

A309b Understanding Requirements for Ramp Meter Control Units Based on NTCIP 1207 Standard v02

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. I am Ken Leonard, the director of the U.S. Department of Transportation's Intelligent Transportation Systems Joint Program Office. Welcome to our ITS Standards Training program. We'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 Standard training is one of the first offerings of our updated professional capacity training program. Through the PCB program we prepare professionals to adopt proven and emerging ITS technologies that will make surface transportation safer, smarter, and greener. You can find information on additional modules and training programs on our website at www.pcb.its.dot.gov. Please help us make even more improvements to our training modules through the evaluation process. We look forward to hearing your comments and thank you, again, for participating and we hope you find this module helpful.

Narrator: Throughout this presentation, the activity slide will appear indicating there is a multiple choice pop quiz following this slide. You will use your mouse to select your answer. There is only one correct answer. Selecting the "submit" button will record your answer, and the "clear" button will remove your answer if you wish to select another answer. You will receive instant feedback on your answer choice.

Today's module is A309b: Understanding Requirements for Ramp Meter Control Units Based on NTCIP 1207 Standard v02. Your instructor, Raman Patel, is a fellow of ITE and has over 40 years of experience in the transportation field, and has been actively involved in the ITS standards development and standards training program, and has served as Chair of ITE Standard Committee and is a member of the NTCIP and the TMDD committees.

Raman Patel: Hi, I'm Raymond Patel. Target audience for this module includes traffic management and engineering staff, people who work at the traffic management center operational staff, those who are involved in freeway management and traffic signal coordination on surface streets, as well as the freeway management, also those who are developers putting systems together in ITS's different functionalities, and folks from

A309b Understanding Requirements for Ramp Meter Control Units Based on NTCIP 1207 Standard v02

private and public sectors, including the people who supply traffic controllers.

The recommended prerequisites for this module include this listing here. It begins with the basic introduction courses at the 100-status level, and several requirements of related courses are also listed here. For example A102 and A201 deals with how to acquire and then moves into the requirements and how to prepare and write the requirements for different modules. C101 particularly deals with the communication aspects that we'll be using today in this module. That is a required module that must be taken so that you understand how this thing works. A309a deals with the users' needs for the ramp meter control units and it is a module that immediately precedes this one, so that needs to be reviewed for better understanding of what we'll be discussing in today's module.

This curriculum path basically is a graphical representation. We are here at the last module, and there will be one more additionally shown here in the curriculum path for RMC will be about the testing.

Some of the acronyms used in this module are listed here. The list is also provided in the student supplement, so you can easily refer to that part of the supplement.

There are five learning objectives for this module beginning with how to develop the requirements using this standard, NTCIP 1207 v02. Then, what do we do? How do we establish interoperability? How do we gain vendor independence? All of those issues are discussed in learning objective 2. Then, we move into how to prepare traceability tables so that we can end up providing a good specification. That's learning objective 3. Learning objective 4 deals with the requirements, which are not supported by the standards if any particular project has certain abnormal or special requirement, then what we should do. The last learning objective deals with the specification preparation, how to develop it. Collectively, you get a better understanding of the skill set necessary for this RMC unit.

Learning objective 1 will cover the discussion of what the standard provides, what's in the standard, what is not available in the standard, and then we will move into the three areas where we find information about identifying requirements. Basically, we will also talk about the A309, the previous module, where we discussed user needs. Then, the standard itself has Annex A. We will cover that part, and then also discuss about how to configure an RMC device based on our normal procedures and what we have learned before. We will also discuss the criteria for writing well-formed requirements and then provide some examples.

The structure of the Standard 1207 v02 has several sections in it. Section one is very general: scope, references, terms, conditions, definitions, basic information. Section two deals with the good definitions of what RMC unit is and also what the RMC system is.

