Module 53—Center-to-Center Reference Implementation Introduction

Ken Leonard: ITS standards can make your life easier. Your procurements will go more smoothly and you'll encourage competition, but only if you know how to write them into your specifications and test them. This module is one in a series that covers practical applications for acquiring and testing standards-based ITS systems.

Ken Leonard: I am Ken Leonard, the Director of the U.S. Department of Transportation's Intelligent Transportation Systems Joint Program Office. Welcome to our ITS Standards Training Program. We are pleased to be working with our partner, the Institute of Transportation Engineers, to deliver this approach to training that combines web-based modules with instructor interaction to bring the latest in ITS learning to busy professionals like yourself. This combined approach allows interested professionals to schedule training at your convenience, without the need to travel. After you complete this training, we hope that you will tell colleagues and customers about the latest ITS standards and encourage them to take advantage of these training modules, as well as archived webinars. ITS standards training is one of the first offerings of our updated Professional Capacity Training Program. Through the PCB program, we prepare professionals to adopt proven and emerging ITS technologies that will make surface transportation safer, smarter, and greener. You can find information on additional modules and training programs on our website at www.pcb.its.dot.gov. Please help us make even more improvements to our training modules through the evaluation process. We look forward to hearing your comments. Thank you again for participating, and we hope you find this module helpful.

Ken Vaughn: Hi. This is Ken Vaughn. This course is the T251 Module: Center-to-Center Reference Implementation Introduction. This course includes a separate student supplement, which contains useful information for a more complete understanding of this material, but we'll try to cover as much as possible in this course.

Ken Vaughn: I am the presenter, Kenneth Vaughn, the president of Trevilon, LLC and I've developed test software to verify device conformance to NTCIP, providing testing services to the industry for more than a decade. I'll be your presenter today.

Ken Vaughn: The course includes four main learning objectives. The first learning objective is recognizing the purpose of the C2C reference implementation; the second is to acknowledge key capabilities and limitations of the C2C RI; the third is following the process for producing test documentation that relies upon the C2C RI; and the fourth will be to recognize results a tester might produce after using the C2C RI.

Ken Vaughn: The first learning objective includes four key points that we want to talk about, including understanding C2C communications, appreciating the purpose of the reference implementation, explaining the communication stack supported by the C2C RI, and then also knowing where to obtain the software. We'll address each of those in order, the first one being to understand C2C communications.

Ken Vaughn: ITS is a complex collection of technologies that are designed to improve mobility and safety while reducing the environmental impact of transportation. While some of these ITS benefits can be achieved by individual components acting alone, most of the significant benefits can only be realized once components are able to share information with one another. This presents unique challenges. Many of the components that need to interact are mobile and need to be trusted. Recognition of this complex environment led the U.S. DOT to identify and standardize these interfaces.

Ken Vaughn: The National ITS Architecture is one of the foundational tools of the ITS standards Program. This figure provides one of the more recognizable graphics resulting from the development of the National ITS Architecture, and provides a high-level, conceptual overview of ITS. The architecture identifies various subsystems common in the ITS environment and categorizes them into four major types: Travelers, that you see here in the upper left; Centers, in the upper right; Vehicles, in the lower left;

and Field devices, in the lower right. Note that this is a reference architecture. An actual implementation architecture may have some items that fit into a gray area. For example, an automated management system could be considered a field device or a center, depending upon the designer's choice. In order for any two of these subsystems to communicate with each other, a communications pathway must exist. The type of subsystems being connected affect which communication technologies can be used. This figure is often called "The Sausage Diagram," because the communication links look like sausages.

Ken Vaughn: We'll now look at the center subsystems in more detail, because that is the subject of the C2C Reference Implementation. We really only focus on center-to-center. This deals with the testing of a center to verify that it conforms to standards to communicate with other centers. This type of communications is known as center-to-center, or C2C, communications. There are a variety of different types of centers, but regardless of type of center, the communications between centers are classified as fixed point-to-fixed point communications, as shown by the sausage at the bottom. Fixed point-to-fixed point communications can generally be considered to be any high-speed link where the amount of data exchanged is not considered a critical issue. This may even include wireless connections for temporary incident management centers.

Ken Vaughn: Now, the National ITS Architecture does much more than identify those centers. It also identifies a range of service packages that summarize capabilities that can be offered by ITS. This graphic identifies a small subset of the service packages identified by the National ITS Architecture. Overall, the National ITS Architecture identifies over one hundred and fifty service packages. As the name suggests, each service package is a subset of the physical architecture that has been tailored to address a real-world transportation problem. Each service package is described in detail, complete with a graphic showing interactions among the various involved subsystems. We'll now take a look at one of these service packages: the Advanced Traveler Information Systems service package.

Ken Vaughn: This is what that diagram looks like. Each service package is described in detail with a diagram like this, complete with a graphic showing interactions among the various involved subsystems. As we mentioned, we're primarily concerned with the center-to-center exchanges, so we'll zoom in on that portion of the diagram.

Ken Vaughn: The C2C RI focuses on the information flows covered by the Traffic Management Data Dictionary. The TMDD defines flows between an owner center and an external center, where the owner center is the center that logically owns the entity being described by the information being shared. The "Traffic Management" title of the standard reflects the scope of flows addressed by the standard. As such, these information flows often have the traffic management center as the owner center and some other center as the external center; in this case, the transportation information center. For example, incident information is one information flow that is sent from a traffic management center to the transportation information center for this service package. Road network condition is another information flow sent among these two centers. However, this same information may be exchanged between other centers as well. We'll take a look at that next.

