Module 54: Applying Your Text Plan to FMS, Field Management Station, Part 1

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'm 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're 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 you'll tell your 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 will look forward to hearing your comments and thank you again for participating. We hope you find this module helpful.

Raman Patel: T304: Applying Your Text Plan to FMS (Field Management Stations) – Part 1 is based on NTCIP 1210 Standard v01. It teaches us how to prepare a Test Plan for field devices.

Raman Patel: I'm Raman Patel. I'm currently at New York University's Tandon School of Engineering teaching urban infrastructure, ITS, and system engineering courses. Formerly, I was at New York City DOT as Chief of Systems Engineering for about 25 years or so. I've also been involved in the standards making process with the SDOs—including IT, AASHTO, and NEMA.

Raman Patel: For this module, we have four learning objectives. Learning Objective 1 describes the role of a Test Plan and testing to be implemented within the context of the system lifecycle. Learning Objective 2 shows us how to recognize the purpose, structure, and content of well-written test documentation. This learning objective relies on IEEE 829-2008 formats. Learning Objective 3 describes how to develop the complete test documentation package that we need for SSM—SSM stands for the Single System Master specification based on NTCIP 1210. Finally, our last learning objective describes how to test an SSM using a sample test document. Collectively, these four learning objectives will provide us with the skill set that we will need to complete the testing of the SSM.

Raman Patel: The first learning objective describes the role played by the Test Plan within the context of the lifecycle.

Raman Patel: What is the role of a Signal System Master—the SSM. We'll be using this terminology throughout the module. The SSM is part of a Field Master Station. A Field Master Station is a traffic controller in the field doing several things besides controlling traffic signals. The main function of the SSM is to coordinate the signal system locals. These local controllers are located at the intersections. They are sometimes referred to as Intersection Controllers since they control the signal timing at the intersection. The SSM supervises a bunch of SSLs on a particular segment of the network.

Raman Patel: The SSM is used within a typical physical architecture. Here on the left side, we show Traffic Management Systems and a field computer—a laptop—connected to a Signal System Master in the field to the right. The Signal System Master is connected to the Signal System Locals (SSLs). These local controllers are used in signal timing. The SSM is simply a device that supervises those SSLs.

Raman Patel: Testing is a process. It uses a Test Plan. We conduct testing using a Test Plan because we want to verify whether the SSM fulfills each requirement stated in the Test Plan. The testing process brings together requirements and verifications. As you can see here on the left side, we have a test setup where the device is being tested—in this case it's the SSM. On the right side, we show a laptop-based setup with a monitor. We have a Test Plan which we will use when we perform the testing to answer the question: Was the system built right? We will be looking for that answer at the end of the testing process.

Raman Patel: Testing Methods. There are several testing methods we use for conformance verification. When we say conformance, we are referring to conforming with the standard. This requires verification. Very often, we start the process with a visual observation of the device to see its functionalities. The second method is demonstration—we connect to the wiring. As you see in this image, we test the traffic controllers using demonstration. A more elaborate testing method is the analysis portion. We can hook up controllers with the signal system using other additional Test Procedures that we conduct—analyze the simulation. We can then assess whether the controller is coming through with the functionality or not. However, our interest is on the right side—this setup tells us there is a controller that needs to be tested using test documentation. Test documentation generally refers to a Test Plan.

Raman Patel: Here is the system lifecycle and testing to be performed during the system engineering process. Documentation preparation takes place during the high-level design and detailed design levels. This is when we prepare the Test Plan so we can use it later on when we conduct the testing procedure, as you see here on the right side of the Vee diagram. That's where the communication interface level has to take place. A device could be tested at the unique device level. Then we bring the systems together with the device in it, and so on. Eventually, this testing procedure will walk us through different stages of the systems engineering lifecycle process.

Raman Patel: Unit/Device Testing—sometimes it's referred to as a Bench Testing. SSM testing takes place in a lab or workshop environment using PC-based software. There are tests of SSM data elements and dialogs to check whether conformity has occurred or not. Our focus is to test one device at a time. Here, we have to test different sets of functionalities. How much do we test? Not everything can be tested. We have to focus on what to test and to prioritize tests. Our focus is generally on failure consequences—what could go wrong? That's what needs to be tested, and that's our focus because there are several issues at the boundary level about how the device is performing towards the end of its functionality. We will check all of those different things. We test the priority-based items based on failure consequences. One more thing to remember here is that we cannot test 100 percent during the timespan. We have to select key functionalities. This allows you to make sure your focus areas remain tied to the functionality that's important for your particular implementation using a field master. In this case, it's the SSM.

