like sensor information from real building systems and the simulations respond to the control actions taken by the controllers. Just as a flight simulator simulates an airplane in real time, the VCBT simulates a building, the weather, the Heating, Ventilation, and Air Conditioning (HVAC) system, and the heating/cooling plant in real time.

The VCBT has the capability to simulate various kinds of mechanical system faults and to introduce control logic and tuning errors. This provides a realistic approximation of real building systems but without the difficulties associated with working in occupied buildings. Unlike a real building, the conditions in the VCBT can be carefully controlled and reproduced. The combination of the Commissioning Test Shell and the VCBT provide an excellent environment for developing and testing commissioning tools.

Application

In the past, automated diagnostic tools have been largely limited to passive testing. These types of tools provide valuable information during the commissioning process, but do not have as great a potential to reduce the manual labor associated with commissioning. The purpose of the Commissioning Test Shell was to enable users to test their commissioning tools through active testing, providing the sensor and setpoint information needed for the system analysis. The testing requirements listed below can generally be applied to any system.

What needs to be tested?

The Building Energy Management System (BEMS)

It is critical to establish that the BEMS is operating as intended before it can be used to commission any system. This includes both hardware and software component checks such as inspecting the wiring, verifying communications and point mapping, calibrating sensors, verifying control of actuators, verifying general control logic, and checking the setup of field control panels and operator workstations.

Systems and Equipment Inspection

All systems and equipment must meet general safety and functional requirements.

Sensor Calibration- Temperature, Pressure, Flow Rate, etc.

All sensors must be in good working order and calibrated. Specific accuracy requirements must be met in order to use the sensors for performance tests. In addition, the sensor placement must be matched with the design intent and mapped to the correct point in the BEMS.

Component Tests

Capacity and other performance compliance of the individual components must be verified. The specific procedures for testing the components depend on the design characteristics of the system.

Control System Tests

Functional compliance of the control system component tests are based on steady state operating conditions. This allows the verification of the components' peak capacities.

Two useful references are the Commissioning Test Protocol [6], which provides a collection of functional test procedures along with information designed to assist in customizing tests for specific projects, and the HVAC Functional Testing Guide, which presents a more general synthesis of inspection procedures for a broader range of equipment [7].

What capabilities are needed to automate functional tests of the components and control system?

Automated commissioning using the BEMS requires confidence in the BEMS operation, sensor readings, and basic operation of the control system before it can reliably be used to commission any specific components or system interactions. Commissioning of the BEMS overlaps considerably with the commissioning of those systems with which it interfaces, therefore it can provide valuable information for the commissioning of components. The aim of automation is to reduce the amount of manual effort required at the test site and the cost of the commissioning process. For example, because of the high number of Variable Air Volume (VAV) boxes and the amount of labor involved, it is common that only a small fraction of boxes are selected for testing, which presents a good opportunity for automation, particularly when multiple components are tested in parallel.

The Commissioning Test Shell Architecture

The Commissioning Test Shell is a tool designed to facilitate the development and testing of automated commissioning tools. It does this by providing BACnet communication with the building controllers that can be used to obtain data and to command specific modes of operation. The commands are handled through a simple scripting mechanism. The data is made available to the Commissioning tool being tested through an ODBC interface. By significantly reducing the communication complexities, the Commissioning Test Shell provides a way for the researchers to focus on the commissioning process and the analysis of the data.

In order for the Commissioning Test Shell to be effective, it must provide all the data that the prototype commissioning tool needs to identify issues with components and control logic. These data, such as setpoints, sensor readings, and occupancy information, must be available in real time. In addition, the tool must be capable of executing a sequence of commands that have been preprogrammed, or entered live by the operator.