Ken Vaughn: Here, we see the same flow—incident information—being exchanged from the emergency management center to the transportation information center. The C2C RI is useful to test any interface that is based on the TMDD standard regardless of the type of center that's being evaluated. However, while the C2C RI will likely be a primary tool to test the traffic management center, it may only be a supplemental tool to test other types of centers. So, whereas the C2C RI will almost completely test the traffic management center, you may need other tools to help test the emergency management center. We also see that other flows defined in the TMDD may appear in other places on the diagram. In this case, we have the maintenance and construction management center connected to the transportation information center. The C2C RI is useful to test any interface based on the standard, even if it's not just for the TMDD.

Ken Vaughn: In short, complex systems require proper verification. The National ITS Architecture defines more than one hundred and fifty service packages, with most service packages relying upon multiple information flows to enable the service. Each of those information flows has to work correctly in order for the service package to be provided as intended, and each of the intended information flows is associated with multiple design requirements in order to allow the proper exchange of information. Of course, most deployments are designed to provide more than one service. In short, any given deployment will have a host of potential interoperability failure points. This level of complexity requires rigorous testing in order to ensure that the system will provide the desired services in a reliable manner. Testing a single information flow is simply not sufficient.

Ken Vaughn: Ultimately, the entire purpose of ITS standards and testing is to verify interoperability. But what exactly is interoperability? Interoperability is the main purpose. It is defined as the ability of two or more systems or components to exchange information and use the information that has been exchanged. In other words, the receiving subsystem must not only be able to receive the information, it must be able to understand the information that it has received. In order to realize the desired level of interoperability, we must clearly define our needs and specify the desired system with requirements. Standardized needs and requirements are defined within the TMDD standard itself. Modules A321a and A321b provide guidance on how to specify these optional features from these standards. The TMDD also defines a precise design for each requirement and the rules for traceability from the user needs to the requirements to the detailed design. Module T321 assists in developing the test documentation to verify and validate the system. This module, T251, goes a step further and describes how the Center-to-Center Reference Implementation can be used to assist this verification using the test plan defined in Module T321.

Ken Vaughn: The C2C RI is designed to verify that a center implements the desired information flows with another center according to industry standards. Verification is a critical part of the systems engineering process, which is often shown with the "V" diagram. Every agency should verify that each procured system conforms to the necessary standards to minimize problems and expenses later. This training module discusses both Version 1 and Version 2 of the tool. Version 2 has not been released yet, but we do know the key additions that will be added to Version 2 software. Since this module will be available on the website for some time, we'll discuss those features as a part of this module, but as we talk about those features, we'll highlight them with the Version 2 logo.

Ken Vaughn: The C2C RI is designed to support testing a single center subsystem, which may be the owner subsystem or the external subsystem. The user needs and requirements supported by the center subsystem should've already been selected through the use of a Needs to Requirements Traceability Matrix. The user of the C2C RI can then select those needs within the C2C RI tool, as well as the specific requirements also within the tool. The C2C RI will then automatically identify which test cases might apply to those particular user needs and requirements selected. The user can then choose to perform any or all of those identified test cases as a part of the subsystem verification. It should be noted that the C2C RI does not validate a system. Validation occurs towards the end of the process, as shown in the upper portion of the upper-right side of the "V" diagram. Validation can only really be checked with the full system deployment, not through component testing. The C2C RI verifies that the owner or external center complies with its requirements to the extent that conformance to the TMDD and 2306 is required. This typically takes place prior to actual deployment operation, and may be tied to acceptance of the system.

Ken Vaughn: The C2C RI is customizable to your project. Specifically, there's three operational modes of the RI that allow this. The first is the configuration mode. The configuration mode allows the user to customize the tool to identify the needs and requirements to which the system claims conformance. This ensures that the C2C RI will only test specific flows of interest. It uses the specific communications profile and settings used by the project. The test mode allows a user to perform the test cases that have been selected. But once again, while the configuration allows the user to identify user needs and requirements that identify tests that may be performed, the user can select which test cases to actually perform within the test mode. Finally, the report mode allows the user to produce reports that document the test cases performed and errors discovered, so that they may be provided to the end client and archived in project files.

Ken Vaughn: Now, if we go back to the National ITS Architecture, we see the details for implementing any information flow are defined in ITS standards. We can use National ITS Architecture to identify the relevant standards and determine whether or not the C2C RI is applicable to this particular information flow. From the service package diagram shown earlier, the user can click on an information flow that connects a source to a destination, and the National ITS Architecture will show a complete description of the associated flow. In this case, we're looking at the incident information flow from the traffic management center to the transportation information center. Within this screen you see the complete definition of the incident information, as well as the traffic management center and transportation information center. At the top of the screen, you have tabs. We'll click on the "Communications Diagrams" tab.

Ken Vaughn: Clicking on that tab will show you this figure. This figure depicts how various standards relate to one another for this specific information flow using the open systems interconnect model. The ITS application information layer identifies the ITE TMDD standard. This standard identifies and justifies specific end user needs for traffic management, and then provides detailed functional requirements and detailed designs for each of these needs in a traceable environment. Other information level standards can be used for other types of centers. The application layer includes an option for NTCIP 2306, and this is transported over HTTP or FTP. NTCIP 2306 relies upon XML at the presentation layer, and there's no need for a session layer in this particular model. It is generally understood that all fixed point-to-fixed point communications will be transported over standard internet protocols TCP and IP, and those can be transmitted over a wide array of data link layers and physical back haul layers.