Raman Patel: A Subsystem Verification focuses on whether the system was built right. That's the kind of question we need to answer during the verification process. The SSM will be tested to ensure that the communication interface with the SSLs in the field is correct, and that we are able to communicate using the central system software. This will be the verification process attached to the SSM and its associated SSLs on the Interconnect. Several examples are shown here. What will we be checking? What will we be verifying? For example, 3.3.1.1 is “Accept Data” from the TMS. TMS stands for Traffic Management System. Here’s the SSL and SSM together as a subsystem. Do they really accept data from the Central Management Station? Is the TMS able to deliver data? Can we explore further functionality through the TMS? All of these things need to be verified because that’s how this kind of subsystem setup is used in the real world. Our focus remains on specific communication capabilities and how the SSM and SSLs work together.

Raman Patel: For the system as a whole, verification starts with the physical architecture. The entire system needs to be tested in terms of requirements. You can see here on the left, we have a Traffic Management System—TMS—with a Signal System Master in the middle. In between, you have NTCIP 1210 v01—the communication interface. Between the SSM and SSLs, there is also another standard—NTCIP 1202 v02—used to provide signal timing and other capabilities. We are not dealing with that standard in this module. We are focusing on NTCIP 1210 that connects to the Center Station and SSM.

Raman Patel: Here’s the validation of a layout that might help us to answer whether the right system is built. What we are verifying is what the user needs are and whether or not they are implemented. In other words, validating whether the user needs have been implemented properly. We can see on the left side a sample Traffic Management System—TMS—with the TMC setup generally recognizable. There’s an operator sitting at the desk and checking the field system on the right side, which has the SSLs connected over the network segments. Several segments are shown here. All of those SSMs at the intersections will be working with a particular SSM—Signal System Master. This will provide us with the connectivity we need to the central system.

Raman Patel: Our first activity here is to answer: Which is not part of the testing process in a system lifecycle? We have four choices: A) Test Planning; B) Preparation of test documentation; C) Test execution and reporting; and D) Identification of system requirements.

Raman Patel: The correct answer is D. It’s correct because identification of system requirements is not part of the testing process. We have already learned that in previous modules. Here, we are focusing on the testing part of the complete connectivity with the SSMs. Test Planning in answer A is incorrect because Test Planning is complete when the system requirements have begun. In other words, in the detail design or high-level design, we already have started planning for the test. Answer B is also similarly incorrect because documentation is created at the high level and the design level. Answer C is also incorrect because test execution and reporting are performed at each level of the testing workflow using this particular documentation we’ve been preparing.

Raman Patel: That brings us to our Learning Objective 2—Recognize the purpose, structure, and content of a well-written test documentation.

Raman Patel: Here, we’ll be using IEEE 829-2008 formats to prepare the structure of the whole documentation process.

Raman Patel: What are the objectives of the SSM testing documentation? Here are four key components. First is the outline of what to test. A test documentation has an outline that will allow us to input everything we need to test the complete set up. That’s number one. Number two is to state clearly how to test. This is a test of Test Planning documentation. It tells us how we're going to perform the actual test and clearly understand each step along the way. Third is to report the results and outcomes. After the test is conducted, we will report the results—such as what worked, what didn't work, what were the other outcomes, what failed, and what passed. Finally, we are focusing on IEEE 829-2008 formats. We will not be using our own format. We will follow a specific process using IEEE 829-2008. In this module, we are adopting and using IEEE standards just like all of the PCB—Professional Capacity Building—modules. The testing documentation will be conducted during the test process, but the objective remains for each of these four key components.

Raman Patel: What is available in the IEEE 829-2008 standard? It gives us the scope, starting with technical management. It provides an overview of how you're going to approach a particular item to test. Then you have resources needed—how are we going to perform the test—then schedule to complete. The Test Plan also identifies test items. What is to be tested? What is not to be tested? Those are the features we'll be covering in the Test Plan. The testing task—how we perform the testing in the test environment and everything else that goes with it. Certain risks require a contingency plan—if something goes wrong what are we going to do—will also be covered in a Test Plan.

Raman Patel: IEEE 829-2008 formats include the MTP—Master Test Plan. The MTP contains integrity level schemes and choices and the overall test processes, activities, and tasks. It also contains the test levels and documents we will use in the MTP. Under the MTP, you have a Unit Test Plan—LTP. The Unit Test Plan refers to a particular device level that we only ever discuss about with batch testing. That is the LTP kind of environment. A subsystem is an integration test. It has communication and tests the unit device with the subsystem so that we can see how the complete subsystem containing the unit device will be tested. Finally, we have system acceptance, which is a very high level of testing. The entire architecture will be tested, and since this is a very high level organization they will provide mode SSLs, SSMs, Central Traffic Management Systems, and communication in between, all of these areas will be covered. A Master Test Plan may not always be required. In this case, we are not asking for an MTP to be prepared. But in certain areas, an MTP will need to be prepared. As we will find out in SSLs and SSMs, there may be a choice in preparing a Master Test Plan.

