TESTING CONFORMANCE TO ENERGY MANAGEMENT AND CONTROL SYSTEM COMMUNICATION PROTOCOLS—PART 1: TEST ARCHITECTURE

S.T. Bushby
Associate Member ASHRAE

ABSTRACT

ASHRAE has formed a committee to develop a standard communication protocol for energy management and control systems (EMCS). The goal of being able to connect control equipment from any vendor and make it work as part of an integrated system will not be achieved until tests to determine conformance to the standard are developed. This paper is the first in a two-part series addressing the question of testing conformance to an EMCS protocol.

This paper reviews international efforts to develop procedures for testing conformance to computer communication protocol standards. A particular variation of the "coordinated abstract test method" (ISO 1987c) is proposed as the best architecture for testing conformance to the ASHRAE protocol. This approach will minimize the burden placed on implementors by the conformance test without sacrificing the ability to conduct thorough tests. No direct access to layer boundaries will be required, and integrity of the implementor's software can be maintained. The proposed structure of the ASHRAE protocol lends itself to this approach because only one additional protocol service and one standard object type will need to be added.

INTRODUCTION

In January 1987 ASHRAE formed Standards Project Committee 135P (SPC 135P) to undertake the task of developing a standard communications protocol for building energy management and control systems (EMCS) (Bushby and Newman 1988). The goal of SPC 135P is to develop an industry consensus standard, which, when implemented, will permit control devices made by any manufacturer who complies with the standard to be easily integrated into one control system. This goal will not be completely achieved until control devices can be tested to determine whether they conform to the standard.

The issues involved in testing conformance to protocol standards have received a great deal of attention in the international community as part of efforts by the International Standards Organization (ISO) to develop standards for open system interconnection (OSI). Since 1981, the International Federation for Information Processing has been holding annual conferences on the topic of protocol specification, testing, and verification. Consensus has not been reached on an international standard for conformance testing, but a draft international standard has been prepared (ISO 1987c). This paper contains a review of these international efforts and makes some recommendations for an ASHRAE conformance testing architecture. A companion paper (Bushby 1990) addresses the question of developing the actual tests for determining conformance.

It is important to keep in mind what conformance testing is and what it is not. A conformance test involves testing both the capabilities and the dynamic interactions of an implementation to ensure that they comply with the standard. This is done by comparing observations made during the test with the conformance requirements of the relevant standard. Conformance testing does not assess the robustness, performance, or reliability of an implementation. It also does not pass judgments about the physical realization of the abstract concepts specified in the standard (ISO 1987c).

The complexity of most protocols makes exhaustive testing impractical for both technical and economic reasons. The nature of conformance tests is such that they detect the presence of errors rather than ensure their absence. Thus, passing a conformance test does not guarantee that two devices will be able to communicate.

Steven T. Bushby is an Electronics Engineer in the Mechanical Systems and Controls Group, Building Environment Division, Center for Building Technology, National Institute of Standards and Technology, Gaithersburg, MD.

It merely increases the probability that different implementations are able to communicate.

THE CONFORMANCE TESTING PROCESS

Two different kinds of conformance requirements are usually included in a protocol standard: static conformance requirements and dynamic conformance requirements (ISO 1987c; Rayner 1985). Static conformance requirements define the minimum capabilities permitted in a system and include such things as the grouping of functional units into protocol classes and the range of values that must be supported by specific parameters and timers. Dynamic conformance requirements specify the acceptable behavior of an implementation with respect to communication. Protocol standards generally allow options that will affect both static and dynamic conformance requirements. In order to conduct a conformance test, the implementor must provide a protocol implementation conformance statement (PICS) that specifies the particular options that have been implemented.

Each conformance requirement must have a specific test, which consists of a sequence of steps or actions to be taken. When the whole sequence is complete, the observed results are compared with some previously defined evaluative criteria to determine conformance or nonconformance to that requirement. All of these individual tests are combined to form a test suite. The information contained in the PICS is used to select the individual tests in the suite, which will be used to determine if a particular implementation conforms to the standard.

Figure 1 is a flowchart indicating a sequence of steps that might be followed when testing conformance of an implementation (ISO 1987c; Rayner 1985). The first step is a review of the PICS to determine if the description of the implementation is consistent with the static conformance requirements of the standard. The second step, basic interconnection testing, is optional. It provides a very limited set of tests to establish that at least some type of interconnection is feasible.