The main components of the Commissioning Test Shell are the BACnet Data Source (BDS), the database program, and the associated ODBC enabling links. A schematic of their interaction is shown in Figure A. The BDS is designed to communicate with BACnet controllers by reading and writing the properties of BACnet objects. The BDS can communicate with any BACnet control system, but in the application discussed in this paper, it is linked to the Virtual Cybernetic Building Testbed (VCBT) which uses real BACnet controllers coupled with computer simulations

to emulate an entire three-story building. The BDS retrieves data from the controllers and stores the information retrieved in an ODBC enabled database program.

The ODBC-enabled database used in this test shell was developed by Natural Resources Canada (NRCan) as a part of their Diagnostic Agent for Building Operators (DABO) [8]. The database portion is a key component of the Commissioning test shell because it provides a convenient user interface to identify the measurement data and setpoint information acquired from building controllers that can be stored for analysis. This database functions to make data available any Commissioning tools connected through an ODBC connection.

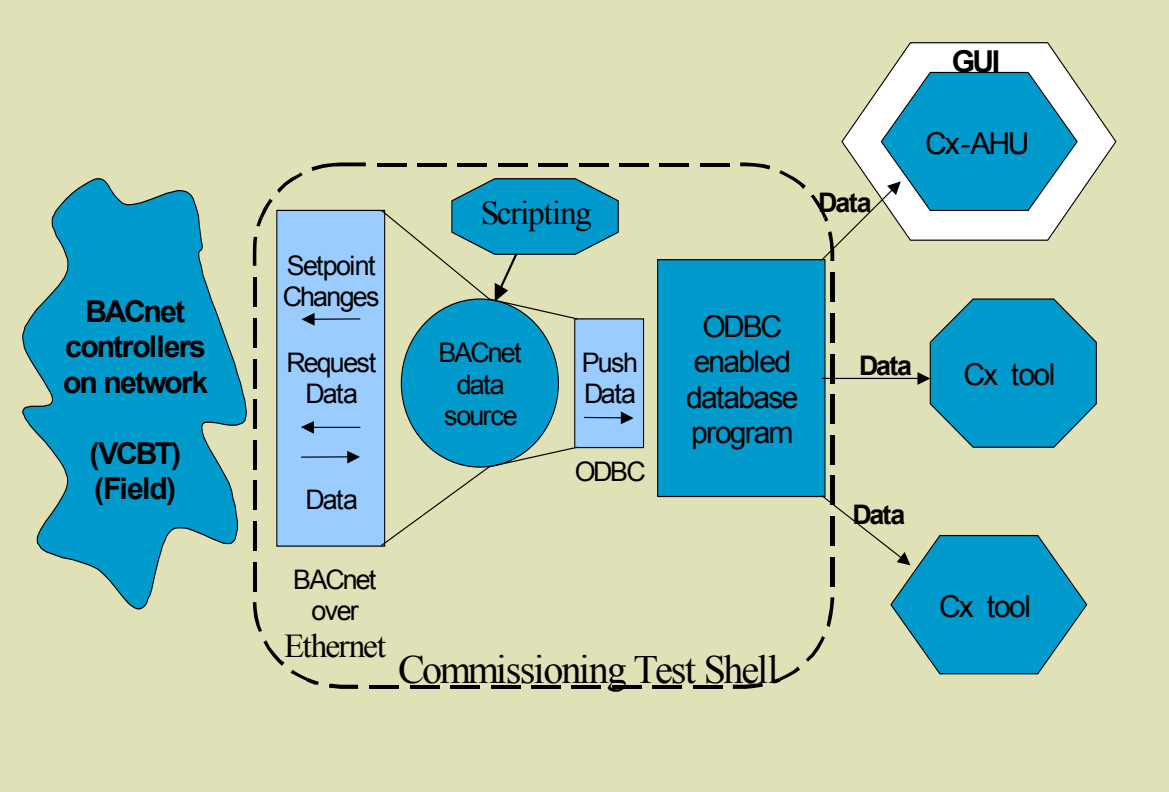

Figure A: Schematic of the Commissioning Test Shell and VCBT

Description of the BDS

The BDS works by sending BACnet ReadPropertyMultiple requests to controllers. The requests currently can be for the Present_Value property of Analog Input, Analog Output, Analog Value, Binary Input, Binary Output, and Binary Value objects, or for the Local_Date or Local_Time properties of a Device object. This combination of BACnet objects and properties provides for the kind of sensor information and control of outputs, such as dampers and valves, that are typically important in commissioning applications. Once the data are obtained from the BACnet controllers, the BDS can write the data to one or more output files for later use, store it in a database with an ODBC interface, or link it to other applications using a ODBC interface. The BDS is also capable

