

Module 14— Specifying Requirements for DMS Systems Based on NTCIP 1203 Standard Version 03

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, and we hope you find this module helpful.

Raman Patel: I'm Raman Patel. The title of this module is "A311b Specifying Requirements for DMS Systems Based on NTCIP 1203 Standard Version 03." This is an update of previous modules, and we have now updated this module to incorporate all of the new developments that have occurred in Version 03. This module is updated to reflect all the user needs and the requirements that came up in Version 03.

Raman Patel: I'm Raman Patel. I've worked for New York City for over 30 years as the Chief of Systems Engineering. I have been involved in the standards making process with the SDOs for a long time now, and currently, I'm teaching at New York University's Tandon School of Engineering—Transportation ITS and Urban Infrastructure courses.

Raman Patel: We have four learning objectives today for this module. The first learning objective is to review briefly the structure of the DMS Standard. Our second learning objective will explain the purpose of the Requirements Traceability Matrix—the RTM—and its benefits. Learning objective three will deal with preparing a project-level RTM with standard supply requirements and design content and the concepts. Our last learning objective will help us prepare a DMS specification, and we'll come up with a checklist that will help us to do just that.

Raman Patel: The first learning objective is about briefly reviewing the content of the standard itself—Version 03—and we will identify sections and details and where to find the information.

Raman Patel: What is DMS communication interface standard? What we have here is a dynamic message sign—it's called DMS—which provides DMS user needs, requirements, and design content. And the second main point about what's in the standard—it reflects the information that we can use to prepare our specification to develop or build a communication interface. We have here on the left side—you see the Management Station—which is communicating to the Sign Controller to the right in the field, and eventually to the sign. Our main interest is between the Management Station and the Sign Controller.

Module 14— Specifying Requirements for DMS Systems Based on NTCIP 1203 Standard Version 03

That's the subject of the NTCIP standard. Between the sign controller and the sign itself, it's a proprietary, vendor-supplied interface that we are not discussing today.

Raman Patel: Let's go into recapping previous module A311a, which was an updated module also, specifically for user needs. In that module we discussed and reviewed in detail the operational needs—and protocol requirements lists in particular—which we need to prepare requirements as well as specifications. We also reviewed in that module the different types of requirements briefly, and today we are going to go into a little more detail, and particularly focus on the requirements. We will introduce the Requirements Traceability Matrix, and how it is to be developed and used in preparation of the specification based on Version 03.

Raman Patel: The structure of the standard tells us about a lot of different things in it. It provides us the information we need to prepare the DMS specification. We discussed the concept of operation and user needs in the previous module. Section 2 deals with the user needs. Section 3 provides us with functional requirements, which we will go into a little detail today. Section 4 provides dialogues. Section 5 provides the management information base, which are design objects that we need for dynamic message signs.

Raman Patel: Specifically, in Part 1, we have a PRL as a template that was provided to us by the standard. We reviewed it in the previous module. Part 2 has an Annex which has the RTM—Requirements Traceability Matrix—that we will discuss today in this module. Annex C has Test Procedures that are useful in testing whether the requirements are met or not—so that's also of some interest to us—which will be reviewed in details in other modules that are coming up after this one.

Raman Patel: What is a requirement? A requirement—defined in quotes here—tells us that it is “a statement that identifies a system, product or process' characteristics or constraint, which is unambiguous, clear, unique, consistent, standalone (not grouped), and verifiable and is deemed necessary for stakeholder acceptability.” So this is a very concrete definition that has been used in all of our standards development work.

Raman Patel: Particularly, the definition that comes from the standard itself—NTCIP 1203—is of some interest to us because it is bringing us a little closer to what we're going to discuss in terms of the behavior of the standard. In that sense, a requirement is described as a condition or a capability to which a system must conform—either derived directly from the user needs or stated in a contract, standard specification, or normally imposed in the document. A desired feature, property, or a behavior of a system. Our focus is on the functionality and the requirements that come out in terms of supporting those functionalities. Here's an example—define a message. The requirement described here by the standard says the DMS shall allow a management station to download a message for storage in the sign controller's message library. This is a structure definition of standards. This is also the format used for the rest of the requirements in Section 3.

Raman Patel: There are several types of requirements. The first type—3.4—is “Architectural Requirements.” These are basic communication requirements that we need so we can set up the DMS signs for remote operations. It also supports the log data inside the sign library and functions, as well as in the memory process. And we can manage the access to the sign. So these are basic requirements that we have in terms of communication. The second type of requirement is the “Data Exchange” requirement. This allows us to manage DMS integration, and it allows us to control the DMS signs. We can monitor the

Module 14— Specifying Requirements for DMS Systems Based on NTCIP 1203 Standard Version 03

status of the DMS signs, and also provide for multi-version interoperability. So these data exchange requirements are set for that purpose. The third type of requirement we have here is called “Supplemental Non-Communication Requirements.” These requirements don’t belong to either one of the above two architectures, or the data exchange. They are derived from this, too, and they allow us to support additional functionality that we’re looking for. And we’re going to go into a little more detail.

Raman Patel: Architectural requirements define the behavior of the system in exchanging data across the communication interface. This example—3.4—between the Management Station and the DMS Controller supports the basic communication. It allows the Management Station to retrieve log data inside the sign and create the excess mechanism with the sign through the DMS Controller. That’s our main interest in terms of architecture requirements.

Raman Patel: We can retrieve data from the Management Station without going to the sign. This is a remote operation that’s typically done. Once we retrieve the data then we can clear the log. Once you have the data at the central location, there is no need for keeping bulky data in the sign itself. We can clear that remotely. As you can see here in this diagram at the bottom, the Management Station is communicating to the DMS Sign Controller in trying to do all of those different things that we just discussed. One of the things is that it will also allow the stations to know what’s going on currently—the access setting, who can access, what are the levels, and all of these different things. So the Management Station can get all of that information from the Sign Controller.

Raman Patel: Data Exchange requirements are listed in Section 3.5. They basically define the required behavior from the DMS system in exchanging data across the communication. There are three major areas. The first type of data exchange requirement sign allows the signs to be managed in terms of configuration. These requirements are specifically for configuration of the sign itself: ID of the signs, location, direction, and so forth. 3.5.2 is the control type of requirements—we can control the messages on the sign itself, which is one of the primary functions that we want the sign to perform. And the third type is the monitoring—what the sign is doing, what is the current status. We can do that through those sets of requirements. Again, we use the Management Station to remotely do all of these controlling functions and monitoring and managing functions.