The third step in the testing process is functional range testing. This is a series of tests to determine that all required services and all of the optional features stated in the PICS are supported. Each service parameter is tested to ensure that it is supported over the range of values specified in the standard. Functional range testing is the primary way to determine compliance with static conformance requirements.

The next step in the testing process is dynamic conformance testing, which is intended to provide as thorough a testing of an implementation as practical over the full range of requirements specified in the standard. The set of all possible combinations of events is infinite, making it impossible to prove that an implementation conforms dynamically in all instances of communication. Testing an implementation can only show that it consistently conforms dynamically in representative instances of communication. Designing the test suite is a trade-off between attempting to minimize the number of nonconforming implementations that pass the test suite

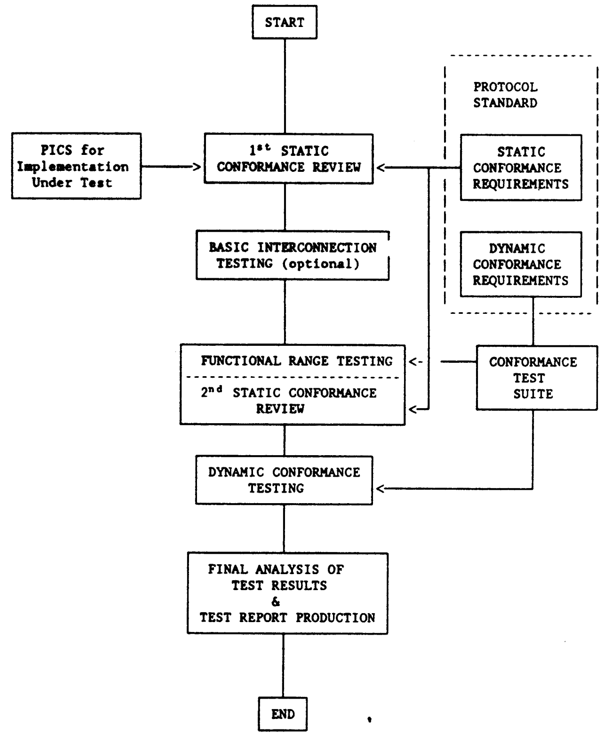


***Figure 1** Typical procedure for a conformance test*

and making the number of tests small enough to be practical to administer them.

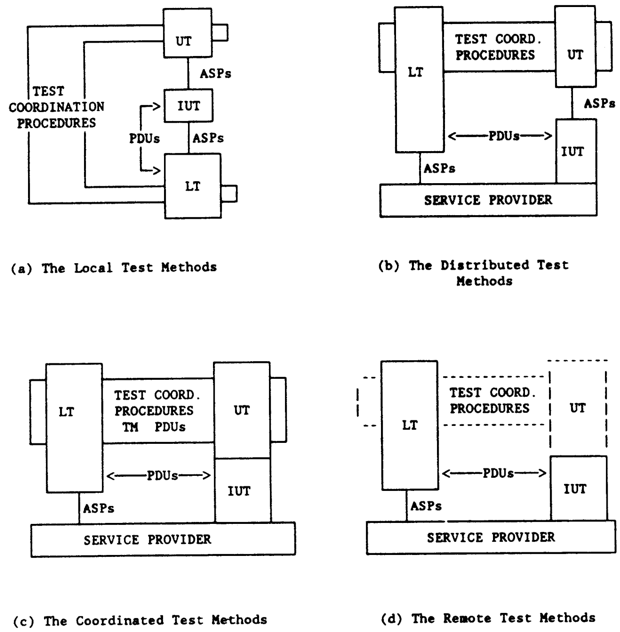
The last step in the testing process is the final analysis of the results and preparation of a test report indicating the results. If an implementation fails the test suite, the report should clearly indicate which test failed and what function that particular test is designed to check.

ABSTRACT TESTING METHODOLOGIES

The ISO draft proposal for conformance testing describes four different types of abstract testing methodology (ISO 1987c). The four approaches are distinguished by the particular inputs to the entity under test that can be controlled and the outputs that are observed.

The four abstract methodologies are illustrated in Figure 2. Each abstract test method describes the role of an upper tester and a lower tester. In the context of the ASHRAE protocol, the lower tester can be thought of as a way to represent communication activity from other devices on the network. The upper tester represents the application programs within the controller that initiate communications. More accurately, a lower tester is the abstraction of the means to observe and control the lower service boundary of the implementation under test (IUT). An upper tester is the abstraction of the means to observe and control the upper service boundary of the IUT. All of the test methods may be applied to testing a single layer or to multiple layers.