Ken Vaughn: For a more detailed view, we've prepared the following diagram that shows how the specific standards relate at the upper layers. The traffic management data dictionary for C2C communications is at the highest layer, and it defines the messages to perform TMC-related operations. Those messages can be transported over either SOAP or Direct XML. SOAP stands for "Simple Object Access Protocol." It's the preferred solution for exchanges that need error handling or include published subscribe support. It is the best standard that's out there. But for simple implementations, we also allow Direct XML. This provides a simple solution for one-way exchanges and very simple request-reply without error handling or significant security. SOAP is transported over HTTP, with or without security. Direct XML can be transported over HTTP, without security or FTP. FTP is limited to one-way information flows, and thus is limited to the XML solution, whereas HTTP supports bi-directional exchanges with optional security. In addition to these main standards, there's one other component which is called WSDL, the Web Services description Language. The WSDL allows an owner center to advertise which services are available in its center. This allows an external center to discover what services and owner center offers, and then use those centers that use those services at a later point. That fully describes the upper layers of the C2C Reference Implementation and what it's designed for. You can find a full list of acronyms within the Student Supplement.

Ken Vaughn: So what does the C2C RI support? Well, it's able to verify TMDD—specifically Version 3.03c—and NTCIP 2306 Version 1.69. It assumes that these standards are implemented over a standard Internet stack. The RI will use the stack as configured by the WSDL. Right now, it's only been tested against IPv4, but in theory, it should also work over IPv6, assuming that the two systems support that protocol. With expert customization, the C2C RI can be extended to test other information and application-level standards and specifications.

Ken Vaughn: So, how do you obtain this C2C RI? Well, luckily, it's available for free. It is available at https://www.standards.its.dot.gov/DeploymentResources/Tools. The download includes a user manual, and technical support is available at C2CRIsupport@transcore.com.

Ken Vaughn: Well, that brings us to our first pop quiz.

Ken Vaughn: Which standard is not supported by the C2C RI without customization? There are four possible choices. A) Internet Protocol (v4); B) XML Center-to-Center Profile (NTCIP 2306 v1.69); C) Transit Communications Interface Profile (TCIP v5.0); and D) Traffic Management Data Dictionary (TMDD v3.03c). So go ahead and make your choice there as to which standard is not supported by the C2C RI.

Ken Vaughn: Now that you've made your choice, we'll review the answers. The correct answer is C) Transit Communications Interface Profiles, TCIP. The C2C RI does not include a test suite for transit data, however a custom test suite could be developed using the extensibility mechanisms. The Internet Protocol is supported by the C2C RI. C2C RI uses IP for all communications and allows users to configure the IP address and destination. XML Center-to-Center profile likewise is supported by the C2C RI. C2C RI includes a suite of tests' support to verify that a test system conforms to this standard. Likewise, the Traffic Management Data Dictionary is supported by the C2C RI. It includes a suite of tests to verify the system conforms to the TMDD.

Ken Vaughn: Well, that brings us to our second learning objective. We've discussed recognizing the purpose of the C2C Reference Implementation.

Ken Vaughn: We'll now move on to acknowledging the key capabilities and limitations of the C2C RI.

Ken Vaughn: The first key requirement is to recognize the environment required to operate the C2C RI. The minimum system requirements are Windows 7 or 8 using a 64-bit professional version of Windows 7 or 8. We have tried Windows 10, but it has not been fully tested on that environment to-date. We also recommend a 2 GHz processor with 4 GB of RAM, 1GB storage, a 1 GBps Ethernet interface, as well as Java SE Runtime environment, Version 7.17.

Ken Vaughn: There are some limitations of the correct version of the C2C RI. Perhaps most importantly, the user requires an extensive skillset. While the C2C RI is an invaluable resource for testing systems, it does require a specialized user with an extensive skillset, which we'll go through a few slides from now. The tool is also limited supporting the TMDD unless you extend it. It does not support any other C2C informational layer standards, such as the incident management standard, the transit standards, or the ATIS standards. Finally, the application layer is limited to NTCIP 2306. It only supports this application layer without significant customization. It does not support other application layers such as CORPA or DATEX.

Ken Vaughn: We talked about the skills that would be needed. This slide provides an overview of the skills. We'll discuss each one of these in detail. First of all, encoding languages. The user will need to understand encoding languages, protocols that are used by the standards; the Windows networking environment. You'll need to understand the ITS standards in detail, and also it's very useful for the tester to have testing experience to understand the test process and what issues may arise during testing. You may also need to have the ability to learn a scripting language, and then also a deep detailed understanding of the system under test.

Ken Vaughn: For encoding languages the tester needs to have a basic understanding of XML, which is the general market language used for all data exchanges using the C2C RI. And it's defined as the “eXtensible Markup Language” -- is what the “XML” acronym stands for. He also needs to be familiar with the XSD, which is the XML Schema Definition. This is the specific XML variant used to describe the allowed structure of other XML documents, which is used by the TMDD. So, in other words, your standardized structures defined in the TMDD use this variance of XML called XSD, and you need to be able to read that variant in order to fully understand the XML produced by TMDD systems. Finally, there’s WSDL, which the “Web Services Description Language.” This is the specific XML variant used to document interfaces between systems. We’ve mentioned before in those upper-layer diagrams that the owner center advertises the services it provides through a WSDL document. This is an XML document in a precise format, and the tester needs to be able to understand and read that format. When we say a basic understanding, we mean the tester needs to be familiar with the basic structures defined by the standards and capable of developing and inspecting packets and/or documents as needed.

Ken Vaughn: The tester also needs to have a basic understanding of several different protocols, including SOAP if you use it. This is the “Simple Object Access Protocol.” This technology is a specific markup language that is used to exchange ITS messages. Also, the Hypertext Transfer Protocol and/or File Transfer Protocol, as used, within the deployment—both of these are options. These are two alternative mechanisms used to exchange the XML documents produced by SOAP. And then TCP/IP networking; these are the Transport Control Protocol and the Internet Protocol. Each one of these the tester would need to be able to understand the basics of it and understand how these packages are exchanged.