Raman Patel: Data Exchange requirements to manage configuration include identifying DMS signs. What type? Is that a blank out sign—BOS? Is this a LED sign? Or is it fiber optics technology? Whatever the details about a particular sign is identified through that kind of configuration requirement. We also have the ability for the Management Station to determine what type of sign it is, and then what type of technology the sign is made of. You also have the ability to determine whether message capabilities are there—basic messages. What is the size of the message, beacons, accessing? Sign face size—what is the physical nature of the sign structure? Character sizes, number of pixels, spacing between the pixels. All of those little details are also handled through the Data Exchange requirements under configuration management.

Raman Patel: Managing the configuration in terms of, let’s say, for example, VMS capabilities. How many number of pages? The message is broken up in one page or two pages or three pages. What is the length of the message? What is the color scheme? Message display capabilities—how many messages can we display on a sign? All of those details are also important. Take a look at this example. It says

Module 14— Specifying Requirements for DMS Systems Based on NTCIP 1203 Standard Version 03

determine maximum message length in terms of bytes—how many bytes? We can remotely manage this to determine the maximum length of the message. What is the limitation out there? Look at the right screenshot here, you can see that there are two pages shown from ODOT—Ohio DOT. You see the first page System Under Test sign. The next message continues, and the second page to the right. And then there could be even a third page which is not shown here. This requirement deals with the configuration for a number of pages. Then we could also deal with the length of the message—how long in terms of characters and the byte size. Then you have color scheming and message display capabilities. All of these things are done through the Data Exchange requirements under Managing Configuration 3.5.1.

Raman Patel: Managing Fonts— as shown here. What are the fonts supported? How many font sizes are supported? We start with the maximum number of fonts. We want to know what is the level of activity that we can conduct during operations—of the need to change the fonts or not. So that's fonts supported. Character size—both vertical and horizontal. Number of characters per font. Whether we have a retrieving capability definition. Configuring a font—if we want to change the font we can delete the old one and put the new one in. We can also validate a font. Sometimes, you also have a need to do the graphics like you see here. For example, if we want to change the graphic, we can delete the graphic and put the new one. We can also put it in the left or we can put it in the right, and left justify and right justify, and so on. All of these things can be configured using these requirements. And these requirements are supported by the standard.

Raman Patel: Let's take a look at an example here of Data Exchange requirements in Section 3.5. How does the control of the DMS sign take place? You see the diagram here on the right. It's the Management Station talking to a Dynamic Message Sign on the right, using the Standard Supply requirements. For example, in 3.5.2.1 it says Manage Control Source. So the DMS can be managed from the central location through the Management Station. It can also be done using a laptop locally out there, or some other place that you are engaged in. The sign can be controlled from any number of different sources where the control mechanism can be initiated. We can also have the capability to reset a sign controller. We don't do that very often, but sometimes there is a need to reset a sign controller remotely without having to go there at the location. To control the sign face—activate a message. What do we place on the message? Manage a default message display parameters. There are certain things that are fixed and we can manage that. Earlier, we talked about fonts—for example, how many fonts. So that could be something that we can control and manage. Managing the message library—how many messages are in the library? We can schedule a particular message for display at a given time. And we can configure event-based messages. If we know about events in advance, we can come up with a very good set of messages that will go with the event as we intend it to.

Raman Patel: In terms of the physical view of message software at the Management Station level, you have the ability in the software to go by month, day, then time, and then pre-plan events. So we can retrieve a schedule, we can modify the schedule, and so on. This is all done through the Management Station and so the requirements are supported that will allow us to do these different things— scheduling messages. If the event is known in advance, a message can be scheduled to run between set times. So a message can alternate or change when it's needed without having to go through a detailed or lengthy process.

Module 14— Specifying Requirements for DMS Systems Based on NTCIP 1203 Standard Version 03

Raman Patel: Another example about Data Exchange requirements could be that of monitoring the status of the signs. The signs have a lot of electronics in them—small parts, large subsystems, the power supplies, pixels—and all of those different components make up a good deal of detail in the sign. So we can perform diagnostics remotely without going out there and opening up the enclosure. We can do it through the electronics process, in terms of performing the diagnostics remotely. So we can operationally test the status of the DMS components; we can check the error status; we can identify the problem of the subsystem, isolate it, and deal with it. We can also monitor the system for failures, in terms of details—for example, pixel errors, which is one of the very common problems that we may be facing out there. We can diagnose all of these different things. Also, the light sensing diodes will adjust. Sometimes they don't work—then we will know how to adjust the brightness for early design, for example. So all of these things can be done remotely.

Raman Patel: Current Message—what is the current message? If we have a situation where we need to alter or modify certain things, we have to go through the process first and say what is the current message, and then take it from there. Monitoring information about the current message is a very important function that we can check and recheck and so on. Monitor status of the DMS control functions—whether each sign is doing what it's supposed to do. So we can go through the whole process of checking the functionality of a sign.

Raman Patel: A real example here—in terms of monitoring status of the current message. A TMC monitors the current message through the DMS Controller and verifies the message being placed at the moment on the sign itself through the controller, and then the same message will show up at a central station on the menu where operators can verify and say this is the exact message out there. Sometimes, we also use cameras—not often—but there are signs that are also being used in conjunction with the CCTV cameras. The camera will project the validation of the sign showing the current message. There are a lot of support maintenance requirements built into this standard that can help us to achieve. In this menu shown on a screenshot, we see the list of different types of problems—checking door status, for example; if there is an error in terms of pixel control; whether the fan is not working or working; what condition, temperature, ambient temperature—all of these things are very important in terms of maintenance of a sign. So the standard allows all of these things you can do remotely from a Central Management Station. Some of the clear-cut methodology that we normally use: for example, light sensor ratings—whether the LED sign needs to be adjusted in real time; power supplies—where the power supplies are working or is it in a failed condition; pixel lamp tests—routinely done kinds of tests—how many pixels are bad; also, test messages—you want to structure a message and say this message is going through properly or not. There are a lot of signs that currently say Sign under Test and this and that. That can be also done remotely. Communication failure messages. Are there certain failures in the sign processing that we can check and recheck and make sure that the condition has been removed? Status of the beacons—it's an extra flashing mechanism that we put on the sign to alert motorists. Now we can check whether it's working or not, and what is the current status. The beacon is not always on, but when you need that on, we can check whether it's working or not. Status of a fan—one of the very common features that we use in the sign, it allows us to check whether the fan speed is okay and if the fan is working or not. Such things are very important in terms of maintenance. And so standard requirements are very supportive of all of these different things.

Module 14— Specifying Requirements for DMS Systems Based on NTCIP 1203 Standard Version 03

Raman Patel: Supporting Supplemental requirements don't belong to architecture or data exchange, but they are derived from them. There are additional requirements we have to be aware of in terms of what the capabilities are. They include many things—for example, how many VMS messages are required to support? We can say we need a library or support mechanism for 20 signs—20 messages in a sign, or 40, or whatever it is. Numerical requirements will be covered in the Supplemental requirements.