Raman Patel: Let's look closely at the structure of the MTP. The first level of the Test Plan we show here is the Level Test Plan for the SSM Communication Interface. This is the focus of our discussion in this module—what we're actually testing based on the SSM. We have the SSM Unit Test Plan shown on the left block. In the middle, you have the SSM Subsystem Integration Test Plan. Finally, at the end is the SSM Acceptance Test Plan. That will be covered under the LTP for the SSM. Level Test Plan for SSLs. This is the intersection level where you have SSLs: signal controllers that control signal timing. And that we will have each usual intersection controller which will be a Unit Test for SSL. SSLs combine with the core activity with the interface to the SSL will be the Integration Test Plan. Communication is involved. Finally, we'll look at the entire system through an Integration Plan, and will also include the Traffic Management System at the central location.

Raman Patel: SSM Test Plan Structure. We are borrowing the structure from the IEEE 829-2008 formats. At the top, we show a diagram of the Test Plan. The Test Plan describes the overall approach to SSM testing. Underneath the Test Plan is the Test Design Unit Test. Test Design is another terminology we use from IEEE. The first is Test Plan—which is the master document. Test Design is a component. In our discussion today, it deals with the Unit Test. Test Design specifies the details of the test approach. What is to be tested? This is shown here as a Unit Test because our focus in today's example is the SSM. There could be similar designs for integration and acceptance. Let's walk through the Test Plan in more detail. Underneath Test Design, you have a third component—Test Case. A Test Case is a specification. A Test Case outlines a set of tests and inputs—what is going to drive the Test Case. Then you have an execution condition. There are certain conditions that we have to actually meet. What do we expect? What is the output? What do you think is going to happen? What are the outputs as the expected results? That will be covered in the Test Case. The last step is the Test Procedure. The Test Procedure is a specification, which defines the steps to be followed during the test. This is where we perform the entire process of what needs to be tested and describe how we're going to do it. All of those things will come out together. To summarize, a Test Plan contains a Test Design, Test Cases, and Test Procedures.

Raman Patel: When we prepare a Level Test Plan—LTP—using the IEEE 829-2008 format, we will provide an introduction that identifies the document scope and references general terms. We will also provide a sequence of what we will be doing first—Unit Tests, Test Cases, and Test Conditions. General introductions will be provided in the first section. You will have all the details about the Level Test Plan—test items and their identified files and Protocol Requirements Lists—we discussed PRLs in the previous module. There is also the RTCTM—Requirements Test Case Traceability Matrix—which is a component that will be used for testing Test Cases. You have features to be tested and not tested. We cannot test everything. There's no need to test certain functionality, and that will also be identified. Our focus is what needs to be tested. General test approach pass/fail criteria—whether the device has actually passed or failed, the system has passed or failed—that's described in the Level Test Plan. Suspension criteria—under what condition a test can no longer be conducted. There may also be an interruption of some kind. In that case, we will terminate the test. These are the conditions written down in the Test Plan. Test deliverables—what do you expect from these results and what are the deliverables. These are the main components we'll be covering in a Level Test Plan.

Raman Patel: General Management Issues. Who is going to do what? Resources and training may be required when you perform testing and involve other people. What are the risks and contingencies? What will you be doing in general? There are certain quality assurance procedures. Maybe there's some kind of glossary needed defining the terminology. All of these things make up a good Level Test Plan.

Raman Patel: Let's look at a sample Test Design outline based on the IEEE 829-2008 format. The Test Design contains the introduction, documentation identification, and will focus on Level Test Design. It describes feature to be tested; the approach to refinements, test identification, and pass/fail criteria; test deliverables; and a general discussion about the document's importance. What we see here in the Test Design is a Requirement ID—discussed in a previous module. A functional requirement begins the process—this is what is tested. You have a Requirement ID—3.4.2.2.1, as shown here. You have a requirement—which is Explore SSL data. You have a Test Case ID. Then you have a Test Case—what the title will look like. We have two ID procedures here—one is for the requirement and the other is for the Test Case. TC stands for Test Case. Test Case 3.4.3.1.6.1 verifies the maximum number of intersections. We'll be using this as an example of a Test Case. Features to be tested will come from a PRL—Protocol Requirement List. We discussed this in previous modules. This is how we're going to extract all our features for requirements to be tested.

Raman Patel: Here's an outline of a Test Case in a table. The Test Case has an ID, as you see on the top. The Test Case has an objective, which requirements will be verified. These include a test dialog with the correct sequence. A number of times we have repeated in previous modules that everything we test is also in the context of dialogs being conducted in the correct sequence, or correct structure and content of the data. All of these are very important. This is what the Test Case focuses on. Input varies. It is used as a variable for testing. The Test Case brings that together. There may be more than one input coming into the Test Case to conduct the process. What will be the outcome? Not only the results but also the errors. How is the device going to behave? What are the errors that we expect? We want the errors to be noticed. We want to make sure about what works and what doesn't work. Environmental Set Up—the Test Case setup outlines this. There are dependencies. Sometimes a Test Case must be executed prior to the other Test Cases. There may be several Test Cases to be performed, and they have to be coordinated with each other.