A309b Understanding Requirements for Ramp Meter Control Units Based on NTCIP 1207 Standard v02

There are references to other standards. The RMC meter control unit consists of the traffic controller, related sensors, detectors, and warning signs that a meter is on or off, and then, of course, the signals, the sequence green amber, red. The RMC system produces metering rate at the end of the process, what we call a metering rate or release rate, is established by the RMC system at a given ramp.

As you can see on the right here, there are two diagrams. The first one lays out on the top the location of a potential ramp, and the bottom photograph also shows how to actually see traffic moving both on the mainstream section of the freeway, as well as on the ramp.

Section three is a basic structure of the standard. Actually, it provides the design data for RMC systems. For the functionalities listed here on the left, you have several functionalities listed—mainline detectors, metered lane, metering plan-level tables, signal control, scheduling, block objects, and on the right, you have for each one of those a functionality that is a set of design data available in section three. That is called MIB, management information base. It has some 80 plus objects organized in section three.

Section four: RMC Block Object Definition, is something new in this version two. The block objects are actually a string of objects put together so that the information can be transmitted to the device in one shot. It is designed to gain some level of efficiency and that's what this thing is about.

Annex A is very important for RMC unit because it does describe how the RMC unit actually works/operates. We get a lot of information about RMC system through Annex A.

Here's an example of a metering plan that's provided in this slide. This example basically tells us about how the metering unit will provide the table that will be listing the metering plan and levels, and several levels of plan. For example, it will contain different occupancy level, flow rate, and speed. These are the three parameters that we have been using in this standard that must make up a level, and then eventually, the levels go into plans, and then the plans are related to the metering rate. How many vehicles we allow in terms of per passing per green, that will generate a certain amount of metering rate during the hour. For example, there could be 500 vehicles per hour passing through that.

Annex B is a listing of conformance groups. Conformance groups means that there are certain objects put together so that a particular requirement is met. There is a listing of conformance groups provided in Annex B, which we'll be using to prepare our requirements for this module. Annex C is a tree diagram. It's an organization of how the structure of this MIB is related to each other. There are some 80 plus objects shown in that diagram. Annex D is for future test procedures. There is really nothing much right

A309b Understanding Requirements for Ramp Meter Control Units Based on NTCIP 1207 Standard v02

now in Annex D. Annex E is a documentation of revision descriptions. What we have done so far, there's some evidence of changes in the documentation.

What is not offered by the Standard v02? As you are all aware, NTCIP standard provides basically user needs and requirements descriptions listing. While this particular standard is not SEP-based, and therefore it does not provide us a listing of user needs. We have to identify user needs and write them as we discuss in A309a modules. We have to do similar things for the requirements part of it. Requirements are also not available for this particular standard, so again, we have to identify them and write them using certain criteria that we'll be discussing. Both user needs and requirements are not available for this standard.

Also, traceability tables, there are tools that offer us some level of putting your specifications together, they're also not available. For example, a protocol requirement list, or PRL allows us to trace user needs to requirements, which is not available, so we have to prepare one for this particular project. It's the same thing for RTM. RTM stands for requirements traceability matrix. It connects requirements to the design elements of the objects. That is also not available and we'll have to learn how to prepare RTM for our particular project. We'll be discussing that in learning objective 3.

The definition of a requirement, what is a requirement? This is a definition we are using in NTCIP 1203 DMS standard, which tells us that a requirement is a description of a condition or capability to which a system is obliged to conform; either derived directly from user needs, or stated in a contract, standard, specification, or other imposed document. Basically, it is a desired feature, property, or a behavior of a system. That is also one of the definitions from ISO, which is included in module A102, which came before that also says that a requirement is a translation of user needs and a very concrete description of that. There are requirement definitions out there that helps us to understand what's behind our effort in here.

The relationship between user needs and requirements, one user need could require one requirement, and it could be a very simple relationship that could exist. On the other hand, one user need may require more than one requirement. It could be one, or ten, or five, or three. We don't know, so we have to establish this relationship as per our user need. Sometimes, many user needs probably could be met by one requirement. Those situations are also out there, which make these relationships align.