Raman Patel: Here is a list of all 13 supplemental requirements that are supported by the standard. They are listed in Section 3.6. For example, Supplemental Requirements for Fonts—how many fonts? If you don't say anything, the default will be one font will be supported by the vendor. So we have to make sure that we are specifying our requirement. It also goes for other things listed here—whether the sign supports graphics, scheduling, message activation—all of these different things are covered under 3.6 “Supplemental Requirements.” And they are non-communication requirements, so they actually deal with the parameters.

Raman Patel: This example will give us a little more detail in terms of how it's arranged. So we have PRL, or Protocol Requirements Lists, that we reviewed it in the previous module. The last column allows you to specify additional specifications. That's where the Supplemental Requirements will be listed. For example, we show here 3.6.1.1 “Support for a Number of Fonts.” That is a requirement. It's mandatory in the Conformance column, and it is required as per the Support column shown here. The first column is a Requirement ID. The second column is a Requirement. The third column is a Requirement ID. You follow this all the way in the Conformance, and you see what number of fonts we want to be supported by this sign. There is a placeholder way of doing things, and you say the DMS shall support at least three fonts. There's a standard that goes all the way from 1 to 255 in the range, but that's not unique. You need some finite number which needs to be predetermined, and that's how we can handle that. A similar mechanism can also be applied for other areas of Supplemental requirements.

Raman Patel: Now, we turn our attention to Section 4 in the standard structure, which lists dialogs. There are three types of dialogs. The first one is a G.1. It's called Generic SNMP GET interface to retrieve from a dynamic message sign. We can inquire and ask the sign what do you have? So that's the kind of inquiry we make, and whatever data the sign has, we can retrieve it from it. So a G.1 dialog will do that. The second one is G.2. It's the same as G.1, but we can do more. With the same one message, we can get more detailed numbers of pieces of data or messages from the sign. We can inquire and explore the sign for more data. The third one is G.3 which is called Generic SNMP SET interface. This dialog is used to tell the sign what to do. This is a control type of thing, and this dialog allows us to write into the sign a process which will produce a control functionality. So there are three dialogs the standard supports—generic dialogs—G.1, G.2, and G.3.

Raman Patel: Here's an example of G.3. The Management Station on the left side is communicating to the DMS Controller in the field using this G.3 dialog, which is a generic SNMP SET interface. It says that the generic SNMP defines a generic process by which a Management Station can send data to the device. This device will, in return, comply with the request and proceed and activate or implement whatever the message that came from the central location says and then provide a response back to the Management Station through the GET response. So this message coming from the Management Station is now implemented, and then responded to, by another message to the Management Station. That completes the whole dialog process between the Management Station and Design Controller.

Module 14— Specifying Requirements for DMS Systems Based on NTCIP 1203 Standard Version 03

Raman Patel: We have now the first activity.

Raman Patel: Which of the following is a false statement related to dynamic message design standard? There are four choices: A) Supports configuration, control, and monitoring of DMS functions; B) Supplemental requirements directly involve communication between the management station and the DMS; C) Supports remote communication to the DMS controller; and D) Standardized dialogs carry messages between two ends.

Raman Patel: The correct answer is B. Supplemental requirements directly involve communication between the management station and the DMS. This is a false statement. The Supplemental requirements cover range of values, such as message and line justifications. Let's look at it in terms of a field example. Here we are showing that the message is justified. If you want to keep your message in the center, you can have a center line justification. These kinds of Supplemental requirements are providing with—here's an example of the left justification. This is how Supplemental requirements are handled in terms of implementation.

Raman Patel: Answer A is a correct statement. It supports configuration, control, and monitoring of DMS function. Answer C is also a correct statement. It supports remote communication to a DMS Controller. And answer D is also a correct statement. We have reviewed G.1, G.2, and G.3 as a generic dialog, so standardized dialogs do carry messages between the two ends.

Raman Patel: Our second learning objective is to go over the purpose of the Requirements Traceability Matrix and its benefits.

Raman Patel: We will now go through what the RTM is and what benefits it provides us at different levels.

Raman Patel: Let's just review briefly. In module A311a, we discussed PRL—the Protocol Requirements List—and we went through all of the different columns, and their definitions, and what each column offers, and the PRL structures. In the PRL, we reviewed that there are User Needs listed in column one and two with their ID number and title, and columns three and four deal with the requirements. Column three has an ID number for functional requirement section numbers. The fourth column has a functional requirement cell. That's what we are looking at here. The fifth column has a Conformance which is "Mandatory M," where "M" is "mandatory," and then the Support column next to it is a "Yes" or "No." All mandatory requirements are supposed to be supported. So wherever you see "M" in PRL, they are supported with a "Yes" column in the sixth column. That's what we dealt with. Now we have in the PRL both the user needs and the requirements. What we take away is that in Section 3, we will deal with further processes.

Raman Patel: The terminology here is what is RTM? RTM is a traceability. Traceability is defined as the ability to follow or study the logical progression among needs, requirements, and design details in a step-by-step fashion. That's what traceability means. RTM plays that role. The Requirements Traceability Matrix—or RTM—is a table which provides us a complete design. Design means dialogs and objects for each requirement we have listed in the PRL. Everything we saw in the PRL in terms of requirements is now coming to the first and second column of the RTM. These are the same requirements we saw earlier in the PRL. The first column has a Functional Section Number, as you can see here: 3.5.1.1. The second column for the RTM is Functional Requirement. It says identify DMS. The third column is a Dialog ID. The

Module 14— Specifying Requirements for DMS Systems Based on NTCIP 1203 Standard Version 03

fourth column is Object Section Number. The fifth column is Object. The last column is called “Additional Specifications” activities, and will allow you to put in the notes and things like that. The RTM is now part of Annex A on Page 233 of the standard.

Raman Patel: Let’s review further using an example for the DMS requirement. For example, this requirement says “Determine Sign Type and Technology.” How do we trace it? Using dialog G.1 and associated design objects 5.2.2 and 5.2.3. It’s the same RTM we just discussed earlier. We have inserted detail under the 3.5.1. Now you have detailed requirements. It says “Determine Sign Type and Technology.” This requirement is associated with a dialog called G.1—generic dialog one. That dialog is now working in conjunction with the design listed under the Object Section, which is the fourth column—5.2.2 is the section number given to the object. 5.2.2 reads “DMS Sign Type.” 5.2.9—unrelated—reads “DMS Sign Technology.” We have the requirement, we have the dialog, we have the object number, and we have the title of the objects. The RTM provides us everything we need in terms of design. Design includes dialogs and objects to complete the message. The last column is empty here because we don’t have much to say about it. We will talk about that in a little while in terms of examples that explore this further. So you see these in the red box that came up? These are the functional requirements: the dialog G.1 next to it, and then the object’s design itself. All three things are now allowing us to trace each other. The requirement is traced with dialog G.1. Then you have the objects, which go right back to the requirement. All three things are traceable through the use of RTM.