When the upper and lower testers have direct access to the abstract service primitives (ASP) and the protocol data units (PDU) at the layer boundaries, the test method is said to be a local test method (Figure 2a). One



Key: ASP - abstract service primitive
IUT - implementation under test
LT - lower tester
PDU - protocol data unit
TM - test management
UT - upper tester

***Figure 2** Overview of abstract test methods*

way to realize a local test method would be to implement the upper tester, the lower tester, and the IUT as separate processes running concurrently on a single computer. Direct access to the ASP and PDU at the layer boundaries and coordination between the upper and lower testers would be provided by interprocess communication.

Direct access to the layer boundaries is not always possible. External test methods are characterized by a lower tester, which is separated (has no direct access to the lower service boundary), together with either direct or indirect access and control at the upper boundary of the IUT. Three distinct types of external test methods have been recognized by the ISO: distributed, coordinated, and remote (ISO 1987c). They vary according to the requirements placed on the coordination between the upper and lower testers; the access to the layer boundary above the IUT; and the requirements on an upper tester. For all external test methods, the lower tester has only indirect access to the lower service layer boundary of the IUT.

A distributed test method (Figure 2b) is characterized by an upper tester that may or may not have direct access to the upper service layer boundary. The functions for the upper and lower testers are specifically defined, but no assumptions are made about how the two testers are coordinated.

A coordinated test method (Figure 2c) also requires specific functions in the upper tester, but the upper tester is included as part of the system under test. Access to the upper boundary of the IUT is not required because interaction between the realization of the upper tester and the IUT is hidden. Coordination between the two testers is achieved by using a special test management protocol. A test management protocol is one or more protocol services added to the standard that are used only for testing conformance. They are not used during the normal operation of the device.

A remote test method (Figure 2d) has no defined upper tester. This approach applies when some functions of the system under test can be used to control the IUT during testing. No assumptions are made concerning the means of coordinating the lower tester with the relevant control functions of the system under test.

Two important issues must be considered when selecting the appropriate abstract testing methodology for an EMCS protocol standard. The first is the means for coordinating the activities of the upper and lower testers. The second is the impact that assumptions about access to layer boundaries will have on implementing the protocol.

Several approaches to synchronizing the activities of the two testers have been tried, most with limited success. Human interaction with the upper tester through a terminal has been tried (Cowin et al. 1983; Rayner 1985). This has been done with terminals directly connected to the upper tester and with terminals connected to the IUT over a telephone line. Using timers in both testers has been attempted (Rayner 1985). Another approach involved developing compatible test scenarios for each tester, which are implicitly synchronized by the normal exchange of PDU during the test (Linn and Nightingale 1983a, b; Linn 1984; Nightingale 1982). Adding additional test protocol services that provide the needed coordination has also been attempted (Cowin et al. 1983; Rayner 1985; Zeng and Rayner 1985). All of these methods, except using timers, are mentioned in annex A of the ISO draft proposal on test realization (ISO 1988). Each of these methods will be considered in turn.

The human interaction approach would require the person conducting the test to access the IUT through some operator interface provided by the implementor. When actions need to be initiated from the IUT side, the appropriate commands are entered at the terminal. For a high-level controller this is not a problem because the same operator interface that is sold with the control system can do the job. This may not be so simple for low-level controllers, because they may not have an adequate interface or the hardware capabilities to add one.

There are other problems with this approach. What happens if the operator makes a mistake? This would probably be interpreted as a test failure. Retesting might be required to ensure that a failure was not caused by human error. Control systems made by different vendors have different operator interfaces. The test personnel would have to learn a different interface for each IUT, and this increases the chance for human error. If the operator uses a terminal connected to the IUT by telephone, new problems are introduced. Any difficulty with this communication link may be interpreted as a test failure. Also, many EMCS controllers have no dial-in capability. Finally, human interaction is slow. Relying on a human operator has not proved to be satisfactory in practice (Rayner 1985).

One approach used to test transport layer protocols was to have peer scenario interpreters, one at each end. The scenarios were so constructed that if a service primitive was to be generated on one side, the peer scenario interpreter would be programmed to expect that primitive and issue the appropriate response. Thus, the scenario interpreters were implicitly synchronized by the exchange of service primitives (Nightingale 1982).