Let's take an example here as an illustration. A user need is a very simple statement. It says, "TMC needs to control RMC functions remotely." It doesn't have much description in it, so we translate that into a requirement using the definition we just discussed earlier and we'll see what it looks like. We'll bring in the elements or players one by one. It says that the RMC unit shall allow. We are using the language "shall" to stress that this is a

A309b Understanding Requirements for Ramp Meter Control Units Based on NTCIP 1207 Standard v02

concrete requirement. The RMC shall allow the management station on the right, which is the actor, to request implementation of an updated metering plan using communications command source. We have three elements here. The specific target is the RMC unit itself, and you have an actor, the management station here on the right, and then action taking place through the management station command request, and that goes out and actually implements how many cars can pass through or how many vehicles can pass through that signal.

RMC requirements are to be identified from several sources. We have SEP-based standards done over the years. For example, we have ESS or DMS, so there are certain requirements that we have learned from those standards that could be helpful to us, and we will be making use of those. We also will discuss user needs in A309a module. Also, those user needs can be translated into requirements and develop requirements based on that. Also, in NTCIP 1201 v02 Standard, Annex A and B are very important. Annex A provides a description of our operational needs requirements, so we can build our requirements on that, and also Annex B, which provides conformance groups listing, which are related to particular requirements, so we can use that. Finally, we also have certain aspects of configuration, control, and monitoring any and all devices in the NTCIP arena, particularly configure, control, and monitor, so we have learned certain perspectives and methods of how to handle those requirements. We can put all these four sources together and identify your requirements and prepare ourselves of the listing of requirements for RMC system.

We use certain criteria to develop the requirements, and we have to go through the structure of a requirement by providing this stress. The actor is the one that identifies who is making the request or who is requesting the action. The action itself identifies what is to happen. What are the devices supposed to do? The target is the device. It identifies who or what receives the action. In other words, in this case, RMC will actually implement whatever action is required from the actor, which is the traffic management station. Also, the requirement has to be within the constraints parameters. The requirement must identify how it's going to measure a success or a failure, and if there are certain localization type of situation, that may have to be implemented through this structure.

Applying certain characteristics is very important when we write or form the requirement. First of all, a requirement must be a necessity. It must be useful and traceable to at least one user need. It has to be a necessity. It has to be very concise. It should be done in a very concise, simple, understandable, and the "shall" language. It should be attainable, very practical to achieve it. It cannot be a wish list type of thing, which may be impossible to deal with, and it has to be standalone. The requirement must speak about itself very clearly and completely in one place. You don't need two different requirements to make one understood properly. It has to be very consistent, so each requirement must not strike against each other and these two stand on their own without harming the intent of

A309b Understanding Requirements for Ramp Meter Control Units Based on NTCIP 1207 Standard v02

each. Also unambiguous, very clear so the interpretation is proper and it's not ambiguous in a sense. Verifiable, eventually, every requirement has to be verified, so it needs to go through the inspection process, analysis to demonstrate that we have achieved. It has to be verifiable and that process should be also kept in mind.

How do we apply these criteria? Let's look at three different types of requirements where we can actually apply the criteria we just talked about. First of all, the architectural requirements are already in front of us. This is the way we talk to a device, communicate with a device. We call this general communication capabilities. That is what this discussion is about. SNMP interface is an example of what the architectural requirements are about.

Data exchange requirements, these are the support requirements that are necessary to control, communicate the action to the device, and features and function, what the device will do. Those requirements are called data exchange requirements, an example being traffic responsive. Traffic responsive means that based on the current condition, the device will implement a change in metering rate. That's a simple example that will indicate that there is a need to exchange data with a device.