of executing scripts for the purposes of actively controlling the commissioning process. These scripts can be used to command particular modes of operation and other activities described in more detail below. The BDS works in a cycle of retrieving data, analyzing any scripts, and waiting until the next time to retrieve data. The cycle time is one minute by default, but can be changed by the user.

Configuring the BDS

The first step in using the BDS is to determine the list of data points that will be monitored. This is typically dictated by the requirements of the commissioning tool, although additional data points that are not used by the commissioning tool may be monitored to assist in expert analysis. There may be up to a total of 400 data points, selected from any combination of controllers in the BACnet system.

There are two format files used by the commissioning tool, one containing BACnet device and data point information, and the second describing how scheduling events are formatted and described. The first is a configuration "Settings" file, which can be modified using a text editor. The Settings file has two sections, the first of which is used to define the data points that are to be monitored and to link them with the BACnet objects and properties that represent them. The optional second section describes the files into which data from the controllers are saved. An example settings file is shown in Table 1.

Table 1: Sample Settings File

Command	Description
DEV 21 04 110 00602D000820 AHU1	Create entry for BACnet device
POINT AO 0 1 AO0	Describe BACnet Analog Output object 0
POINT AO 1 1 AO1	Describe BACnet Analog Output object 1
POINT AO 3 1 AO3	Describe BACnet Analog Output object 3
POINT AI 2 1 AI2	Describe BACnet Analog Input object 2
POINT AV 3 1 AV3	Describe BACnet Analog Value object 3
POINT AI 9 1 AI9	Describe BACnet Analog Input object 9
ALIAS AO3 OSAdamp	Create alias to BACnet object
ALIAS AV3 SATsp	Create alias to BACnet object
ALIAS AI2 OAtmp	Create alias to BACnet object
ALIAS AO0 Hcoil	Create alias to BACnet object
ALIAS AO1 Ccoil	Create alias to BACnet object
ALIAS AI9 SAtmp	Create alias to BACnet object
VAR myvar 32	Create a variable, set value to 32
TIMER mytimer	Create named timer
END	End of section
DIR c:\data	Change default directory for output files
FILE OSAdamp100.dat 15	Save data to this file every 15 minutes
END	

BACnet controllers and objects are described in the format file using keywords. Each description is on a separate line of the file, beginning with a keyword that determines the type of object being described. The keywords are DEV, POINT, VAR, ALIAS, and TIMER. The DEV keyword is used for BACnet device descriptions. The POINT keyword is used to describe BACnet objects,

such as an Analog Input object residing on a device. POINT descriptions are associated with the preceding DEV description. VAR is used to create a variable, which may be assigned an initial value, and may have its value changed during scripting execution. ALIAS is used to create another name for a POINT, which may be useful for scripting. TIMER creates a named timer, the value of which may also be used for comparisons in scripting.

This file defines one device, with six data points. There is also one variable, six aliases, and one timer defined. Note that the variables and timers may be defined anywhere in the first section. The directory to which the data files will be saved is set to c:\data\, and one file is defined.

The second section of the Settings file describes the files into which data from the controllers is saved. The data collected by the BDS may be saved in up to eight files, with any subset of the data going to each file. If data is not going to be saved to any files (i.e., it will only be saved via a Dynamic Data Exchange (DDE) or ODBC connection) then this section of the Settings file may be omitted.