Raman Patel: Let’s look at another example in terms of a Management Station communicating to a DMS Sign Controller in the field. The RTM presents the standardized design content to build the DMS communication interface. So you and I as the users or specification writers—we don’t have to design, because the standard is providing us that design as a complete set of objects and the dialogs that go with it. That’s what we are using between the Management Station and the Sign Controller as we are showing here. The interface will conform to the standard only if we use the functional requirement that’s implemented with all of the objects that are listed in the RTM. That’s what we are saying. The second point is that the Management Station implements all dialogs traced from the functional requirement. We are keeping the dialogs and the objects together. We are not letting them fall apart from each other. That is what the Conformance design will do. As long as we keep the dialogs and objects together as prescribed by the standard, the design will be conforming as designed by the standard.

Raman Patel: This example will further tell us how this is done in terms of parts of the table. The first line of the RTM gives us the title’s name, as you can see here: the headings, as we might say. The first line is a section number of functional requirements, titles, and the dialogs. So that completes the headings.

Raman Patel: Then we will go into actual object ID a little more details here. The object ID has a section number—where the object is listed in the standard—and that object will actually fulfill the functional requirement. So what is the name of the objects? And then you have the last column, which will provide us additional details if we need to put it in there.

Raman Patel: A single message is a very simple type of message completed using dialogs G.1, G.2, or G.3. These are generic dialogs that are straightforward to use. For example, in this RTM table that we’re looking at is determine the sign type and determine the technology the sign is made of using G.1. G.1 is the retrieving process; it’s the generic way of asking the sign what is your type, what is your technology?

Module 14— Specifying Requirements for DMS Systems Based on NTCIP 1203 Standard Version 03

G.1 will deal with that. 5.2.2 deals with the DMS sign type. 5.2.9 deals with the DMS sign technology. If you want to know whether it's the blank out sign—BOS? Is it a changeable message sign—CMS? Or is it a line matrix? You are at the Management Station and you are asking the sign what type of sign it is. The sign will display to the respondents saying I am a line matrix, and it'll say I'm a singular message sign, or whatever. So that's the first level of discussion or conversation you will have in terms of communication with the Management Station to the Field Controller. The second level will probably involve telling me about yourself—like what is this? An LED—Light Emitting Diode sign? Is it a flip disk? Is it fiber optics? The operator at the Central Management Station has a need to know all of the details about this particular sign. You know the location through the configuration mechanism. Now, we also can find out details about what's in the sign, what type of sign it is, and what technology it's made out of.

Raman Patel: For example, more complicated data exchanges will require complex messages. Or if you have longer messages, for example, they will require a message set that comes from Section 4. In Section 4, the standard provides us complex messages. The more detailed type of message is already done for us by the standard. We don't have to struggle to create a long message. The dialog will handle that. When we have a long message to send, the dialog will take care of all of those different things. This example here—Activate a Message—it's a pretty lengthy message. We'll use 4.2.3.1, which is a complex dialog which is provided in Section 4. We will not use the generic one. If we use generic, it will take a long time. The standard says use this particular dialog and you will get through the process of creating a communication scheme, which will allow you to active a long message. Now, the message itself will come from these objects listed in the red box to the right. 5.7.3 and the address of the other DMS objects that are listed will combine and provide us that activation message. That's what the takeaway is here.

Raman Patel: A special note on the importance of dialog order. As you just saw, the dialogs can be simple generic, or it could be a complex and lengthy one. This special note is prepared to alert us that data exchange depends on the order. There is an order in terms of how we can communicate to the sign. The data exchange order is important unless the dialog states otherwise. We rely on what the dialog tells us to do. All of this work is done already by the standard. Second, if we alter the sequence, we will be in big trouble. We don't want to do that. So we must always follow the sequence prescribed by the standard dialog that's given in the standard—Section 4. We will be only be conformant if we maintain the dialog sequence. If you go out and change the dialog sequence, we will be making a mistake, which will probably cost us in terms of interoperability. This special note is alerting us in terms of how to handle dialogs. If you are a system developer, a lot of these issues have been dealt with by you, and if you're not facing any of these things, at least be aware that this situation could be something that needs to be addressed.

Raman Patel: What are the benefits of RTM? The benefits of RTMs are quite a few that we need to acknowledge and understand. If you are a procuring agency, you have a certain number of benefits in terms of preparing a specification. If you are in traffic management operations—yes, it will also help you understand your needs are met. Then you have a system developer's benefits. Also, the manufacturers and conformance testers—the people who test the dynamic message sign. There are a number of ways we can analyze our approach to understanding RTM's role and the benefits that it offers us at different levels. Let's ask this question: If you are a procuring agency, you have a question—is anything missing? When you prepare a specification, this question will be answered through RTM. If you are a traffic

Module 14— Specifying Requirements for DMS Systems Based on NTCIP 1203 Standard Version 03

management center of operations, you will say—will the interface support my operation? Because you're an operator, you're a system person, you always want to make sure that the system eventually will do what you expect it to do. That kind of thing. If you are a system developer—do I have to do it all over again? You did this before somewhere else. You implemented dynamic message signs a couple of years ago. Now, do you have to do something different or similar? RTM allows you to maintain that handshake with how different things are done in system implementation. If you are a manufacturer, you have a question—how do I know what the customer's requirements are? A typical simple question, but it's a real question. So we say RTM tells us what the requirements are. It tells the design. It gives us the object. We've got a pretty good idea of what the actual design is supposed to be. If you are a tester—finally, somebody has to test whether this system works the way it is designed or expected by the customer. What do I test for? Conformance to the standard is required. You want to buy something that's conformant to the standard. All of these different questions you may have are only answered at the stage when you test the sign. RTM comes in very handy in answering all of those different questions.