Raman Patel: Here's a sample outline of a Test Procedure. A Test Procedure has steps. In this example, we show six test steps for conducting the testing process. This is the last step in the testing process. As shown here, the first step is Configure. We're going to make sure that everything in the device is configured properly, then we will move to the next step. And somewhere here we will focus. For example, in step three we'll be looking at the Set Up to ensure that the pixels are functioning prior to the Test Case. This is an example from another standard to give you an idea. As we move along the steps and Test Procedures, things will begin to get more complex—the Test Procedure will have a number, title, results, and additional references related to a particular standard.

Raman Patel: Documentation for Test Reporting. This is also based on the IEEE 829-2008 standard. We'll need to make sure that our records are accurate when we execute a test process. That's the starting point, when we have to be aware of what we're going to actually document for reporting requirements. Make sure that we understand the process early on. Executing the Test Plan begins with that thinking. Then we have a Level Test Log. LTL—Level Test Log—is a chronological record date wise and time wise. What happened Monday? What happened Tuesday? What happened Wednesday? That's a chronological way of making sure that everything we are testing is actually recorded. Then we have an anomaly—what worked and what didn't work. That's our focus throughout these issues. The Anomaly Report will focus on events during the testing process—which may require further investigation. We find something and say this needs to be investigated further. That's what an Anomaly Report will focus on. The third one here on the right is a Level Interim Test Report. This report is a summary. As the test progresses, we summarize results and when they occur. This kind of interim report is sometimes necessary at the agency and developer level. The last one is a Level Test Report, which summarizes the results, evaluation, and recommendation—everything that happened. It gives you an general idea of what really happened. This is a very nice setup of preparing ourselves to actually document everything that's coming out of the testing process.

Raman Patel: Our second activity is to answer: Which is not included in a structure of a Test Plan: A) Test Logs; B) Test Design; C) Test Case with inputs/outputs; and D) Test Procedures with steps.

Raman Patel: The correct answer is A. It's correct because test logs are not part of the structure of the Test Plan. The Test Plan is performed very early at the planning and the design level. The test logs are actually performed after conducting the test itself. Test executions are performed and then we have test logs available for reporting in IEEE 829-2008 format. Test Design is incorrect because the statement is actually true. Test Design provides detail on what to test. Answer C is also incorrect because the statement Test Cases with inputs/outputs is true. Test Cases provide input and the Test Case also provide outputs. The statement is actually true. The last one—D—is also an incorrect answer because the statement is true. Test Procedures have steps—maybe five, six, seven, eight. You never know how many steps are required to conduct a test.

Raman Patel: That brings us to our third learning objective. Here, we will discuss how to develop a complete test documentation package based on NTCIP 1210 Standard version.

Raman Patel: For the SSM, this plan will be helpful and required for the testing process.

Raman Patel: Let's look at the key elements in preparing a Test Plan. The key elements begin with the User needs and Requirements. User Needs and Requirements are found in the PRL—Protocol Requirements List—as we discussed in the previous module A304A. That's the beginning of our preparation. The second level of preparation includes objects and dialogs. These are the Requirements Traceability Matrix—RTM—issues. The RTM contains a design object. It has also dialogs—generic dialogs and standardized dialogs. All of these provide us the information we need to prepare a Test Plan.

Raman Patel: Here is an example of a PRL. What does the PRL provide? The PRL identifies features to be tested. As you remember, the PRL has several columns—User Needs, User Need Title, Functional Requirements, Functional Requirement title. Then you have Conformance, which is the fifth column. In Conformance, we have M. M stands for mandatory. In the Support column, next to it, you have Yes and No. When we have completed the PRL, we have identified which requirements are mandatory and which are optional and selected by the users. All of this information identifies which features need to be tested. Everything that the PRL has selected will be required for testing this process or Test Plan.

Raman Patel: The RTM takes the process further. Identify provides us with the objects to be verified. Every object in the SSM has a particular structure. There is a syntax with a range. How many? It's a quantification process. We have integers. For example, the maximum number of intersection SSLs can go from 8 to 255. Nobody has 255—that's a very high figure. There will be some numerical value that will tell us—in our case, we can start with 8 and it could be 10. We'll find that out. The RTM gives us this information to verify. What are we going to verify from the range values? These examples taken from the A304b module show there are several objects. There is also a dialog. Through the requirements, the SSM intersection and dialogs and functionality will all come together and tell us what needs to be verified.

Raman Patel: Developing an SSM Test Plan begins with the Scope, Approach, General Resources, and Schedule. Then we move on to the specific items to be tested, features to be tested, what is not to be tested, testing tasks to be performed, personnel—identifying people who will actually be doing the testing. A certain risk associated with the Test Plan will be also used. Our approach here is to provide information features to be tested and not tested from the PRL. This is what will go into the SSM Test Plan.