If a special test management protocol is used to synchronize the testers, it may be implemented using the same connection that carries the test traffic, a separate parallel connection through the IUT or an external parallel connection. For all three cases, the IUT is required to have increased capability to support the synchronization protocol. It also clouds the issue somewhat with regard to a test failure. Did the failure occur because of a protocol error in the synchronization service or in the portion of the implementation being tested? If the synchronization services are included in the conformance requirements of the standard, the answer to this question becomes largely irrelevant.

An interesting variation of this approach is the use of "ferries" (ISO 1988; Zeng and Rayner 1985). In this scheme, both the upper and lower testers are in the same machine and can synchronize by using interprocess communication techniques. Each tester has an encoder/decoder associated with it. The encoder/decoder for the lower tester is local to the test machine, and the other one is connected to the upper layer boundary of the IUT. The test protocol works like a ferry, carrying instructions between the decoders. These instructions are equivalent to the peer scenarios described earlier.

None of these approaches appears satisfactory at first glance. The human interface, use of telephone connections, setting timers, and implicit scenario synchronization have all been reported by some experienced users as unsatisfactory (Linn 1985; Rayner 1985). The additional test management protocol approach has been regarded by some as more promising (Rayner 1985). It will be argued later that the test management protocol approach is the best one for EMCS protocol conformance testing.

The level of access to layer boundaries is the other important issue to be resolved. If direct access to layer boundaries is required by the test method, this forces the implementor to place "hooks" in the code that permit the test software to access the layer boundaries. This not only increases the complexity of the implementation code, but it also could present a security concern. Anyone with knowledge of the test environment could use the same hooks to access and affect the implementor's software.

One solution to the problem of leftover "hooks" is to write the code with the "hooks" flagged for inclusion only when special compiler options are used. After conformance testing, the code is recompiled, this time ignoring the specially marked code. This solution raises a new question about whether the recompiled code still conforms to the standard.

The market for EMCS devices is very competitive, particularly for controllers at the low end of the performance scale. Conformance-testing requirements that force a manufacturer to add capability to its low-end controllers or provide a way for someone to interfere with the operation of the proprietary control software are not likely to be accepted. These restrictions become more important when the issue of single-layer testing vs. multilayer testing is considered.

If a single-layer testing approach is taken, the implementor must provide this access to the interfaces between all layers. If a multilayer testing approach is taken, only the upper boundary of the highest layer and the lower boundary of the lowest layer under test are involved. Multiple-layer approaches with limited requirements for access to boundary layer interfaces will be favored by manufacturers. It is possible to accommodate these concerns and still have a useful conformance test.

PROPOSED ABSTRACT TEST METHODOLOGY FOR ASHRAE CONFORMANCE TEST

The Standards Committee is considering a collapsed architecture for the ASHRAE EMCS communication protocol standard (Bushby and Newman 1988). It is unclear exactly which layers in the OSI model will be used, but the ASHRAE standard will include the physical layer, data link layer, and the application layer. The transport layer may or may not be included; the presentation and session layers almost certainly will not be included. Network layer functionality will probably be provided implicitly in the addressing scheme and will not require a separate protocol layer. The standard will probably provide the option of more than one physical medium and more than one medium access control scheme. This will affect the physical and data link layers of the protocol. These two layers will most likely be implemented in hardware and tested independently. Testing the hardware components is outside the scope of this paper. If the data link layer were to be implemented in software, it would have to be included in the test architecture described below. This approach of testing the lower layers separately has been advocated by the Corporation for Open Systems in its plans for developing conformance tests for OSI (Davidson 1987).

This narrows the problem to testing conformance only in layers above the data link layer, including, at most, the application, transport, and network layers. If two or more of these layers are included in the standard, then a multilayer abstract test procedure should be used to minimize the constraints placed on the implementor. The idea of testing layers in groups is becoming generally accepted in the international community (Davidson 1987; Rayner 1985). In this paper it will be assumed that the transport layer is included in the standard and the network layer is included implicitly in the addressing scheme.