Ken Vaughn: The tester also needs basic networking skills. He needs basic understanding and permissions for configuring the Windows environment. Specifically, he needs to be able to configure the internet security to authenticate users and encrypt communications as appropriate, as designed for this particular interface. And, of course, the user also needs the necessary permissions of the C2C host to configure the system correctly.

Ken Vaughn: The tester also needs ITS standards skills. He needs a thorough understanding of the information layer standards to be tested, mainly the TMDD version 3.03c. The TMDD defines the information to be exchanged. 2306 at the application layer, they also needs to be very familiar with; this defines how to exchange the information across the standardized internet link. The TMDD defines the ITS information, and NTCIP 2306 defines how you transport that information. By a thorough understanding, we mean that the user not only has a basic understanding, but the user also understands how conformance is defined within these standards. We can define when something is non-conformant as opposed to simply not supporting a feature. He also needs to understand the difference between conformance to the minimal standard requirements and compliance to project-specific requirements. So something may be optional within the standard, but he needs to be able to read the project specifications to understand that something may be required by a project even though it’s optional within the standard. Next, he needs to understand the traceability principles, and is sufficiently familiar with the standards to be able to quickly reference any and all detailed requirements and design. This is because as problems arise during testing, you need to be able to quickly identify the specific clause in the standard to which the problem relates, and then be able to identify what is at fault: Is it the system under test? Is it the C2C RI itself? Did the user enter some information incorrectly? Is the standard ambiguous? All of these are potential issues. The tester needs to be able to quickly reference those detailed clauses within the standard to decide who is to blame.

Ken Vaughn: Further, the tester also needs to have experience or training in testing systems. This includes an understanding of what test documents need to be prepared in advance, during, and after the test. He also needs to know how to use a Needs to Requirements Traceability Matrix as well as a Requirements (to design) Traceability Matrix, and a Requirements to Test Case Traceability Matrix. All three of those matrices come into key play as you try to resolve issues that have been identified during testing.

Ken Vaughn: If there's a need to extend the functionality of the C2C RI to support additional tests—for example, if you want to test transit information, or if you want to support custom project interfaces—the user will also need to have at least a basic understanding of how to develop user needs and define them within the C2C RI; how to write requirements and define them within the RI; how to develop the C2C test script file; how to structure all of this information into an overall test suite package; and then to provide a detailed process for this. The detailed process for this is defined in the user manual. We don't go through it in detail within this online course because it is an advanced feature, but the process is defined in the user manual.

Ken Vaughn: Finally, the tester needs to have a knowledge of the system under test. In particular, the user needs to be able to understand the custom configuration for each system, and to understand how to configure each system and operate its user interface. He also needs to understand the user needs and requirements of the system, and some of the test cases will require user interactions and configuration. The user will need the necessary permissions to operate the system under test.

Ken Vaughn: Well, that provides an overview of the skillset summary. It includes encoding languages, protocols, Windows networking, ITS standards, testing experience, perhaps scripting languages, and the system under test. It is an extensive list which requires an expert in the field.

Ken Vaughn: We mentioned that the C2C RI verifies conformance to the TMDD and NTCIP 2306. The software is designed to work over IPv4, but it does not include any test cases to verify the IPv4 implementation. The RI can emulate either side of the connection: the owner center or the external center. However, users should be aware that emulation is not simulation. In other words, the C2C RI does not mimic the functionality of an owner center or external center. It merely follows a script with pre-programmed values. In other words, do not expect the C2C RI to functionally respond to the ITS information contained within a message. The user must manually provide the data to be transmitted, and the tool merely ensures that the messages received are valid and responds per a set of defined test procedures. If the user provides wrong information to the C2C RI, it will likely incorrectly report an error. This is one reason why the tester must be so skilled in so many different technologies. Test scripts provided with the C2C RI check both positive—i.e., valid conditions—and negative, or invalid, conditions. Positive testing ensures the system under test works as expected when conditions are normal. By comparison, negative testing ensures the system under test works as expected when errors occur. When deploying a new system, it may be tempting to simply connect the two components together—in this case, an owner center and an external center—and see if they work, especially if the deployment is behind schedule. However, this approach will fail to properly check a wide range of conditions that can appear in a real-world environment. Connecting two systems to each other generally only provides positive testing. In other words, both flag systems are designed to work properly, and each test tends to be a simple exchange that may not reflect the complexity of live operations over time. Live operations can reveal bugs in the software, which affect how data is exchanged. The C2C RI cannot test every scenario, but it can test a wide variety of invalid messages and ensure that a system recovers when these errors are identified correctly. This is a major difference between the test software, such as the C2C RI, and the real system. The C2C RI allows investigation of anomalous conditions on-demand, and then assists the developers in debugging the problem. If these tests are not conducted during the verification stage, the deployed system is likely to experience more anomalies, especially under unusual conditions when the performance of the system is most critical.

Ken Vaughn: For example, in the case of a system that claims to be an NTCIP 2306 conforming owner center, the RI will verify if the system properly advertises its Web services through a valid WSDL file, it conforms to 2306, and that this file advertises the services that the RI has been configured to expect.

Ken Vaughn: It will also verify the system is able to connect through the advertised services, which may be any of the SOAP HTTP, XML HTTP, and/or XML over FTP. Likewise, these services may be available in a request/reply exchange pattern, a subscribed publish exchange pattern, or through a one-way publication exchange pattern. There are lots of different possibilities that may be a part of the requirements that are used infrequently in a real system. This is part of the advantage of testing with the C2C RI. It will make sure that each of these various options are tested. The project specific Needs to Requirements Traceability Matrix should specify exactly which of these variations should be supported by the system and ensure that each one of those can then be tested. That Needs to Requirements Traceability Matrix information is input into the C2C RIs as part of the configuration process, and the C2C RI then ensures that only applicable tests are performed.