Raman Patel: The Management Station communicating here is shown to the DMS Controller and asking all of these different questions. It can answer all of these different things and what are we doing collectively. Benefits specific to the agency include the standardized design it provides to the users. You as a user work as an agency organization—people who write specifications. Everything is standardized in the standard; we don't really have to design it. That's a good peace of mind. The second thing is RTM will enable the DMS testing process at a later stage. Right now we are procuring, but once you have finished procuring and everything, then the system is built and we have to test whether the system is doing what it's supposed to do. You will be using RTM to help you conduct that process. RTM also enables interoperability. It makes sure that all design processes, sequences, dialogs, user needs that came from a PRL earlier in RTM we were looking closely at the design itself. Conformity to the standard will be coming through the RTM for that purpose. Sitting together and making sure that we understand each other's needs and all the parties have common understanding. That's the main part of what working together means. It removes the ambiguities. The vendor knows what the agency wants, the agency knows what is expected from the vendor, the developer knows all of these different viewpoints, and everything is already in the open. Everybody is on the same level playing field.

Raman Patel: Particularly for the system developers, the benefits come through in terms of reduction and design work. You don't have to go through this cumbersome, detailed, ongoing, long process to design something in terms of the capabilities of the dynamic message sign. It's already done by the standard. In that sense, it reduces design work for the developers. It's also a very powerful traceability matrix. Are we missing anything? It maintains the order of interoperability, the sequence of a dialog. You're not missing anything. This is really a great checklist for a developer in terms of how the central system will be built and implemented. The protocol implementation and its implementation in the system—this comes in very handy as a checklist. It gives you all the different elements to look at very closely in terms of whether we are conformant to the 1203? After all, everybody is asking questions at the end of the day. Is this system going to be conformant to the standard? Am I meeting all of the things the standard says I should be doing? This is a very good way of handling those kind of questions.

Raman Patel: And now the vendor—what the customer wants. You have this expectation that the system will do this, this, and that. The vendor knows what users are looking for through the RTM in a very

Module 14— Specifying Requirements for DMS Systems Based on NTCIP 1203 Standard Version 03

unambiguous way. All of the capabilities through the design process are listed, and that's what the vendor acknowledges. It also ensures the product functionality. You are in a factory environment. You also know what the customer's needs are—what functionality the product is going to provide. Before you ship it to the customer, you're testing prior to the shipping. This is a very good activity. You don't want to send a client something that's going to come back to you. This is a good way of doing that kind of a final check. And then one thing is always on everyone's minds. I asked for this—you didn't give me this. That can be avoided. There are always some kinds of legal disputes here and there that can be eliminated from the start. If you have RTM, it speaks for everybody together, therefore such level of disputes may not arise. There may still be some minor issues to deal with each other but not with the sense of complexity and the legal type of issues.

Raman Patel: In terms of the general marketplace: marketplace means let's say multiple vendors; it's not just one vendor. You have more than one vendor. You have more than one agency that procures. You have a range of products, different types of signs, different technologies, fiber optics, LEDs. All of those different things. So we have different vendors, different agencies, and different products. On the other side, you have what the sign does. The sign implements a lot of different things. It supports multiple messages, multiple applications, traveler information, and control mechanism parking information. You name it—all sorts of applications we have under the ITS deployment arena are supported by dynamic message signs coming from different vendors and different technologies and different functionalities. Everything is working out good. We really have a very healthy marketplace in terms of deployment and manufacturing of dynamic message signs.

Raman Patel: Our second activity, then, is about...

Raman Patel: Which of the following is a false statement as it applies to dynamic message signs? Again, you have four choices. The first choice: A) RTM provides the standardized design content; B) Generic dialogs are used for single message to and from a DMS controller; C) Testing process uses RTM to verify each DMS requirement; and D) RTM does not reference dialog.

Raman Patel: The correct answer is D, because RTM does not reference dialogs. This is a false statement. RTM does provide you the order in which dialogs must be implemented. It's a very specific and very straightforward way of saying you must implement these dialogs in this order to make everything error-free. To make the communication error-free and to achieve interoperability, RTM has provided this outlet. Answer A is a correct statement: RTM does provide the standardized design content. B is also a correct statement: RTM has G.1, G.2, G.3 generic dialogs, and they are meant for single conversation between two ends. Answer C is a correct statement: the testing process uses RTM to verify each DMS requirement. Later, when we receive the sign, RTM will be helping us to conduct the test procedures and test cases out there.

Raman Patel: Our third learning objective now is to prepare a project-level RTM with standard supplied requirements and design concepts.

Raman Patel: We will learn how to deal with RTM in terms of what requirements are available in the standards, including the design content.

Raman Patel: Let's look very closely in terms of how the organization will help us, and we will reinforce our understanding using the various components that are provided. Here is the beginning. The first step

Module 14— Specifying Requirements for DMS Systems Based on NTCIP 1203 Standard Version 03

in any specification is to prepare a PRL. The Protocol Requirement List is the beginning of how we will organize our specification. We reviewed this before, and we said that for every user need, there is a requirement or a set of requirements listed in the PRL. We are looking at that in this red box—the set of requirements to identify dynamic message sign identity. There are several of them here listed as Requirement—3.5.1.1, for example, “Determine Site Type and Technology.” It’s mandatory in the Conformance column. Therefore, we have selected “Yes.” The next one is the H.2.1 “Determine Device Component Information”—which is also mandatory—therefore we selected M in the Support column next to it. 2.4 Also, “Determines Supported Standards.” Again, it’s M—Mandatory—therefore, we have to select “Yes” in the Support column. And further down at the very end, you have “Determine the Size of the Sign Face”—which is mandatory. We have selected “Yes.” We can see that everything mandatory we have to select “Yes.” And anything that you see O or Optional—it depends on you whether it’s required by the project or not. If it’s required, we select “Yes.” If it’s not required, we select “No.” This completes the preparation of PRL. This is the first step in preparing the specification.

Raman Patel: If you say “Yes,” this will lead us into the requirements as scheduled in the specifications. RTM provides the design for each supported requirement. How do we complete the RTM? How do we complete it in the way say of populating the table? We want to provide as much as information as required in the RTM table. In RTM, we have columns and rows. We will put in this red box—as you’re seeing here now—between the dialogs and objects. It covers dialogs and objects together; it will provide us the design. Our focus is design in this particular discussion. Design means dialogs plus the relevant objects. That’s what we are looking at. G.1 is the dialog we will use—generic for this particular example. 5.2.2 and 5.2.9 are the two objects that we use to identify sign type and sign technology, respectively. The first DMS sign type will say I am a blank out sign—BOS. That’s a type. The second object leads us to say I am a LED technology. This identification of the requirement in the specification will be fulfilled by these two objects. That’s what we are saying.

Raman Patel: In a complex way, you have now a few more capabilities in this table here—more examples. For example, we are dealing with Manage Configuration. How do you manage configuration as we discussed in Section 3.5.1.1? We have a bunch of functional requirements listed in column one and two, as we are looking at it. Then you have in the Dialog column G.1, G.1, G.1, and all the way to the four—the last one has G.3. There’s another dialog we also need. Then you have a set of objects in the Object column. What happens now is that, based on the capabilities that you are desiring, your process will continue further and identify those requirements, and identify the objects related to those requirements using the generic dialogs that we just listed.