Figure 3 shows the proposed test architecture. This is an example of the coordinated test method. It is assumed that the lowest two layers have been tested by other means and are known to be reliable. The right side of Figure 3 shows the IUT. The left side shows the conformance tester, which is a reference implementation of the protocol known to conform to the standard. The test driver, which controls the tests and analyzes the results, can access the layer boundaries on the tester

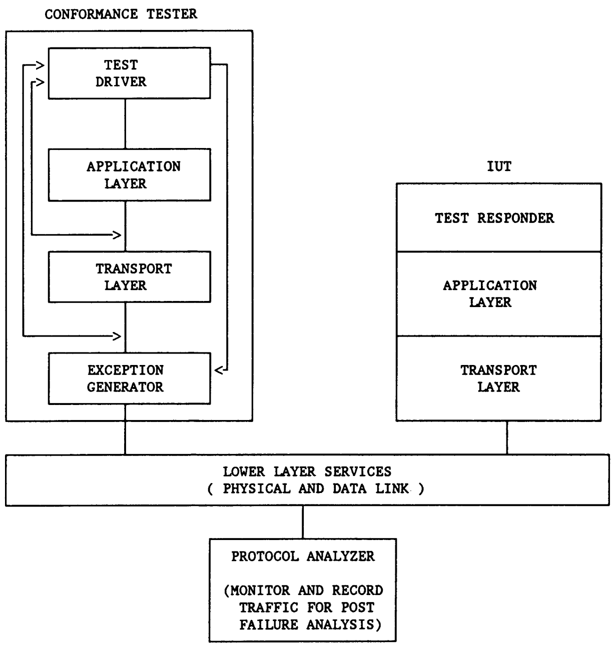


***Figure 3** Proposed ASHRAE abstract conformance test architecture*

side. By examining the incoming PDU at the lower boundary of each layer, it is possible to infer the status of the boundaries on the IUT side, thus eliminating the need for direct access to the IUT layer boundaries. In Figure 3, the conformance tester fulfills the role of the abstract lower tester and the test responder corresponds to the abstract upper tester.

Objections to using reference implementations for testing protocol conformance have been raised on two grounds (Cowin et al. 1983). Reference implementations can only encode valid sequences of PDU and should treat all invalid or out-of-context PDU as fatal errors. This problem can be overcome by using an exception generator, as shown in Figure 3. The exception generator can synthesize invalid PDU, duplicate valid PDU, and suppress PDU altogether (Linn and Nightingale 1983a, b). These activities are controlled by the test control and analysis software in the test driver and permit a thorough testing of the ability of the IUT to handle these types of problems correctly.

The second objection to the reference implementation approach is that a fixed algorithm is generally used to encode and decode PDU (Cowin et al. 1983), which makes it difficult to generate or respond to alternative correct encodings. This is not a problem for the ASHRAE protocol because only one encoding scheme for PDU will be permitted in the standard, which is the reason for not including a presentation layer. Reference implementations have been successfully used for protocol testing in the past (Linn and Nightingale 1983a, b; Nightingale 1982), and the ASHRAE protocol will be well suited to this approach.

The alternative to a reference implementation is the use of an encoder/decoder to process PDU at the two layer boundaries. A special test language, which controls the encoder/decoder, is used for the upper and lower testers (Cowin et al. 1983; Muralidahr 1987). An exception generator is not needed if the test language includes the ability to generate duplicate and incorrect PDU.

The reference implementation approach is appealing, because it can be closely tied to the use of formal description techniques (FDT) such as ESTELLE and LOTOS (ISO 1987a, b), for specifying a protocol. Using an FDT as part of the protocol specification can avoid the ambiguities about both the specification and the conformance requirements that often occur with English language specifications (Rayner 1983). Techniques are being developed to apply formal approaches to generate test sequences from FDT specifications. Other techniques are being developed to semiautomatically generate implementations of a protocol from the formal description. This can be used to help generate a reference implementation.

The automatic generation of implementations and test sequences is not fully developed at this time, although they are being used with some success. There is a potential for time savings if these approaches are used, but the real benefit is more likely to be increased certainty of correctness for both the test suite and the reference implementation.

Application layer protocols are not symmetrical. For example, sending a file is a different process than receiving a file. There must be a way to cause the IUT to initiate various application layer interactions with the reference implementation in order to test for conformance to the protocol standard. This is the purpose of the upper tester in the abstract test methodology. In a coordinated test approach, the activities of the upper tester are synchronized with the lower tester by exchanging messages through a special test management protocol. This eliminates the need for direct access to the application layer boundary and some other external method of communication between the upper and lower testers. There is, of course, a cost involved with providing the additional protocol services.