Ken Vaughn: The reference implementation also can verify conformance to the TMDD. Specifically, it will verify that the system will support valid requests such as an inventory of message signs, while also verifying the system properly responds to invalid requests, such as ones containing a number different types of errors—like missing fields and requests, or invalid values, etc. When the C2C RI acts as an owner center, it tests the connected external center in a similar fashion by sending valid or invalid messages based on the test case selected.

Ken Vaughn: So what are the planned enhancements for the C2C RI? When in owner center mode, the C2C RI is able to respond to external center requests such as those asking for device inventory, device status, detailed information about a device or system, or controlling a device. In Version 1, the reference implementation responds to request messages by simply sending a preconfigured static response message. It is defined within the configuration file. In Version 2, additional logic is being added to this feature so that it will allow the RI to consider the contents of the received request. For example, the request for a DMS inventory may contain a geographic filter. In other words, instead of asking for all of your message signs, I'm interested in all of your message signs within a geographic region. In Version 1, the RI will respond with the same response without considering the filter in the request. In Version 2, the RI will apply the filter to the preconfigured information and respond with a more accurate message that would only contain records of those DMS that are within the defined geographical area.

Ken Vaughn: In another enhancement, Version 2 will also add SAFETEA-LU Section 1201 compliance reports. SAFETEA-LU is the Safe Accountable Flexible Efficiency Transportation Equity Act, a Legacy for Users. Section 1201 of this Act ensures agencies offer real-time monitoring and sharing of traffic and travel conditions of major highways across the U.S. In Version 1, the system identifies whether each standard requirement passed or failed. Version 2 will provide an additional report that also indicates how these results line up with a Section 1201 requirements. The report still references requirements contained in the standard, but it also references Section 1201 requirements known as Data Exchange Format Specification requirements as defined in the U.S. Code of Federal Regulations Title 23, Part 511. Finally, the report shows whether each of these items passed, failed, or simply was not tested.

Ken Vaughn: That brings us to our second pop quiz.

Ken Vaughn: What skill is not needed to use the C2C RI? So which of the following are not needed to use the tool? Answer choices are A) Windows networking; B) X.509 certificates; C) HTTP; and D) WSDL. So go ahead and make your choice. What skill is not needed to use the C2C RI?

Ken Vaughn: So the correct answer is B) X.509 certificates. The user does not necessarily need to understand how X.509 certificates work, as they are not required within the standards. Windows networking is a requirement. The user needs to be able to configure the RI host to support incoming messages. Likewise, HTTP: the user needs to be aware of how HTTP is often used as the underlying protocol for NTSIP 2306. And WSDL: the user needs to be familiar with WDSL, as this is how the interface of the system is described. While we've recognized the purpose of the C2C RI Reference Implementation, we've also acknowledged the key capabilities and limitations of that tool.

Ken Vaughn: Now, we're going to follow the process for producing test documentation that relies upon the C2C RI.

Ken Vaughn: Looking at the top of the slide, we can see the order in which test documentation is generally produced. This is defined within the IEEE 829-2008 standard. A test plan is a high-level document that mainly considers the administrative aspects of testing. The test design specification identifies the specific features to be tested and maps these to specific test cases. Test cases define specific inputs and outputs that are expected when following the defined procedures. Likewise, the test procedures define the process used to perform each test. Finally, after execution, the test results are recorded in logs and instant reports that are summarized in a test summary. The first four documents are addressed in this learning objective. The execution and the reports are addressed in the next objective. More information about this process is covered in a variety of other modules, including T101: Introduction to Standards Testing; T201: How to Write a Test Plan; T202: Test Design, Cases, and Procedures; T203: How to Develop Tests Cases; T204: How to Develop Test Procedures; and T321: Applying your Test Plan to the TMDD. Each one of those modules will provide additional information, but we'll go through a summary of key information about how it relates to the C2C RI in the following slides.

Ken Vaughn: The test plan is a high-level administrative report that provides context to system testing. The project may include a Master Test Plan. This provides a context of how multiple Level Test Plans fit together within the project. Each Level Test Plan provides a plan for testing one aspect of system requirements. For example, a test plan that will focus on conformance to the TMDD and NTCIP 2306 standards. The Level Test Plan will provide the context of how the testing fits into the overall project, and will address project management issues such as schedule and budget. The document will also include references to the test design.

Ken Vaughn: For example, test plans that use the C2C RI will generally relate to test plans that focus on communications testing. In other words, the C2C RI verifies communication details. It generally does not include logic to check functionality or environmental issues. For example, the C2C RI ensures that the data are properly exchanged; it does not ensure that a device control request is actually implemented. The C2C RI is typically used for laboratory environments. The nature of a C2C RI is that it emulates a center, and therefore implies an artificial environment. It is seldom used for onsite testing other than for connecting to an artificial—i.e., that's a test system itself—C2C RI. There's also verification testing. The C2C RI tests do not prompt for user comments related to whether needs are met. It merely checks to see if the requirements of the standards are met. Further, as a C2C RI emulates its center, it implies testing the component of the system under test, not integration of two real systems. Most often, this sort of testing will be used just prior to accepting payment of an item—perhaps not final acceptance, but certainly some sort of payments or acceptance—in order to move to the following stage. As it is still a laboratory environment, it typically is not considered a deployment test. Currently, the tool only checks for TMDD over NTCIP 2306. Other standards would either require customization or other tools. This does not prohibit its use during other types of testing. We are merely indicating where it's most likely, most commonly to be used.