Raman Patel: Test Design specifies the detailed approach. It identifies features to be tested by the Test Design, requirements to be tested by the Test Design, and Test Cases associated with the Test Design. This is a more coordinated approach to using PRL so we can prepare a Test Design.

Raman Patel: Test Cases are specific. Test Cases are identified by the Test Design specification. What are the input and output specifications? What will go in the Test Case, and what will come out of the Test

Case for output or results? That's the focus of the Test Case. Agency specifications will provide us the PRL, which will be used in a Test Case. The other part is the standard itself—NTCIP 1210 v01, the RTM, and the MIB. The RTM provides the data objects for input to the Test Case. The MIB has a value called data objects, given to us with the OID—Object ID number—the title of the object and the range and details that will be needed to verify a range of the object.

Raman Patel: Test Procedures is the last step in the Test Plan preparation. We have specific steps to execute in sequence. We cannot skip anything provided by the Test Procedures. We have to conduct these in the proper sequence. For example, agency specifications will provide the PRL, which will provide us with requirements which will go into the Test Procedures. Then we have the standard itself providing the RTM. The MIB will be also used to prepare Test Procedures.

Raman Patel: Test Cases. Here's an example of a Test Case for intersection unit control status. It has an ID on the left—Test Case 001. Objective is in the second row. Our objective is to verify system interface implementation (positive Test Case requirements) for a sequence of object identifiers. We're going to make sure this Test Case verifies whether the SSM is responding as anticipated or not. This is the focus of this particular Test Case. Inputs are on the left. We show in the text box, under the Input, integers—1, 2, 3, 4, 5, 6, all the way up to 8—are already assigned by the test objects by the standard. For example, if you look at 5.8.1.1.5 IntersectionUnitControlStatus to '1.' When we verify the value 1 to 8, we're going to make sure which is the correct value range implemented by the SSMs. For example, time-based coordination has a value of 6. We'll check that. Then you have an interconnect value of 7. We'll check that. We'll include all the control status—1 through 8 are included. That's a range we'll remember during the Test Case process outlined here. The outcomes are reported in terms of whether the Test Case has been carried out and correctly shows the boundary conditions. All of these numerical values that we just mentioned will be also be verified. For example, at the bottom we show 5.8.1.1.5 and 5.8.1.1.6. These are the design objects. They provide us with the intersection control status and log size. By using these objects, we can verify Test Case 001 in this example.

Raman Patel: Developing SSM RTCTM. RTCTM stands for Requirements to Test Case Traceability Matrix. It connects the Test Case to the requirements. We are now sharing the same requirements we use in the PRL and in the RTM—and using the RTCTM the same way. Every requirement has a unique ID with a title as you can see in the first two columns. In the third column, we have Test Case ID, then Test Case, Test Procedure ID, and Test Procedure Title. Here, we have a complete set of items that we need to use during the Test Procedure, but we are actually connecting them with each other so there is traceability from requirements to Test Case. That's the first part we show in the text box. It's very important because the Test Case has to connect what needs to be tested and what is not to be tested. That's the requirement issue. Then you have a Test Case that is now finally a test with this procedure. For every Test Case, a certain number of Test Procedures will be used to carry out each Test Case. That could be one Test Case and many Test Procedure steps, or several Test Procedures broken up in pieces. And then we can verify this. In this example about the requirement for exploring SSL data by Traffic Management System, you have a Test Case 3.1.6.1 to verify the maximum number of intersections. It will trace to the Test Procedure 3.4.3.1.6.1 shown here. This will verify objects ranging from 8 to 40. It starts with 8, which is the minimum value we will be checking. Forty is the maximum value in the object range. That's the function of the RTCTM.

Raman Patel: The RTCTM lists Test Procedures for each Test Case. The RTCTM has one or more Test Cases to verify conformance. The RTCTM also lists one or more Test Procedures. It depends on how complex the Test Plan is and your situation. Generally, this is how procedures will be attached to each Test Case.