Raman Patel: This gives us more information. This is a snapshot of an actual menu at the Management Station, and you can see that these requirements are about determining message display signs: the number of pages and maximum length. To the right, we show how many pages we are talking about—what will be required. We can see here that there are two pages shown. A page means a portion of the message shown, and then the next page will show the rest of the message. In that sense, you’re looking at two pages—page one and page two. There may be a page three, but the requirements to the left—3.5.1.2.3.2, or above that is 3.1—will tell us how many pages. If your requirement says two, you will have two capabilities. Just make sure that all of the requirements that you have stated in the specifications are fulfilled. We’ll be eventually checking it during the testing process.

Module 14— Specifying Requirements for DMS Systems Based on NTCIP 1203 Standard Version 03

Raman Patel: Setting the Sign Controller—Manage Control. Sometimes we have to go through this process in resetting. This one, G.3—which is a SET dialog—it now has an object with which we can accomplish the function. 5.7.2 says “DMS Software Reset.” Using this object, we can use the G3, and then complete the process and reset the DMS Sign Controller. If you have a requirement that says I want my DMS sign to provide me the capability to reset it remotely without having to go there. You’re looking at the Management Station on the left which is communicating to the DMS Controller in the field on the right. It says the DMS shall allow a Management Station to reset the Sign Controller. The design above, you just saw, allows us just to do this. So the requirement is fulfilled using 5.7.2: a design object and a G.3—a generic dialog three—which is a set dialog. This completes what should be in the RTM.

Raman Patel: One more example that takes us in a different area. In this case, it’s managing control with the monitoring status of a sign. What is a sign really showing right now? These requirements are listed here. This is a little more of a complex process. We have complex dialog here—4.2.4.1 and 4.2.4.2. These two dialogs will allow us to implement this requirement in the real world. We can see on the bottom that you have photographs that show the pixel is missing: the bad weather, the ‘T’ is missing. Then on the right, you have a whole pixel set that is faulty. It’s showing dark. What this means is that we can remotely monitor a sign and see what’s going on. Those requirements are now implemented in this RTM you are looking at.

Raman Patel: An additional example is if the door of a DMS controller open or closed. If you don’t have this requirement, that’s one thing, but do you really want to send somebody to check it? You know something is not right. A lot of windy conditions or vandalized conditions, or many times something could have happened. You want to make sure that the sign controller door is closed or not. These requirements will allow us to do that. You can monitor without going to the sign and say there is an object of 5.1.1.1.6. It’s called “DMS Status Door Open.” Is the door open? The answer is “Yes” or “No.” It’s a binary answer. Whether it’s opened or closed the answer will come through this requirement.

Raman Patel: An example of monitoring the message itself. These requirements are implemented as we see here. You have a set of requirements which allow us to monitor the status of the current message—what the sign is showing. And you will have a visual verification at the Traffic Management Center. In the Management Station, you can see the message the exact way it appears in the field, which is also an important activity that the operator goes through.

Raman Patel: The process to conduct this communication from the Management Station to the sign. As we show here in the photographs below, we can see the Management Station wants to talk to the sign. But how do we do that? How do we communicate to the sign? We show on the top in the RTM the design objects necessary, and what kind of dialog we will need. We show H.3.1.3 and G.3 as a dialog, and then we show objects listed in standard code 1103 Version 02, which is a companion standard where we have all of the global objects listed. That standard is invoked, and we say we’re going to make sure that we are now able to look at the sign and see what’s in it—what the log data is. It could be, for example, at two o’clock in the morning the door was open. At 3:15 the power was on. Any of these conditions, for example, can be viewed by the Traffic Management Center remotely—when the communication comes back, if it’s broken, or the dial up is used to know this. But whenever there is a need to check what the sign has done, and which messages are logged inside the sign, you can retrieve it. This process is similar to many other NTCIP devices. In particular, dynamic message signs are an interesting way to know what

Module 14— Specifying Requirements for DMS Systems Based on NTCIP 1203 Standard Version 03

happened to the sign. That kind of inquiry can be made when the communication is there, and when there is no communication, or when the communication is restored. That's what these architecture requirements will do for us.

Raman Patel: Our activity now here is to answer this question.

Raman Patel: Which of the following statement does not apply to RTM? Four choices: A) RTM includes architecture requirements to communicate with the sign controller; B) Includes DMS user needs; C) Includes dialogs and objects; and D) RTM lists requirements for retrieving data from a remote Dynamic Message Sign—DMS.

Raman Patel: The correct answer is B—Includes DMS user needs. It's the correct answer because it's false. We are saying that RTM does not include the user needs. User needs are in PRL. In RTM, we only have requirements and a design. In PRL, we have user needs and requirements. That's what it means. Answer A is an incorrect answer because the statement is correctly stated. It is a valid statement. Architecture requirements make the interface operational. C, similarly, is an incorrect answer because dialogs and objects are part of the RTM. D is also an incorrect answer because the statement is true. We can retrieve information from a sign, and that part is RTM at play.

Raman Patel: Our last learning objective then is to prepare a DMS specification.

Raman Patel: How will a checklist look? We want to make sure that everything we need to understand is covered as a checklist so we can complete the specification properly.

Raman Patel: There are several ways we can deal with these issues. In terms of procurement specifics, we have our focus on, for example, a hardware specification. When you procure, you want your sign size, physical size, functional requirements, performance requirements, structural requirements, mechanical requirements, electrical requirements, environmental, temperature, humidity. Those are all kinds of hardware specifications. That's one type of specification we have in the procurement process. The second type is the software specification, which are functional requirements and performance requirements. The third one which we are focusing on in this module is the communication interface specification. This is the third piece of these specifications we are focusing on. In this part of the specification we will have user needs, functional requirements, project PRL, project RTM, and testing documentation that applies to DMS. We are focusing on five different components. That's what makes up the communications interface specification. That's our focus. All of these things are also attached in the contract requirements. You have system development, testing, deployment, integration, operational maintenance, project management—all of these different pieces of specification going into the general category—and they also apply to the communication interface. When all told, at the end of the day, we want to see that this interface between the Management Station, shown here on the left, and the DMS Controller, shown on the right, is taking place and the sign is connected to the DMS Controller. These three components will tell us that the picture is pretty complete. You are now able to communicate from the Management Station to the Dynamic Message Sign Controller in the field. That's what we are covering in the third type of interface specification we have shown here. It's certainly not a type, but it's a third component in a major document.