BDS Scripting

The next step is to determine the commissioning requirements, and to transform them into a form the BDS can understand and execute. This is the content of the second file, or the "Scripting" file. Scripting is implemented by defining named events, with each event consisting of a list of requirements to "succeed" and a list of actions to take when the requirements are met. The requirements consist of comparisons of values from data points, variables, timers, constants, or the success of another event. Actions specified upon success are setting a data point or variable to a new value, or restarting a timer. There are two classes of events that may be declared, cyclic and non-cyclic. A non-cyclic event will be tested only until it succeeds once, and will be marked as passed for the duration of the test. A cyclic event will be retested each cycle, and marked as not passed at the beginning of each test cycle even if it passed the previous cycle.

Event requirements are a comparison between data point values or variables, a comparison between a timer value and another value, or a check on the status of other events. For data point values, the last value received is used. For variables, the last value assigned to the variable is used. For timers, the interval between when it was set and the current time is used. For event status, the requirement is satisfied if the event named has succeeded. Comparison operators that may be used are: less than (LT), equal (EQ), greater than (GT), less than or equal (LTE), greater than or equal (GTE), or steady state (SS). A requirement is defined by using the keyword REQ followed by the desired requirement.

Event setting statements are implemented when all of the requirements for that event are satisfied. The setting may result in a new value being applied to a BACnet object, to a variable, or a timer may be reset. The value that is applied may be a constant, or the current value of another BACnet object or of a variable. BACnet objects and variables may be accessed directly, or through an ALIAS. Example scripting files are shown in Table 2 and Table 3.

Preliminary Tests

To test the Commissioning Test Shell, sample functional tests were developed. The need was to verify that the communications and data transfer worked between each component in Figure A. For preliminary tests a typical air-handling unit with VAV box configurations was selected for functional testing. The functional test script was developed from expert knowledge of how the system is intended to operate along with recommended steps outlined in test guides [6] [7]. Two types of tests selected for automated are:

Steady State Tests

Steady state conditions are used primarily to verify the peak capacity of components by checking that the changes in sensor readings are appropriate under the prescribed test conditions. An illustrative sample script used to test the damper operation one of the air-handling units within the VCBT is presented below. The test outlined below is a simple test of the damper function that can be automated. For large systems, this type of operation could take considerable longer without the use of the BEMS. By taking advantage of remote sensor readings, remote control of dampers, and the capability to run tests in parallel to automate this test, a commissioning tool can save the commissioning agent valuable time. The script shown in Table 2 below uses objects defined in the sample settings file in Table 1.

Table 2: Sample Script File Demonstrating an Operational Test of a Damper

Command	Description
EVENT OSAdamp100.0	Names the first step in the functional test of the outside supply air damper 100 as an event
REQ START GT 5	Establishes a time delay at 5 min
SET OSAdamp 15	Sets outside air damper to 15 % open
SET mytimer	Resets timer for test duration
END	Ends the first step
EVENT OSAdamp100.1	Names the second step in the functional test of the outside supply air damper 100 as an event
REQ mytimer GT 30	Establishes the time delay at 30 min
REQ OSAdamp100.0	Requires that the first step of the functional test be completed
SET OSAdamp 100	Sets outside air damper to 100 % open
END	Ends the second step

When the test script is run, the operator is able to check the system response. The test script can be configured with a staged step test designed to capture the switchover temperature and demonstrate the response of the economizer cycle. All of the measurements specified in the settings file are read recorded in the database at the prescribed intervals for analysis using a commissioning tool.

Operational Tests

This test is used to verify that the control logic has been implemented correctly. An illustrative sample test script used to test of an enthalpy-based economizer in one of the air-handling units

within the VCBT is presented below. By changing the test conditions, the dynamics of the economizer function to make use of free cooling when possible, and the switch to mechanical cooling can be witnessed. It is necessary to test under varied seasonal conditions because improper operation can lead to significant energy waste and potential indoor air quality problems. This script shown in Table 3 below uses objects defined in the sample settings file in Table 1.