The supplemental requirements are few, but sometimes they do exist. They are the ones which are not met by the above two, for example, architectural requirements or data exchange requirements. If certain requirements are not falling in those categories, then they could be handled as a supplemental requirement. Particularly, for example, as shown here, you may have a local project that may be very specific to a particular region or conditions that already exist, and those requirements can also be classified as supplemental requirements. These three requirements types are identified as architectural requirements, data exchange requirements, and supplemental requirements.

Then we need to organize these requirements. Here is a sample format that we can follow in this module. We are recommending Section three for RMC unit requirements. It could be any number, but for consistency and continuity with the standard like ESS or DMS, we have chosen here to show you that this is possible. Section three couldn't be done by RMC unit requirements. Section 3.1 will provide a general background information. Section 3.2 can provide architectural requirements we just talked about. In there, we have subrequirements, 3.2.1, which is to provide live data, and 3.2.2 is to provide off-line logged data.

We move on to section 3.3, data exchange requirements, they're all listed in this particular section as the subrequirements 3.3.1, 3.3.2, 3.3.3. This gives us a way to organize our requirements in a very systematic and organized fashion. Section 3.4, supplemental requirements are also listed. For example, 3.4.1 is a supplemental

A309b Understanding Requirements for Ramp Meter Control Units Based on NTCIP 1207 Standard v02

requirement. Maybe there is a particular traffic controller that already exists locally, and you may have certain requirements related to that, so you could possibly list supplemental requirements under 3.4.1 and that's what this example provides. There is a detailed discussion about how to organize requirements in your student supplements, so we will urge you to go to the supplement and look at it and see how you can use it for your particular local project.

Here's a list of user needs that we have derived in module A309, some 15 or 16 different user needs that are identified in that, and we can build our requirements based on what we've actually already done in module A309. Again, I will urge you to go back to the student supplement and look at these needs and see how they are connected. Basically, you can see here, user needs 2.1 was identified to provide live data exchange, and 2.2 was to provide logged data exchange, and 2.3 to provide capability to retrieve RMC identity. These are examples, so now we have learned how to translate these user needs into requirements, and that's what we need to learn in this module.

Architectural requirements, as we had discussed, here is an example that says to provide live data. Translating that to a subrequirement here, we now have structure using the criteria we discussed earlier. The RMC unit, which is a target, shall allow the management station, which is the actor, to retrieve data, which is an action, from the RMC unit. Using the criteria we just discussed earlier, we have now formed a very concrete statement here. This concrete statement is standalone, is concise, is a necessity, is attainable, and all of those things are apparent in this particular statement, which will now remove the ambiguities and very clearly present what this RMC unit is supposed to do or is expected to do.

Another example is 3.2.1.2, which is to deliver data. Using the diagram underneath, we can understand that RMC units shall allow the management station to deliver data to the RMC unit about configuration, commands, and control kinds of activities or features. The management station is now capable of delivering data to RMC unit, and this is what is reflected in this statement, which is a very well-formed requirement. It has an actor, it has action, it has a target, and is a very stand-alone necessity statement. To provide the live data monitoring, controlling in the systems when we are connected with the communication linkage or channel.

Let's take another example. We also, if not very often, sometimes when we lose communication to a device, the device continues to work and collects data. We want the capability through this requirement. We want to make sure that we are able to retrieve logged data from the RMC unit when the situation exists for off-line conditions, when the communication is lost to the RMC unit. Let's look at this 3.2.2.3, retrieve logged data collected during off-line situations. In that sense, we're saying that the RMC unit shall allow the management station to retrieve one or more available logged data from the

A309b Understanding Requirements for Ramp Meter Control Units Based on NTCIP 1207 Standard v02

event log gathered during certain off-line conditions such as, as I mentioned earlier, loss of communications. We should be able to do this retrieving when the communication is restored to the device, or when we establish a dial-up link. This requirement is saying very clearly that the management station must be able to do what is required of it to do in the sense that the RMC unit should allow it, so this statement has all the criteria for a well developed requirement.