Raman Patel: Let's look at building a communication interface specification—our main focus: the user needs, functional requirements, project PRL, project RTM, and testing documentation. Now we want to

Module 14— Specifying Requirements for DMS Systems Based on NTCIP 1203 Standard Version 03

make sure what that standard provides us in terms of our needs for the specification. The user needs come from Section 2. We already discussed that, and we went through the whole discussion on PRL. Section 3 provides us functional requirements in the standard, and Annex C will provide us test procedures. All of these three things we will be using to prepare the communication interface specification on the right. What will the user provide? We just talked about the standard. Now, we have to talk in terms of what we will do as a user—the person who writes or people who prepare the specification. We have to prepare the project level PRL. The standard doesn't do PRL project level; the standard provides a whole, big PRL. We have to now customize it and tailor it to our needs. That's what we will do. At the project level we will prepare a specific PRL and populate it with columns and rows. Second, we will do the same thing at the project level for RTM. We will have everything that we need in terms of each requirement and the design that comes with it to fulfill it. That's what the RTM will do. And the third thing, we will also prepare a set of testing documentation, which you will learn in the next module. These three things are done by users. That will go in the specification.

Raman Patel: Here is the checklist of key elements that we must acknowledge and that should be present in the specification. First thing, address interoperability. Second, integrate project PRL and RTM in the specification. Third, maintain consistency with DMS product specification. Fourth, specific performance requirements. And fifth, coordination requirements. All of these things must be addressed in the specification that we are preparing.

Raman Patel: Let's talk about addressing interoperability. Why is it important? Well, DMSs are deployed over a general wide area—over a region—and often procured from different vendors. You buy something today, and then a year later, you buy something from somebody else and this and that. Everything is done in a modular way over a period of time. That's one thing. The second thing is that the signs are shared by multiple agencies. You are interested in what the other signs are showing owned by somebody else. Signs are generally shared over the large area so that's very important. Different jurisdictions are involved. Third, major objectives require capability. In other words, to share and control a sign everyone's capabilities must be similar or the same. These are kinds of points that we work with. We start our process. How is it obtained? Well, if you are interested in interoperability—meaning that you want to be able to share and control your signs with other people—you want to go over a wide region, and you want to take care of everyone's needs, and so on. In that sense, we are saying here in the how part that all agencies seeking interoperability must have the same user needs, the same requirements, the same design objects in their project PRL and RTM. Both the PRL and RTM must have the same user needs, requirements, and design objects in order to achieve interoperability. And second, we must use SNMP—Simple Network Management Protocol—with other applicable standards. These two are a must to achieve interoperability. SNMP is a protocol used by the standard for dynamic message signs. We have to use that; we cannot use a proprietary protocol from the vendor. This is how we can ensure that we are getting what we are looking for in terms of interoperability.

Raman Patel: The second point is that we need to integrate PRL and RTM in project specifications. A bunch of issues related to defining user needs to create an interface for exchange. The PRL will do that for us. RTM provides the standardized design. We don't have to struggle coming up with the design and say, how do I put these things together? The standard has done that for us. Third, communication standards are underlying—there are several levels as you all know in NTCIP. We have a stack. We are

Module 14— Specifying Requirements for DMS Systems Based on NTCIP 1203 Standard Version 03

only dealing with information level vocabularies here and all of that design stuff. The other standards will come—transportation standards, subnet standards, and so on—they have to be also specified. SNMP was mentioned for the implementation or integration purpose. That's another communication protocol we talked about. References to other interface standards. There may be a standard you also mentioned—for example, 1103 in our RTM. That has to be mentioned. With the version number and protocol, the version number of the standard and the publication date should be also specified. And the last is the value ranges. All of these objects have ranges. Number of fonts. So you probably have to say very clearly in the additional column that we are looking for two fonts. If you don't say anything, the vendor will only give you one. That's an example of how we handle ranges—how many messages, that's also a physical number. These ranges and their parameters must be very clearly clarified inside the RTM process. Finally, we want to say give me everything you've got. That notion that we live through in many ways is not valid. You want to be conformant to the standard that we have to deal with the PRL in a proper way. We need to stay away from the wish list thinking. Just ask for what your needs are.

Raman Patel: The third point here is the consistency between DMS product specifications. Many times we say the hardware means types of signs, technologies, mechanical, physical constraints, operational parameters, temperature, and humidity, for example, comes to mind. Those kind of hardware-related specifications must be consistent with the communication interface specification. In one place, you ask for a sign that should withstand a temperature range and a humidity range. That should be also consistent in communication. Both of these different areas must deal with these parameters in the same way. So that why consistency is very important.

Raman Patel: Performing the requirements. For example, how many devices are on a channel at onetime? Response mechanism is the only way we can discuss this in terms of performance. What we want to say here is that performance requirements are not covered. We are only dealing with the communication aspects of the implementation process and requirements. For example, response time is addressed in a standard called NTCIP 1103. It is a nominal way of saying minimum requirements. We are not engaged in terms of saying how fast a sign should write the message. For that, we will defer this discussion to the other standards and reference here is provided to G.5.5, which tells us about how DMS shall propose the GET and GET NEXT, or SET request response. There are some limitations which are subject to this particular clause. We should be aware of it and include that in the specifications.

Raman Patel: Coordination requirements. Often, we have to make sure that we are looking at the standardized solutions specified in the RTM. We need to coordinate that part in terms of making sure that we are not uncoordinated or we are missing something and we are saying the same thing differently. Coordination is to be adjusted accordingly. The copy of a PRL—completed PRL, not incomplete—if we don't leave it to the vendor or developer we have to do as an agency: prepare a complete PRL and complete RTM, and put it inside the specification. Sometimes we need to specify coordination needs with the users. We have to put a clause here and there to make sure that vendors, developers, and maintenance types are all coordinated among each other. Each one of these areas sometimes requires touching base with each other—as we say in the industry. We need to know who is doing what and how this is going to come out. We need to look at that in the context of how we're going to complete the specification in the sense that coordination has occurred.

Module 14— Specifying Requirements for DMS Systems Based on NTCIP 1203 Standard Version 03

Raman Patel: Conformance versus Compliance. These two terms are very often discussed in the Professional Capacity Building (PCB) modules. Conformance meets a specific standard. For our discussion in this module, we say to claim conformance to NTCIP 1203 Version 03, the vendor shall minimally satisfy the mandatory requirements, which were identified as “M.” “M” means mandatory. Those requirements must be met. Your Support column in the RTM stresses that. That’s conformance. The second point to it is that the vendor has additional features to provide. If they are available and they are done within the context of the standard, and they are not proprietary, and you want your contracting agency to have those features, that’s by all means. But they have to be within the preview of the standards. They cannot be in an isolated way. They cannot be proprietary. If you want to offer those, that’s fine. That needs to be conformant with the standards. These two components are very important. Now, we deal with compliance. Compliance in general means meeting agency specifications. We are guided by the agency specs for the compliance. These two terms are often discussed in specifications at length.