Raman Patel: Test Case and Test Procedures. In this example, you can look at the previous modules for details. For example, in T204 Parts 1 and 2 we show how to perform Test Procedures and preparations in conjunction with the Test Cases. Here we show an example of Test Case TC 1.1 to verify the maximum number of SSMs that can be set. A set is a process used by the Central Station. Variables for what goes in the Test Case—maximum number of SSMs of one, less than one, or more than one. If you have a certain number of SSMs, we will test that maximum number, then we'll test one less and one more. That will give us an idea of whether our Test Procedure is going to do its job and bring us the results that we expect or not. In the third row, we show pass/fail criteria. What will make the device under test—DUT stands for "device under test." What will make the SSM accept data or maximum SSMs? How are we going to make sure that the SSM conforms to these different values? That's what the Test Procedure will check through the Test Case. Here's a Test Procedure. There are four steps to carry out the previous expectation in Test Case 1.1. We will use four Test Procedures here. These are steps—not Test Procedures themselves. There is one Test Procedure with four steps. The first step will be to configure a set to a maximum number of intersections for a maximum number of SSMs attached to a particular intersection setup. This is the device setup. We have chosen a minimum of 8, so that will show up as a maximum of what we are going to start our process with. In Step 2, we will take one more SSM. We expect a one-to-one relationship—in other words, our expected results will be 1. If the Test Procedure is to check one—whether or not one is a valid number—the response will have to be 1. Similarly, if the SSM is 2, then we expect an answer of 2. For the last step of the SSM, the number of SSMs is 10. The response is going to be wrong. We want to make sure that it is 8 not 10 because we have selected 8. Earlier, we discussed the maximum number of SSMs as 8. When we check the controller using the range of 10, the controllers should provide with a correct result of 8, not 10. This procedure is established so the error will come up. If the error shows up in this process, we are that far ahead.

Raman Patel: There are certain ways that we can produce Test Procedures. This particular standard does not have a Test Procedure built into it, so we'll have to develop this at the agency level. There are certain tools. One is the Test Procedure Generator—TPG—which is a software that guides us in developing Test Procedures. It's used for center-to-field devices—such as SSMs and other devices in the NTCIP family. This is a relatively new product. It's at version 2 now, and can be downloaded from the website listed here. Make sure you have version 2.1 if you are preparing Test Procedures.

Raman Patel: How to use the Test Procedure Generator—TPG. Software will be downloaded from the website to install on your computing platform. Make sure that the TPG imports the standard—in our case, it's NTCIP 1210. The SSM standard will be imported to the Word file—that's important to know—then the Word file goes into the TPG. The standard provides the requirements, objects, dialogs, and the RTM. All of these items are available from a particular standard for each device. In our case, we're talking about the SSM. That will be used by the TPG to produce the Test Procedure, in the sense that the interface is provided by the TPG. This Test Procedure feature will allow the user to begin the process of developing Test Procedures. Developing Test Procedures is covered in detail in the next slide. Let's just look at the next slide.

Raman Patel: The Test Procedure with TPG defines the title, description, and pass/fail criteria—we discussed earlier what should be in the documentation requirement to be tested. The TPG has imported these items from the standard. The variables are from the standard's RTM. This is the kind of input that will be used by the TPG. The TPG can also be used to produce detailed steps—its role is to help agencies create interfaces and hopefully to develop the procedures they need.

Raman Patel: There are several benefits to TPG. Reduces the development risk, effort, and the cost. Without the software you will have to do this manually, that is time consuming and has a lot of risk because you don't know what you missed or what the standard intended to do for conformance. This also has a cost. In general, all of these will be helpful in reducing the extent of the task and the risk that we create with a software tool like TPG. For example, we'll be able to minimize these kinds of issues. Ensures traceability and conformance. The whole purpose of testing is to make sure that we are conforming to the standard. That's what the purpose of TPG is—to establish traceability and conformance to a particular standard. It also determines compliance with the extended standard. Sometimes, if you have proprietary or vendor-specific solutions, you also want to test these. TPG will be helpful in this area as well. It promotes interoperability. If you test one device using TPG, you will be also able to do something similar for the other devices and so on. TPG will also help with these different interoperable issues. TPG also creates in-house expertise. Once you know how to use these kinds of software tools, it will be helpful in performing many different activities relating to different devices within the organization.

Raman Patel: Our activity now is here to answer: What is the primary purpose of RTCTM? A) Sets the testing workflow sequence; B) Correlates User Needs to Requirements; C) Contains only Test Cases; and D) Traces Requirement to Test Case to Test Procedure.

Raman Patel: The correct answer is D, because it takes us from test Requirements—what to be tested, what is not to be tested—to the Test Cases as input and output. Test Procedures actually depict conducting a real test. Everything comes through the RTCTM. A is incorrect because the testing workflow is part of the Level Test Plans. B is also incorrect because the user needs to the Requirements Traceability Matrix are part of the PRL. C is incorrect because it contains Test Cases and Test Procedures for each different Test Case.

Raman Patel: Our last objective here is to describe testing an SSM using a sample test document.

Raman Patel: Let's walk through a sample testing document and see how we can bring together all these different areas that we have just discussed.

Raman Patel: Where is the Test Plan located? It's a good question to ask because the Test Plan is part of the general procurement contract procurement. In the contract specification for the communication interface, you have general information. Then you have SSM User Needs, which is a primary step to start with. Then you have Functional Requirements—these are attached to the SSM functionality describing what the SSM does and which requirements are outlined in the PRL. The PRL and RTM are project specific. That's the fourth area. Finally, the test documentation is generated using all of these different components.