Table 3: Sample Script File Demonstrating a Steady State Test of a Cooling Coil

Command	Description
EVENT CCtest1.0	Names the first step in the functional test of the cooling coil as an event
REQ START GT 5	Establishes a time delay at 5 min
SET OAtmp 30	Sets outside air temperature to 30 C
SET SATsp 20	Sets Supply Air Setpoint to 20 C
SET mytimer	Resets timer for test duration
END	Ends the first step
EVENT CCtest1.1	Names the second step in the functional test of the cooling coil as an event
REQ mytimer GT 120	Establishes the time delay at 2 h (120 min) after completion of first test
SET OAtmp 35	Sets outside air temperature to 35 C
END	Ends the second step

Sensor readings provide valuable information about the operation of the system. It is envisioned that once set up, in the initial commissioning, these scripts could be used on a regular basis to ensure the continued operation of the system as part of an on-going commissioning process.

Summary & Future Work

Building energy consumption can be dramatically improved through the practice of commissioning. Automated commissioning may hold the key to lowering the cost of commissioning by reducing the manual labor associated with commissioning and enabling the simultaneous testing of multiple components. The Commissioning Test Shell was designed as an aid in the development and cross comparison of automated commissioning tools. The communications environment and scripting capabilities were developed and tested at NIST. The results verify the functionality of the scripting and communications features of the test shell for a simple air-handling unit.

The communications portion of this tool is currently being tested/implemented by NRCan and NIST. The next step in this research is to link a commissioning tool to the test shell and test its performance using the VCBT. Various automated commissioning tools are under development within the Annex 40 research project. NIST is collaborating with the Scientific and Technical Building Centre (CSTB) in Paris, France on the development of a prototype automated commissioning algorithm of an air-handling unit using a rule-based method.

The Commissioning Test Shell currently provides a platform for the cross comparison of any ODBC speaking tool. It is a collection of software and hardware components that cannot be readily moved outside of the laboratory setting, thereby requiring that the tools being tested be connected at the NIST site. Future work includes adding the capability of a complete web-access. The goal is to set up communication with a tool via the Internet that would enable active testing using the VCBT.

Acknowledgments

This work was supported in part by the Department of Energy. The development of the Commissioning Test Shell is part of the IEA ECBCS Annex 40 research project with contributions from National Resources Canada (NRCAN) and the Scientific and Technical Building Centre (CSTB).

References

[1] ANSI/ASHRAE Standard 135-2001, "BACnet- A Data Communication Protocol for Building Automation Systems."

[2]Claridge, D.E., J.S. Haberl, M. Liu, J. Houcek, and A. Athar. 1994. "Can You Achieve 150% Predicted Retrofit Savings: Is It Time for Recommissioning?" In *Proceedings of the 1994 ACEEE Summer Study*. ACEEE, Washington, DC.

[3] T. Haasl, A. Potter, L. Irvine, and L. Luskay. 2003. Portland Energy Conservation, Inc., "Retrocommissioning's Greatest Hits". http://www.peci.org/papers/rCommissioninghits.pdf., 2003.

[4] Visier, J.C. 2003. "Commissioning Of HVAC Systems For Improved Energy Performance: The Annex 40 Approach" in Proceedings of the International Short Symposium on HVAC Commissioning. Kyoto, Japan.

[5] Bushby, S.T., Castro, N.S., Galler, M.A., Park, C., House, J.M. 2001. "Using the Virtual Cybernetic Building Testbed and FDD Test Shell for FDD Tool Development", NISTIR 6818.

[6] Gillespie. K. and Bruceri, M. 2001. "Library of Commissioning Test Protocols", In *Proceedings of the 9th National Conference on Building Commissioning*, Cherry Hill, N.J., May 2001.

[7] Kao, J. Y, 1992 "HVAC Functional Inspection and Test Guide", NISTIR 4758. Functional Test Library.

[8] Choiniere, D. "Diagnostic Agent for Building Operators" Natural Resources Canada.