Ken Vaughn: Module T321: Applying Your Test Plan to the TMDD Standard provided the indicated graphic as a typical test environment, but it was developed before the C2C RI was made publicly available. It assumes the test environment would include a sniffer that captured the communications between two real-world deployed systems. The sniffer would monitor those communications and analyze the data that was being exchanged between the two systems. That captured information would then be verified against the standards. While this process works in theory, it's a very, very labor-intensive process. The C2C RI changes this environment. The C2C RI replaces one of the two centers and largely eliminates the need for that external sniffer. This simplifies the test environment and also allows the test to be conducted over secured link while automating many of the checks, as well as verifying the system under test properly responds to invalid conditions. The C2C RI does not act as a sniffer. It acts as one of the two end systems.

Ken Vaughn: However, it is still important to develop a complete test plan before using the C2C RI. The test plan is more of a management plan than a test detail document. The recommended outline for the test plan is provided in the Student Supplement and detail.

Ken Vaughn: The test plan provides context to the test design specification. The test design specification details the specific requirements to be verified, and identifies its cases for each of the requirements—i.e. it verifies proper handling of an invalid header. Test cases and procedures are often defined in detail later. In the case of the C2C RI, these already exist. Using the C2C RI only requires that the user select the user needs and requirements according to the NRTM, or Needs Requirements Traceability Matrix. The tool will then automatically select these test cases that trace to the selected requirements. The test configuration report identifies the test cases that are mapped to each selected requirement.

Ken Vaughn: For example, in the test plan, this section is intended to provide context. In other words, it is intended to distinguish how this testing is different from—and complementary to—other test plans in the overall testing framework. The test plan may list out all features, or it may provide a higher level overview with pointers to specific test designs. For example, it may indicate TMDD interface will be verified in a laboratory environment for the purpose of approving Stage 1 payment. In each test design, the section is intended to identify specific features that will tested by the design. The design will also identify the test cases used to verify the identified features. Note that a single requirement may be verified through the use of multiple test cases, and a single test case may be used to verify multiple requirements. It is a many-to-many relationship. For example, the test design will identify the need to verity SOAP message encodings and connection active requests. The test plan and the test design specification are sometimes combined into a single document that includes a Requirements Test Case Traceability table to provide the mapping that is often included in test design reports, while reducing the number of documents for a project.

Ken Vaughn: This graphic shows an example of the test design specification that was used to develop the C2C RI that identifies the identifier of each user need defined in a TMDD. It also maps those to the test cases used to verify the user need. It then shows the requirements tested in each test case. This information can also be shown in different orders, if desired—i.e., user needs to requirements and then to test cases. While the complete list of tests requires several hundred pages, all of their information is built to the C2C RI so it's all handled automatically.

Ken Vaughn: A complete traceability table is available on request. In Version 1, this table is built into the tool, and when the user selects a requirement, the associated test cases will appear as options to be performed. In Version 2, the user will also be able to generate the complete traceability report showing all of the needs and requirements selected in the test cases that trace to those.

Ken Vaughn: Each test design will identify what needs to be verified in more generally referenced multiple test cases. Each test case identifies one test scenario, including its objective, inputs, outputs and a reference to a test procedure. Inputs may be specific values or variables to be defined while performing the test. Outputs may be specific values or a definition of what constitutes a valid output for the defined inputs, which may include variables. The test procedures then define the specific process to be followed to perform the test.

Ken Vaughn: For example, the test design may identify the requirement to process SOAP messages. It will also point to the test cases related to the requirement. For example, a test for properly handling a valid message and various invalid messages. Each test case will identify inputs. These may be specific values—perhaps stored in a file in this case—or merely a description of what the input is and allow the tester to provide the data later. Outputs will also be specified in a similar manner. Finally, there will be a reference to a test procedure.

Ken Vaughn: This is an example of the most relevant parts of one test case specification. It identifies the name of the test case and provides its description. It also defines the inputs and outputs for the test case, and a reference to the procedures used to implement the test case.

Ken Vaughn: The description of each test case is available to the user of the C2C RI. A complete example of what the test case looks like is in the Student Supplement, and a complete listing is available on request. With Version 1, the descriptions are available by hovering over a test case. In Version 2, the user will be able to produce this information in a report.

Ken Vaughn: As we mentioned, each test case will identify one test scenario, including its objective, inputs, outputs, and a reference to a test procedure. The test procedure will identify the exact step-by-step process to be used to check the system under test. Each test defines a specific action to be taken.

Ken Vaughn: For example, the test case may indicate that it will verify that valid messages will be accepted with defined inputs and outputs. The test procedure provides step by step details by giving instructions, such as configuring the test environment, verifying conditions, performing calculations, including conditional and/or looping logic, or sending messages, etc.

Ken Vaughn: This graphic shows a snippet of a sample test procedure that is implemented within the C2C RI. Each step is assigned a number for easy reference. Each step also includes the description, words in all uppercase, or keywords that indicate specific actions to be performed, and some steps are defined to generate pass/fail results. In these cases, the cause numbers of the specific standardized requirements are indicated.

Ken Vaughn: The Student Supplement contains a complete example of the snippet that was shown on the previous slide. The procedures are implemented using a script file. The scripts are defined in an XML format that will be described in the user manual. Version 1 of the software is able to print these test scripts. Version 2 will also allow printing of the test procedures—i.e., the more human readable form of the scripts.

Ken Vaughn: Well, that brings us to our third pop quiz.