Raman Patel: SSM testing setup begins with the Test Plan. What to test? What to test verifies each feature and requirements. Our focus is to make sure that what we test remains foremost in our understanding. With that in mind, we have a test setup. This includes a device under test—in this case it's

SSM. You have a laptop shown here with the communication interface and then a test analyzer, which could be another computer—or the same computer, but it provides you with analyses, results, and other items that we are expecting. We'll document the output of this whole process—results and outcomes—as long as we are in the process itself, but we also will prepare a final test report at the end. This is important in understanding a typical testing set up.

Raman Patel: How is the SSM Test Plan developed? We'll be using the IEEE format, which has a Test Plan and a component beginning with the Test Design for the unique test. In this case it's the SSM Test Case and the Test Procedures. The Test Design and Test Plan can be combined in one document, but they could also be separate. There's no specific way of developing a Test Plan—it depends on each agency's particular requirements. Test Cases and Test Procedures can be also combined into one document.

Raman Patel: The Test Plan outlines key parts. Section 1: Introduction. Section 2 has Unit Testing containing subsections that identify and outline the test items to be included in the Test Plan. Each has a specific ID number we can keep track of during the entire process. The RTCTM also connects the Test Design and Test Procedures. We have SSM features to be tested from the PRL. Also objects and their ranges from the RTM. These are the design objects that implement the particular requirement. Then you have a general approach and item pass/fail criteria that should be included so we can measure how to pass or fail. This criteria will be in front of us throughout the testing process. Suspension criteria—what does the criteria look like when terminating a test or resuming a test later. All of these different items are outlined ahead of time and not after the testing. The outline has to be developed during the Test Plan preparation.

Raman Patel: Key Parts. Test Deliverables will always require a Test Plan, a Test Case, a Test Design, and Test Procedures before the test begins. Early on, we have to prepare all of these different items. For reporting during and after the testing, we use test logs—chronologically prepared. Test incident reports are now called test anomaly reports. You also have Interim Test Case status reports. Sometimes agencies want to know what really happened. We provide them with these interim reports that are not complete. As we go along, these reports help us to understand more about what we are testing. The final report that combines all these different areas and outlines all our activities during the testing process.

Raman Patel: Let's look at the case study.

Raman Patel: Here is the city of Midsize SSM communication interface specification. The Central TMS has a requirement of 1210 v01. It has a response time of 600 milliseconds. It also says that one SSM will control/monitor 10 location intersections. The responsive strategy covers 30 SSLs out there in the network. The existing communication interconnect is adequate—there are no issues. The project PRL and RTM are also included. Both are provided by the agency.

Raman Patel: What are the project parameters that we have to outline? For example, here we say that there are 10 SSLs. Going in, we will be testing 10 SSLs for each SSM. One SSM coordinates 10 SSLs. Then you have a total of 30 SSLs, and there are 3 SSMs—one for each segment. So we have 3 SSMs we will be testing in this Test Case, and there are 3 sections as shown here.

Raman Patel: The PRL provided by the agency outlines what needs to be tested and not tested. That's what the PRL is good for—for the agency and for the people who will be using the PRL in the Test Plan.

You have 3.4.2 manage the SSM configuration. We have a number of design requirements circled in this box. They have been selected "Yes" because Mandatory—M—is required by the standard. All Ms are selected—these will be tested. The agency has also selected Configure timer reference. Even though it is Optional—O—the agency selected "Yes." That will be also included in what will be tested. The next to last requirement determines cycle timer capability. The agency has selected No, so this will not be tested. Likewise, the last requirement—Determine SSM software version, even though it's Mandatory, there is only one version 01. Perhaps the agency has taken that into account and decided this will not be tested.

Raman Patel: Find Object Ranges from Project RTM. We have just seen the PRL. Now, we will look at the RTM. The RTM provides data objects. One of these is very specific—it says maximum intersection. Here we have a range from 8 to 255. That's the number of intersections provided by the standard. We don't have 255—agencies generally start with 8—a small number. Here by default we will start with 8 and go to some other number. The range will be 8 to X-number. In our case study, we have 10 SSLs. There are 10 specific ranges here in our test study—not 8 or 255, but 10.

Raman Patel: This is how we will proceed in the RTCTM. We have a verified maximum number of intersections and have developed a Test Case to verify the object range. This Test Case has certain Test Procedures attached to it. Our focus is 10 intersections, shown here by circling, because that's what we have: 10 SSLs. The Test Procedure we will perform will check the boundary condition. The maximum is 10, so we will go slightly below to 8, and then go slightly above to 12. We start off with 10; At the bottom we have 8; At the top we have 10. This means 8, 10, and 12 will be the range of values we check.