Once we are done with retrieving the data and downloading the data at the central, we can clear the log. We have no need for having the same set of data at both places. The 3.2.2.4 requirement says that the RMC unit shall allow the management station to clear any or all log entries of a given event class. Now, we have a concrete path to remove the data that's not supposed to be anybody's to have, so we can remove it and make room for its future event log. This is very a nicely done requirement that is being used in RMC unit configuration and control and retrieving.

In terms of data exchange requirements, section 3.3 will deal with that. In A309, we identified data exchange user needs, so now we have to translate those user needs into requirements. There are three categories here. One is the configuration. Configuration means setting the metering table, for example, and it's very important in RMC operations to have a metering table, which has numeral plans and labels and so on, so that configuration, once that is set up, must be done through the configuration requirement.

The control is the control of the functionality that RMC does in terms of operations. What is it that we expect it to do? Whenever we want to communicate to the device, we need the control requirements apparent, and so, we need to make sure that actually is provided. Also, monitor all NTCIP devices, including RMC units are constantly monitored, so we need to know what the status is of the device at the particular time, how many detectors, data, ratings available, for example. Such things make us require data exchange requirements. We can understand from this diagram, on the left we have a traffic management center. On the right, we have RMC unit in the field. In between, we have center-to-field communication. Basically, we are gaining the capability of communicating to the RMC from a central location, and able to configure, able to control, and able to monitor the RMC device.

Those categories will produce a certain number of requirements which are necessary. For example, data exchange requirement for a configuration, let's just look at this. What we're looking at here is this a well-formed requirement? Is everything present in this requirement that makes it a very concrete requirement? The management station shall be able to retrieve information about the configuration of the RMC unit. We have now provided the actor, the management station, who's making the request, and what is the action? To retrieve information is the action. Who is the target? In this case, it's the RMC unit.

A309b Understanding Requirements for Ramp Meter Control Units Based on NTCIP 1207 Standard v02

Then, we are applying the criteria or characteristics in this case. Is it necessary? Of course, it's necessary. No device in the field will work if you don't configure it, so it's necessary. This requirement reflects that. Is it attainable? Of course, we have been implementing devices in the field, so such requirements are already successfully met. In this case, yes, it is attainable. Is it standalone? Obviously, the statement is very clear, and it tells us what is expected from this requirement. Is it verifiable? Yes. If the device will tell us eventually whether it's functioning properly or not as intended, that will be the verification process. All told, this data action requirement is very well-developed here in terms of configuration management.

Let's look at another example. In a similar way, in this case, we are monitoring and controlling the RMC unit, so we have elevated the functionality to a different level. In this case, it's operational whether we are achieving through the RMC control and monitoring what we expect to do from a central location, so let's just look at it and see how it's well-developed using criteria.

The central system, which is the actor, shall be able to make a request to the target, in this case the RMC unit, with the communication command source (priority 2), which is what happens in terms of the central system communicating to the field device to exchange data, which is the action to inquire about the current configuration. We have completed the players' role in these things, the actor, the target, the action, and we are also making sure that it reflects the necessity. Is it necessary? Of course, it's necessary. It's a central requirement here. Is it attainable again? The answer is yes, it is attainable, and standalone clearly, and then it's verifiable. Again, we are applying the criteria that we discussed earlier and characteristics for a particular requirement and see if it's well-developed or not. We are learning, in this module, how to do that. I'll be repeating this theme again and again because this is really what we need to do because these requirements don't exist. Nobody has done them for us, as I said earlier, so now, we have to do that, and to develop these requirements, we have to use the criteria that we discussed earlier.