Raman Patel: Backward compatibility is a very good question to ask. This version of 1203 Version 03 is compatible with the previous two versions 01 and 02. The answer is yes because in Version 03, we have a specific column provided as an additional column. In there, if you want your implementation or specification to be backward compatible, you have to assume the standard will provide it. That’s by default—Version 03 provides that. But if you don’t want this procurement—for example, you are buying brand new signs; you’ve never done that before; and you are acquiring a set of newer signs—then backwards compatibility may not be an issue for you because you’re starting everything from scratch. But if you have a few signs that you bought years ago and they were Version 01- or 02-based, and now you are buying new signs, in that case, you will have to check the support which is may or may not be required for Version 01 and 02. The reason is that there are certain objects that we have redone in Version 03 which may cause problems for you. If you check and say I want support for Version 01 and Version 02, then the vendor will be more informed and your procurement will be that much more successful. That’s what we are saying. Otherwise, we want to conclude by saying that Version 03 is backward compatible with Version 01 and 02. That’s why we need to make sure the additional column is used in the PRL.

Raman Patel: At the end of the day, somebody said what if my requirements, my user needs are not met? I have something special that I need to do, and there could be something that the standard is not supporting. You can probably do that, and we are not encouraging anyone to go out and do extensions to the standard, but if there is a need for it and it’s a specific requirement, you must do then it has to be within the context of the standard. Specifications must include direct dialogs and objects in sequence. We talked about this at length in this module. If you keep the order of the dialogs and objects and these things are available for everyone to know that these specific things are done—which is different, which is not in the standard—then you have a reasonable chance to get things done successfully. You will not be able to define new dialogs or objects. The dialogs and objects have to be properly structured and used the way the standard prescribes. Standards are very clear about that, and you will lose the conformity. As soon as you do the extension and do your own thing, you will not be conformant with the standard. That’s a major dis-benefit that you have—the damage to interoperability that might occur. On the other hand, the benefit is that yes, you are still within the context of the NTCIP family. You are able to solve your problem. That’s a solace. That’s something that you say I got through my project successfully. Other than that,

Module 14— Specifying Requirements for DMS Systems Based on NTCIP 1203 Standard Version 03

there is really no good benefit that comes out of using extensions. And in the standards, we have experienced that any time you do the extension, it always brings up some kind of issues that you may have to deal with.

Raman Patel: Drawbacks and consequences of these things, there are quite a few. If you look at it very closely, you may end up with the Management Station not being able to exercise the new capabilities because the proprietary objects and proprietary dialogs of all of those things will not work with the system that's already integrated at the Management Station. The second point is that if it's not consistent, then the interoperability is going to be missed. It's going to not be achieved. Also, it may not be possible to share each of these devices because you have this implementation, and somebody else has something different proprietary, then it will not work with each other. We want to also go into how at the later stage, after the procurement and the signs are in you're going to test them. At the testing level, you will also have additional issues. For example, you need a special skill because of the proprietary objects design—you really don't know about it, so you need knowledgeable people and that's going to add costs. So the cost and complexity of putting a special extension is also another issue that you will have to consider and stay away from for all of those reasons.

Raman Patel: If you do have to go through this process and extend the standard, make sure that the configuration is done properly and follows the rules according to the standards, and that the orders of the mechanism are maintained. Dialogs and object definitions are not allowed to be redefined. You have to follow the standard methodology that we have. Also, all of the extended work must be published. Everybody should be able to know. We cannot say that I have this information, but I cannot give it you. That's not going to work. All of these things we should be aware of and acknowledge in an orderly way.

Raman Patel: Our last activity then is to answer this question.

Raman Patel: Which of the following is a false statement related to a DMS specification? Four choices: A) Specification includes PRL-identified user needs; B) Project RTM provides project-based design content; C) To achieve interoperability, either PRL or RTM is required; and D) Extended standard is not conformant to the DMS standard.

Raman Patel: The correct answer is C—to achieve interoperability, either PRL or RTM is required. It's correct because the statement is false. To ensure interoperability, we need both PRL and RTM, and SNMP—Simple Network Management Protocol—in the specification. These three things are the basis on which the interoperability is ensured. That's why it's a correct answer. A is a specification and includes PRL-identified user needs. It's an incorrect answer because the statement is true: The statement reflects PRL does have user needs. B is also similarly an incorrect answer because the statement is true: RTM is the complete source of design—it has a dialog and a design object. And answer D is also similarly an incorrect answer because the statement is true: extended standard is not conformant to the DMS standard. As soon as you extend a standard, you are not conformant with the NTCIP 1203.

Raman Patel: Module summary. In Learning Objective One, we touched base with the review of the standard structure. We discussed what Sections 2, 3, 4, 5 have to offer. In Learning Objective Two, we discussed the role of RTM, what it is, columns and rows, and how we achieve the benefits using RTM. In Learning Objective Three, we went over how to prepare and reinforce the thoughts that RTM brings to the table. We want to make sure that eventually we have a RTM that has requirements and associated

Module 14— Specifying Requirements for DMS Systems Based on NTCIP 1203 Standard Version 03

design content. And in the last learning objective, we just went through the checklist in terms of what should be included at a minimum and discussed issue points separately. Together, these learning objectives have provided us with an additional skill set, and the methodology and the information that we can find from the standard, and how to use it in terms of preparing both RTM and PRL in a complete specification as we require.

Raman Patel: We have now completed module A311a—previously “User Needs for Dynamic Message Signs.” We have completed in this module the requirements discussion for the 1203, and in a future module, we’ll also deal with the testing—how do we apply your test plans to be compliant and tested for each requirement.

Raman Patel: In that particular module, you’re now going to learn four things. One is that you’re going to see the role of a test plan. What does a test plan look like? What it will do for us in the lifecycle of the whole system. Second, what are the key elements? What should we absolutely have as a minimum in the test plan? Third, in the application, how do we apply a good test plan for testing dynamic message signs? And finally, we will also learn the process of adapting the test plan based on your selected user needs and requirements. These four things you will learn in the next module, which hopefully will be scheduled as soon as possible.

Raman Patel: We thank you and remind you to complete the feedback. We appreciate your time in taking this module. Please provide us feedback in the form that’s provided to you and give us as much information as you can so we can make training improvements as we go along. Thank you.