The ASHRAE protocol will include standardized objects of various types and services for requesting the value of particular properties of those objects. A special object called a TestObject can be created with the properties shown in Table 1. This object will contain all of the information an upper tester will need to initiate a particular application interaction.

Adding only one additional service to the protocol will be sufficient for coordinating the activities of the upper and lower testers. This service will amount to a message that tells the test responder to read the properties of the TestObject and start a transaction based on what is found. The conformance tester can manipulate the values of the properties of the TestObject and the timing of the instructions to the test responder in order to completely control the required tests.

In operation, the conformance tester would use the protocol WriteProperty service to write the value of a particular service to the property ServiceName. For each protocol service, there will be a set of parameters included in the PDU when the service is invoked. The lower tester will write each of these parameters, in order, to the

TABLE 1
Properties of a TestObject

Key Property: Identifier {the name of this object, e.g., ConformTest}
Property: ServiceName {the application service to be invoked by the IUT}

Property: ServiceParam1 {1st parameter needed to construct APDU}
Property: ServiceParam2 {2nd parameter needed to construct APDU}
Property: ServiceParam3 {3rd parameter needed to construct APDU}
.
.
.
Property: ServiceParamN {Nth parameter needed to construct APDU}

Note: The number of parameters included as properties of this object will be the maximum number needed to construct all application protocol data units (APDU) for the conformance class to which this device belongs.

appropriate ServiceParam property of the TestObject in the IUT. At this point, the tester can use the conformance testing service to tell the IUT to initiate a transaction. The IUT reads the values of the TestObject properties and constructs the appropriate PDU, passing it to lower layers in the normal fashion. To the IUT this should look similar to a request from an application program for protocol services.

This approach does require the implementor to include one more protocol service and to write software that can interpret the information contained in the TestObject and take the appropriate action. The advantage is that there is no need to build any hooks for direct access to any layer boundaries in the IUT and no need for external communication links. This should minimize the additional effort required to accommodate conformance testing and eliminate altogether any concerns about possible security problems associated with access to layer boundaries.

At the bottom of Figure 3, a protocol analyzer is shown. This is a device that can monitor and record all of the PDU being transmitted over the physical medium. It also has some capability to filter these PDU to permit analysis of protocol interactions between the two systems. The protocol analyzer serves two functions. It was assumed that there were no problems with the physical and data link layers. If a conformance test fails, the protocol analyzer could be used to investigate the possibility of a problem in these layers. It could also be useful for a different type of testing, not previously mentioned, called conformance resolution testing.

If two implementations pass a conformance test but cannot interoperate, special tests, called conformance resolution tests, may need to be conducted. This will determine which implementation is not in compliance with the protocol standard (ISO 1987c). This test would have to be customized to the particular circumstances involved, and the detailed information available from the protocol analyzer may be helpful in resolving this question.

SUMMARY AND FUTURE DIRECTIONS

The meaning of conformance testing has been defined in terms of international standardization activities relating to open system interconnection. Various abstract test methodologies have been proposed by the international community, and each has its own strengths and weaknesses. Each approach also has different implications from the standpoint of the cost involved in implementing the protocol.

The issues of single- vs. multilayer testing, access to layer boundaries, and test synchronization procedures have been analyzed in the context of the current state of the building control industry and what is known about the likely content of a future protocol standard. A coordinated test approach with added protocol services for test synchronization has been proposed. By defining a TestObject that has properties containing all of the information an upper tester will need to initiate a required interaction, synchronization can be achieved with only one additional protocol service. The proposed test methodology should minimize the added cost to implementors of the protocol without sacrificing the ability to conduct thorough conformance tests.

The second paper in this series (Bushby 1990) addresses the problem of generating the actual tests that will be needed to determine conformance to the standard. Although much work remains to be done before the standard is completed, enough is now known to begin development of the conformance tests. Developing conformance tests in parallel with the standard will assist the committee in making the conformance requirements clear and will enable a conformance testing procedure to be ready for use soon after the standard is complete.

REFERENCES

Bushby, S.T. 1990. "Testing conformance to energy management and control system communication protocols—part 2: generating test suites." *ASHRAE Transactions*, Vol. 96, Part 1.

Bushby, S.T., and Newman, H.M. 1988. "Standardizing energy management and control system communication protocols." *ASHRAE Journal*, Vol. 31, No. 1, January, pp. 33-36.