Our first activity. Which of the following is a well-formed requirement? There are four answer choices here: a) TMC shall be allowed to retrieve the plan from RMC unit, b) TMC needs to retrieve information from the RMC unit, c) TMC needs to monitor metering operations, d) the RMC unit shall comply with all requests. The correct answer is a) TMC shall be allowed to retrieve the plan from the RMC unit is the correct answer because the statement does meet the criteria. There is an actor, action, target, standalone, necessity. All those things we just discussed are present in this statement. Answer b is not correct or is incorrect because this is a user need, not a requirement. It's the same thing with answer c. It is also incorrect because it is a user need, not a requirement, and answer d is also incorrect because the statement is not well-formed as it does not include actor,

A309b Understanding Requirements for Ramp Meter Control Units Based on NTCIP 1207 Standard v02

action, and not standalone. We have learned from this example here how we can actually go back and check and see if the criteria is present or not in each one of these statements.

What we have learned in learning objective 1 is that we know how to develop requirements using the 1207 Standard. We reviewed the structure of the standard, what's in it, what is not available, what we need. We also reviewed three different types of requirements that we can identify from various sources, Annex A and B. We also learned how to apply the criteria and characteristics in writing the well-formed requirements, and we also looked at several examples to see if that criteria is applied properly or not.

Moving on to learning objective 2, we will be discussing how to interpret, understand SNMP interface and the dialogues. What the NTCIP objects provide us so we can clearly define a dialogue. OID is object identifier values. We'll be talking about varBinding, PDU (protocol data unit), and there are three sample dialogues that we have adopted from other standards that we'll be using in learning objective 2 to move on to more RMC-related activities.

Achieving interoperability with SNMP interface, we have to understand what are the parts of SNMP interface are? There are three parts that we have listed here, and SNMP itself was discussed in great detail in module C101, so those of you who have not taken that so far, I would urge you to do that because you will learn everything about SNMP and how it's being used and formed, so that we can actually use it for RMC and any other device for that matter.

SNMP basically has two parts, as shown here. On the left, we have SNMP Manager, which is a software module installed at the management station when a previous management system application, for example, ramp meter, or whatever there is the arrangement is at the central station. On the right, we have SNMP agent installing RMC unit itself, which is also a sort of software module. It does the processing for us with the RMC. There are two key parts here, SNMP manager and SNMP agent that you will be hearing about in many modules, as well as during the discussion in the supplement.

In between these two parts, there are messages going back and forth, and these messages have content that relates to action. Messages deliver, to the RMC unit, what it is to do, what is expected from it, and what the instructions are telling the device what to do. Those messages are passing through in this dialogue process between the two parts. The dialogues themselves are considered conversations because dialogues are conducted in both directions, from the management station to the agent, and then the response will come from the agent to the management station. These are the three key parts of SNMP interface that we'll be talking about in this particular learning objective.

A309b Understanding Requirements for Ramp Meter Control Units Based on NTCIP 1207 Standard v02

There are three types of SNMP messages at the top, and the fourth one here is a response. Basically, there are four SNMP messages that we need to understand. Each one of them has a distinct task or purpose, and they will perform accordingly. The first message is the "GetRequest" message, which is a READ operation. It allows us to read a particular object in the device, a particular condition, particular data in terms of what the device is doing. This GetRequest is a READ operation. Similarly, GetNextRequest is also a READ operation, and basically, it's the same as GetRequest, except it allows us to do more elaborately and more efficiently retrieving more data. Both of these, GetRequest and GetNextRequest, are basically READ operations designed to provide the retrieving action from a device.

The third one is SetRequest, which is a WRITE operation. It actually makes changes as needed in terms of control, modifications inside the RMC device. It will actually alter. For example, it will change a current metering plan from number three to number four, if that's what you really want to do. SetRequest is allowing us to make changes, modification, through the writing process inside the action of the RMC.

The last one, GetResponse, it actually is a response coming from the device, and it's a reply. It forwards you the status, so if it is coming from a device to the management station, and every time you do any of those three top messages, the response will come through this format. There's only one format for all three top messages. GetResponse is a common format that we are using in SNMP here in RMC.