Ken Vaughn: Which test document requires information from sources beyond the C2C RI? A) Test plan; B) Test design specification; C) Test cases; or D) Test procedures. So which of those documents requires information from sources beyond the C2C RI? It's not in the C2C RI itself, but we're looking for some other source in order to write this particular document. So go ahead and make your choice, and we'll review results.

Ken Vaughn: The correct answer is A) Test plan. The test plan is a high-level document and contains management information beyond the scope of what's contained in the C2C RI, whereas the test design specification is able to be produced from the C2C RI information. The C2C RI configuration report allows the user to map between the requirements and the associated test cases. Likewise, test cases themselves are part of the C2C RI. The configuration report also identifies the test script used for each test case. And then finally, the procedures—the C2C RI test script report—provides the test scripts as implemented within the C2C RI tool.

Ken Vaughn: While we've recognized the purpose of the C2C RI, we've acknowledged the key capabilities and limitations. We've also followed the process for producing test documentation.

Ken Vaughn: We'll now move on to recognizing the results a tester might produce after using the C2C RI. We'll talk about four key items here. One is understanding that the hands-on guidance is provided, explaining what should be included in results. We'll also explain how to interpret the test results and recognize limitations of those results.

Ken Vaughn: So first, looking at the benefits of the C2C RI. Without the C2C RI, each step of the procedure would have to be performed manually—a very time consuming process. The C2C RI automates this process, making testing simpler, more repeatable, and less prone to error. As a result, it decreases the costs because it requires less time to test in both labor hours and in schedule. This also allows the process to be more thorough, which improves the quality of the product and identifies bugs faster and earlier. Module T351 will discuss the process of actually using the C2C RI in detail. We'll assume the testing has been performed, and we will now look at the documentation that can be produced.

Ken Vaughn: Testing should conclude with the production of a variety of documents, including test logs, test anomaly reports, and test summaries. The C2C RI can assist in the production of these reports by using its built-in reporting feature. However, production of complete reports requires expertise and analysis of the tester. For example, the C2C RI produces a number of logs, but the tester may also log his own notes during testing that should also be recorded in the final set of logs. The C2C RI does not produce an anomaly report, but its logs and summaries will allow a skilled tester to capture the information needed for those reports. Finally, the C2C RI provides summaries of the raw results, but a complete test summary should include an assessment of the results and recommendations for future action. These are items beyond the scope of the C2C RI. Nonetheless, it's important to understand all of the logs that are available.

Ken Vaughn: Test logs are intended to capture the details of a test run in chronological order. This assists the tester in repeating the test to determine if problems are repeatable—i.e., occur in the same order every time. Repeatable problems indicate flawed logic at some step, and the log can assist the user in isolating the problem. Some of these problems may occur over multiple test cases, and that log identifies which test case was performed after which test case in which branching logic within the code was provided. The log provided exactly what occurred and is very useful for being able to repeat those tests. However, if it is a sporadic problem, it indicates a deeper problem in the system—such as a memory leak, a lack of resources, or other issues—and the log can assist the reader in identifying conditions where these problems are more likely to arise so the problem can eventually be isolated. So even in those cases, the log provides benefits.

Ken Vaughn: The C2C RI provides three distinct test log reports. The Test Case Detail Log Report provides step-by-step logging; the Test Script Action Log Report provides logs at the start and end of each test script; and the Message Detail Report provides a log of all messages sent and/or received by the C2C RI through the C2C interface. Each of these will be investigated in detail.

Ken Vaughn: The Test Case Detail Log provides a description of each test step logged in order of execution. It also shows the results from calculations and values that are stored in memory, whether a step passed or failed, a summary of why the test case failed, a timestamp for each test step, and the test script being executed.

Ken Vaughn: The graphic shows an example of a Test Case Detail Log Report. As described on the previous slide, the body of the report provides a timestamp, step-by-step explanation as to what was performed during the test and whether the test step passed or failed, along with any reason for failure. The top of the report provides background information on the test run.

Ken Vaughn: The Test Script Action Log is primarily a tool for debugging scripts. It includes a chronological record of the starting and ending of each test script, whether the script was completed successfully or was terminated early for some reason, file names and line number references to find any problem, and timestamps.

Ken Vaughn: This graphic shows an example of that Test Script Action Log Report as described in the previous slide. The report shows a timestamp and action associated with each test script action, providing an associated line number and column. It also indicates the results of the action as appropriate. As with the Test Case Detail Log Report, the top of the report provides background information on the test run.

Ken Vaughn: The Message Detail Report itemizes the messages sent between two centers. For each message, it records the type of message, the source of the message, the destination of the message, the contents of the message, and the timestamp.

Ken Vaughn: Once again, the top of the report provides background information as shown here.

Ken Vaughn: All test logs can capture the full detail of testing. It is often difficult to see the forest for all of the trees. It's difficult to see the overall summary of the testing results for all of the details provided in the logs. Test summaries are intended to summarize the test results and evaluate the overall conformance and compliance of the system under test, noting any significant issues that arose during the testing process. The summary should also make recommendations. Just because a test revealed a technical failure does not necessarily mean that the system is unable to provide any benefit. Some failures may reflect minor issues that will seldom, if ever, occur, while others may be much more fundamental. The summaries should evaluate such failures and provide guidance as to what impact it might have on operations.

Ken Vaughn: The C2C RI provides three reports that may assist in producing the final test summary, but they are not complete IEEE 829 test summary reports by themselves. These are the Test Case Summary report, the Test Message Summary report, and the Test Conformance/Compliance report. In addition, Version 2, as we mentioned before, will also support the Section 1201 Conformance/Compliance report.

Ken Vaughn: A Test Case Summary includes a high-level overview of the results of each performance of the test cases in the order they were performed. The results list each test case that was executed in chorological order, indicate whether each performance of a test case passed or failed, and provide a timestamp for each test case performance so that the associated test case detail report may be easier to locate.