Raman Patel: In general, with boundary conditions, we have to make sure that we test above the limit, below the limit, and at exactly the limit. These are the three ways to carry out this process. The boundaries vary. If the process is successful, we will respond accordingly. We will be looking for results according to the boundary condition we just determined. If there are errors, we will notice them because we already know to expect. It's going to show up as 10—not 8 or 12—because our focus is 10. This is how we will verify it.

Raman Patel: We have two ways of checking for errors. For positive testing, we validate input variables by values, dialogs, and sequence. We expect the result to be an output from the SSMs. For negative testing, we will again look at the invalid values. If it's 10, we don't expect 8 or 12. Then we have a sequence of dialogs we want to make sure occur. Then every disposition is carried out in a particular process of priority, so we will make sure that the sequence of the Test Procedure is in conformance. We will also do this for negative testing—errors will be reviewed. What do we do next to continue the test? If there are errors, we will review them and figure out what needs to be done.

Raman Patel: In this example here, what are the error conditions? Test Case 1 tells us that we want to check the status condition. The objective is to verify interface requirements from the positive Test Case. Let's look at the inputs. You have several inputs beginning with the object ranges here—the maximum or the intersection unit control status, for example. Two objects are shown here as an input. If there are errors, we expect they will be attached to the specific inputs. We expect to test System Control as a 2. We will make sure that the SSM returns to its correct value of 2. In some cases, if you do this manually, 5 will show up. The status of the control unit—in this case, the SSM—will tell us whether things are working out the way they're expected to. The status condition will be one of the 8 shown here. We will reduce the errors. We will understand what went wrong, what worked, and so on through this results analysis.

Raman Patel: The PRL will provide us with the item's mandatory requirements—marked here in the Conformance column as "M" and selected by the user in the specification as "Yes" in the Support column. Everything listed with "M" and "Yes" in these two columns will be tested. Which tool can we use? There are many ways we can look at these testing tools.

Raman Patel: There are available tools to check the SNMP. For example, Internet communication—this is a widely-used protocol nationwide and within the computing industry. We can understand how the Internet communication will work or how the network management protocol will work in general. That covers SNMP. For data analysis, there are all kinds of electronic machines available, such as a protocol analyzer that can also be used to analyze whether the device is communicating properly or not. There are several NTCIP testing tools—one is the Internet and serial communication testing of all objects within the MIBs with Set/Get. These are the dialogs. Set means modifying the function and the device. Get means retrieving the data from the device. Certain objects are read only; they are not settable. It depends on which process you are conducting and which message you are sending to the device. There are also logs and reports. So there are tools available that you can use at various levels.

Raman Patel: Additional information about the Test Procedures can also be obtained from other NTCIP standards. We have a very helpful Test Procedure available for dynamic message signs—Annex C. The ESS standard has also produced an Annex C—the Test Procedure can be used for other devices as well. Module T313: Applying Your Test Plan to NTCIP 1204 ESS also teaches us how to conduct an actual test. Through using the Test Procedures for this module by default, we can understand how other standards have used Test Procedures and bring that knowledge back to the SSMs. TPG 2.1 software was mentioned earlier. These are some of the tools that are available for testing.

Raman Patel: Your last activity is to answer: Which is not a valid statement related to SSM testing documentation? We have four choices: A) Test Plan contains an overall testing approach; B) Test Design contains project RTCTM; C) Test Procedures are provided by the manufacturer; and finally D) Test Procedure includes error detection.

Raman Patel: The correct answer is C. It's correct because the statement is not true. Only the agency specification will specify what the Test Procedures are. Test Procedures provided by the vendor are invalid. Answer A is incorrect because the Test Plan does provide us with the overall design approach. Answer B is also incorrect because the Test Design contains the project RTCTM. The RTCTM is actually the core of the Test Design. Finally, D is also an incorrect answer because the Test Procedure includes error detection. The Test Procedure gives us both positive and negative testing for expected and unexpected results. That's what we are looking for towards the end of the testing process: to understand what worked, what didn't work, what will work, and what will not work in the future. That's what the Test Procedure will establish: a pass/fail kind of expectation from the testing.

Raman Patel: To summarize the module. We discussed the system lifecycle, where to prepare the Test Plan, and where to use it. We also went over the structure based on IEEE 829—what does a Test Plan structure look like, purpose, general approach, and then discussed well-written documentation. In Learning Objective 3, we discussed the complete test documentation package using SSM specifications and the role of PRL, RTM, and standards. Finally, we described testing the SSM using a sample test document case study. Collectively, these learning objectives will give us a good set of skills that we need to prepare a Test Plan for testing SSM.

Raman Patel: We completed module 304A earlier—it's available. Module B about requirements is also available and completed, and now we just completed T304: Applying your Test Plan to the SSM. Collectively, the curriculum for SSM is now complete and is available to those who perform these activities.

Raman Patel: We want to thank you for taking this course. Module feedback is always appreciated. We would like to hear your thoughts. This will enable us to improve the training process as we move along. Thank you again.