Each message contains a PDU. All of these messages have a particular PDU, and again, PDU stands for protocol data unit, which carries the message to the device. What's in the PDU? We need to understand what is the content of a PDU? We said that a PDU actually does the delivery process of the device, so now, we need to know what's inside in the PDU. How do we do that? The structure of an object helps us to understand. We are now reviewing here the maximum number of meter lanes. For example, this is the name of the object, 3.4.1.1. The structure gives us a definite way of looking at what it is that we are actually looking for or actually need, so that we can put a PDU together. For example, take a look at where it says, "Maximum number of metered lanes." For example, how do we know how many lanes we are metering? In this object, there is a provision for us to look at the syntax, the value. From 1 to 255, is the capability the standard is offering us, and we only need maybe 2, 3, 4, or 5. We're not going to have an impossible number like 10. Let's just say there are three metered lanes out there at some location on a ramp. We now have three, so there would be a number three here, which is a positive integer. This will be called "value." That's one.

Now this object has its own identification number here. This organization is provided us the scheme of things, and this is called OID, object identification number. Now, we have OID, so value and OID form a pair as a varBinding. This pair is appearing as a

A309b Understanding Requirements for Ramp Meter Control Units Based on NTCIP 1207 Standard v02

varBinding. So var stands for variable, and binding is that they are stuck to each other. They cannot be separated. Every object has its own set up here that cannot be separated. Sometimes also, there is a varBinding list down here, and that means that there are more pairs of OID and values that follow each other within the PDU. What we have gained from this structure are two important things, value and OID. Together, they take us to the picture of the function that we are trying to alter, or retrieve information about. In this case, how many number of lanes are out there, so we'll be using three as a practical example to understand what the value range is. Very often, we call the syntax a range; 1 to 255 is a range. Users need to select the range and many in specifications, you probably have seen the range appears as a numerical number.

Now, we move onto the generic SNMP dialogues, generic because they are used for all different devices, so really, there's no restriction in how we can use this dialogue. We have adopted these three dialogues from very successful implementation standard 1204. In this case, Annex, section 3.5 in that 1204 standard, we have three generic dialogues listed for us, F.3.1, which is the SNMP Get interface to retrieve data from a device, F.3.2 SNMP Get-next interface retrieves more data, and F.3.3, SNMP Set interface sends data to the device. These three generic dialogues are making it possible for us to conduct our operation from the management station through the RMC. Communication interface is developed. The first message will be a request message from a management station, and the response will come back from RMC to the management station. That will complete your dialogue. A dialogue has two messages in it, the outbound and the inbound, if you want to say it that way. We have adopted this format from ESS standards, and now, we're going to use these dialogues to conduct how RMC unit can be managed.

As a first example here, our interest is to know how many lanes are out there, for example, as a requirement. This requirement speaks very clearly about management station retrieves metered lane data. We want to know how many lanes are being metered. This will happen through a GetRequest. F.3.1 is a generic SNMP Get interface dialogue that we just talked earlier about. We'll make a request through a GetMessage, which will be part of the PDU, which will have a varBinding list that will say what it is that we are trying to retrieve. This process will start with that request message. It will go to the RMC unit, and a response will come back and says that this is the information that you requested. Practically speaking, we want to know whether one lane is metered, two lanes are metered, or three lanes are metered, so when we send the request about how many lanes are being metered, the response will come back as one, two, or three, whatever that number of lanes being metered. This has a very practical approach to conducting these dialogues for the required information.

Similarly, the requirement here says that the current metering plan is now required to be changed to a certain other number because the condition in the field has changed, for example. A proper way of saying this thing is that we want to modify or we want to

A309b Understanding Requirements for Ramp Meter Control Units Based on NTCIP 1207 Standard v02

change something, so immediately, we are turning to generic interface dialogue 3.3, which is a Set operation. We're going to make a request and say, "We want you to change plan number to this one," so this varBinding listing on the PDU will take that message to the RMC unit, and RMC unit will implement that plan number, whatever the number is here in that value, and then it will send the response back to the central management station and say, "I did what you asked me to do. You asked me to change plan number three to seven, and I just changed plan number seven." The current plan in operation right now, at the moment, is number seven. That is what the management station expected RMC unit to do. We're now able to complete our functional operation to meet our current need in the field. This is the way to control the RMC functionality. For that, F.3.3 is used, so whenever you have to change something in the field because your conditions changed, Set generic dialogue process is the one that you'll be using.