Ken Vaughn: This graphic shows an example of what that Test Case Summary report looks like. The top of the report provides background information on the test run, and the detailed section provides a timestamp, test case, name, and result of the run.

Ken Vaughn: The Test Message Summary report summarizes the messages sent between the two centers. It records a simple description of the message, the source and destination of the message, and the timestamp of the message.

Ken Vaughn: This graphic shows an example of the Test Message Summary report showing the fields described, and the top of the report provides background information.

Ken Vaughn: The Test Case Compliance report uses the built in traceability tables that we discussed earlier to trace the overall test results back to the associated requirements and user needs. One test case may be associated with multiple requirements, each of which may be associated with multiple user needs. Thus, a single failure may impact several requirements and even more user needs. This report allows the reader to recognize the associations and then assess a potential impact of each failure. However, the user should remember that some failures may be due to configuration or tester errors. The tester will need evaluate each reported failure to determine what the real-world impacts might be.

Ken Vaughn: This graphic shows an example of the Test Conformance/Compliance report. The top of the report provides background information—as shown before—and the body of the report shown here indicates the requirements associated with each user need, and whether or not the test passed. When a test fails, a summary of the error is provided, and the information is highlighted to standout. However, as mentioned on the previous slide, any failure will require further inspection.

Ken Vaughn: Finally, the Section 1201 Conformance/Compliance report uses built-in traceability tables that we discussed earlier to trace the overall test back to Section 1201 requirements, as well as the standardized requirements and user needs. This report allows the reader to recognize the associations and then assess the potential impact of each failure. Once again, users need to be aware that failures may be due to user errors as well as system under test errors.

Ken Vaughn: This graphic shows an example of the Section 1201 Conformance/Compliance report. Once again, the top of the report provides background information, and the body of the report is shown here. You see that we've added a column for the data exchange format specifications—DXFS—so that readers can quickly assess where Section 1201 issues may exist.

Ken Vaughn: The test summary reports produced by the C2C RI are not complete test summary reports—they are only summaries of log data. They help in digging through the more detailed logs to find issues of interest. They help in preparing an actual IEEE 829 test summary report. A tester will need to analyze information. The reports assist the user in screening out configuration and user errors and assessing the impacts of problems, as well as making recommendations.

Ken Vaughn: Results from the C2C RI are strictly algorithmic. If the user enters wrong information, a failure may be erroneously reported. Garbage in, garbage out. In addition, while the C2C RI indicates pass or fail and provides a summary of issues, it does not rate the severity of the discovered issues. The tester will need to investigate all reported failures and analyze what impact they might have on the final deployment. Some reported errors may be due to misconfiguration, while others may have no operational impact on the system. Ultimately, the person responsible for testing will need to consider these practical impacts against real-world, institutional issues. For example, if the Olympics start this weekend, we have to make the decision: Do we use the system as is or not? Once he analyzes those real-world, institutional issues against the practical impacts, he's able to prepare a recommendation as to what actions should be taken in the near-term.

Ken Vaughn: The C2C RI is one tool to assist in verifying systems conformance with NTCIP 2306 and/or the TMDD. It is not a complete solution for all testing. It does not test implied functionality related to the messages, the user interface, algorithms or reliability of the system. Some of these features could be verified at the same time while using the C2C RI, but other tests will likely need to be performed separately not when you see the C2C RI. The Master Test Plan should indicate how all of these issues are addressed in individual Level Test Plans. The one Level Test Plan that uses the C2C RI is what we've discussed here.

Ken Vaughn: That brings us to our final activity.

Ken Vaughn: Which is the best way to describe the C2C RI-generated test reports? A) The conformance report provides the final assessment of whether the implementation conforms to the communications interface; B) When combined together, the various reports produce all of the elements of the IEEE 829 test documentation; C) The reports assist the user in identifying potential issues for further analysis; or D) The reports fail to provide the information they were intended to provide. So which of those are the best way to describe the generated test reports? We'll give you a second to answer that.

Ken Vaughn: We'll take a look at the results. The correct answer is C. Issues identified within a report should be investigated to determine why a failure was reported. Answer A is incorrect because the conformance report produces a formulaic result, and it may produce erroneous results if an error was made in testing. B is also incorrect. A proper test summary includes analysis that requires manual review, not just automated tool testing. And finally, D is also incorrect. The reports provide useful information as intended, but require analysis before final conclusions are drawn.

Ken Vaughn: Well, that completes our four learning objectives. We just talked about recognizing the purpose of the C2C reference implementation; we acknowledged the key capabilities and limitations of the tool; we followed a process for producing test documentation that relies upon the C2C RI; and we recognized the results a tester might produce after using the C2C RI.

Ken Vaughn: This module completes the TMDD curriculum for managers. For those who wish to gain hands-on understanding about the C2C RI works, one additional module exists. We've already completed T321: Applying Your Test Plans to the TMDD Standard. T251 was this module: the Center-to-Center Reference Implementation introduction. The one remaining module is T351: Center-to-Center Reference Implementation—Applying the C2C Reference Implementation. That course is designed for those who want hands-on experience with the C2C RI.

Ken Vaughn: As mentioned previously, that module is T351. Its title is Applying the C2C Reference Implementation. The learning objectives for that module include learning how to install and configure the C2C RI on its host system; learning how to operate the C2C RI to perform tests on a system to be tested; learning how to retrieve the C2C RI results from a test; and learning how to prepare a report based on the C2C RI test results.

Ken Vaughn: Thank you for completing this module. Your feedback can be sent to the link below, and you can provide your thoughts or comments about the value of this training. Thank you for your time.