Cowin, G.W.; Hale, R.W.S.; and Rayner, D. 1983. "Protocol product testing—some comparisons and lessons." *Proceedings of the IFIP WG 6.1 Third International Workshop on Protocol Specification, Testing, and Verification. Protocol Specification, Testing, and Verification, III.* North-Holland, pp. 477-492.

Davidson, I.C. 1987. "OSI protocol testing at the corporation for open systems." *Proceedings of the IFIP WG 6.1 Seventh International Workshop on Protocol Specification, Testing, and Verification. Protocol Specification, Testing, and Verification, VII.* North-Holland, pp. 3-12.

ISO. 1987a. ISO draft international standard 8807, "Information processing systems—open systems interconnection—LOTOS (formal description technique based on the temporal ordering of observational behavior)." International Standards Organization. Available from ANSI, 1400 Broadway, New York, NY 10018.

ISO. 1987b. ISO draft international standard 9074, "Information processing systems—open systems interconnection—ESTELLE—a formal description technique based on an extended state transition model." Available from ANSI, 1400 Broadway, New York, NY 10018.

ISO. 1987c. ISO draft international standard 9646-1, "Information processing systems—open systems interconnection—OSI conformance testing methodology and framework—part 1: general concepts." International Standards Organization. Available from ANSI, 1400 Broadway, New York, NY 10018.

ISO. 1988. ISO draft proposal 9646-4, "Information processing systems—open system interconnection—OSI conformance testing methodology and framework—part 4: test realization." International Standards Organization. Available from ANSI, 1400 Broadway, New York, NY 10018.

Linn, R.J. 1984. "An evaluation of the ICST architecture after testing class 4 transport." *Proceedings of the IFIP WG 6.1 Fourth International Workshop on Protocol Specification, Testing, and Verification. Protocol Specification, Testing, and Verification IV.* North-Holland, pp. 611-621.

Linn, R.J., and Nightingale, J.S. 1983a. "Some experience with testing tools for OSI protocol implementations." *Proceedings of the IFIP WG 6.1 Third International Workshop on Protocol Specification, Testing, and Verification. Protocol Specification, Testing, and Verification, III.* North-Holland, pp. 521-531.

Linn, R.J., and Nightingale, J.S. 1983b. "Testing OSI protocols at the National Bureau of Standards." *Proceedings of the IEE, Special Transactions on Communications,* December.

Muralidahr, K.H. 1987. "MAP 2.1 network management and directory services test system." *Proceedings of the IFIP WG 6.1 Seventh International Workshop on Protocol Specification, Testing, and Verification. Protocol Specification, Testing, and Verification, VII.* North-Holland, pp. 359-369.

Nightingale, J.S. 1982. "Protocol testing using a reference implementation." *Proceedings of the IFIP WG 6.1 Second International Workshop on Protocol Specification, Testing, and Verification. Protocol Specification, Testing and Verification, II.* North-Holland, pp. 513-522.

Rayner, D. 1983. "Towards an objective understanding of conformance." *Proceedings of the IFIP WG 6.1 Third International Workshop on Protocol Specification, Testing, and Verification. Protocol Specification, Testing, and Verification III.* North-Holland, pp. 493-503.

Rayner, D. 1985. "Towards standardized OSI conformance tests." *Proceedings of the IFIP WG 6.1 Fifth International Workshop on Protocol Specification, Testing, and Verification. Protocol Specification, Testing, and Verification V.* North-Holland, pp. 441-460.

Ural, H. 1987. "A test derivation method for protocol conformance testing." *Proceedings of the IFIP WG 6.1 Seventh International Workshop on Protocol Specification, Testing, and Verification. Protocol Specification, Testing, and Verification, VII.* North-Holland, pp. 347-358.

Zeng, H.X., and Rayner, D. 1985. "The impact of the ferry concept on protocol testing." *Proceedings of the IFIP WG 6.1 Fifth International Workshop on Protocol Specification, Testing, and Verification. Protocol Specification, Testing, and Verification V.* North-Holland, pp. 519-531.

DISCUSSION

D. Underwood, M.E., USA CERL, Champaign, IL: Does your future work include creating something to do the conformance testing or just the outline of procedures to perform conformance testing?

S. Bushby: We plan to build a working conformance-testing system at NIST and make this system available to control manufacturers for testing prototype controllers. This will provide valuable feedback about both the conformance tests and the protocol itself. NIST will not become a certification agency for the ASHRAE protocol, but we hope the tests developed here will become the basis for such a certification process.