Our activity here for this learning objective. Which SNMP interface will modify the current metering plan? A) Get interface, b) GetNext interface, or c) Set interface? Let's look at the correct answer. The correct answer is C. Only Set operation modifies (write) the current metering plan. We discussed earlier that Set is the only of the three commands that we have that will modify RMC operation or functionality. Get will not be used because the Get is only for retrieving information. It's just a READ operation. It cannot modify anything. It cannot change anything. It only reads the current condition. Similarly, b answer is also not correct. GetNext interface is also not able to write anything. It only retrieves. In this case, GetNext interface retrieves more data, multiple data from the RMC unit. So, we just reviewed the types of dialogues that we use for proper purposes.

Let's summarize learning objective 2. We have responsibility in understanding how we gain the interoperability and vendor-independence using SNMP interface. For that purpose, we have now discussed SNMP interface and dialogues for communication to the RMC device. We also reviewed the structure of the NTCIP object, the value, and the OID. That is what forms the message content. We discussed about varBinding value and OID as a pair for each different type of message out there to the RMC unit. We also developed sample Get, GetNext, and Set dialogues.

Moving onto learning objective 3, we have to now prepare traceability tables for RMC. As we discussed earlier, we don't have these traceability tables for this RMC unit, so now we have to develop them. Protocol requirement list, or PRL in short, is a table that connects user needs to requirements. We'll discuss the benefits of PRL. It's the same thing for the requirements traceability matrix, RTM. RTM connects the requirements to the design and we need to understand how to do that and prepare for a project. We'll also discuss the benefits of RTM.

What is traceability and how is it achieved? Traceability, you have heard this phrase many times earlier in different modules. Traceability is the ability to follow or study the

A309b Understanding Requirements for Ramp Meter Control Units Based on NTCIP 1207 Standard v02

logical progression among the need, requirements, and design details in a step-by-step fashion. These three things are always in front of us in any NTCIP specification, including RMC: the user needs, the requirements, and the design of the objects. These three things need to work together and that's what the traceability is about.

User needs to requirements is done through the PRL. Requirements to design is done through the RTM. Both tables are distinct and they both are useful for preparing our specifications. PRL and RTM for RMC are not available. We are repeating these things because this standard was not prepared using SEP, so we don't have the benefits of PRL and RTM, so we have to develop them. Both the ESS standard and DMS, for example, they have these tables available. That gives you an idea how we have to deal with RMC.

Here, we have to develop using the format that we adopted from ESS. It's a very good successful standard, and we will make use of it, the RMC as well. Here is a PRL format that we adopted from SEP ESS standard. Basically, the arrangement here is the columns and rows. The columns are identified here in yellow in the first row. The first column is user need ID, and the second column user need title, followed by functional requirement ID, followed by functional requirements, and then conformance, project requirement, and additional requirements if any. These columns are here to stay because we have adopted them for consistency and continuity. We urge you not to change the columns' names or the sequence and just follow this format to prepare. However, you can alter the RMC PRL in the rows. You put your own requirements accordingly. Suggesting that, we have now moved on with the continuity and the consistency throughout the NTCIP implementations.

With that said, the example here shows that user need 2.1 in column one, with the title of user need "RMC configuration" in column two, and 3.1, so you can notice that now, all functional requirements are listed in section 3.0. In this case, the requirement appears as 3.1. We have now followed the organization we suggested earlier in learning objective 2.

RMC configuration is M, which means it's mandatory, if this functional requirement is mandatory or optional, so M stands for mandatory, and is the project requiring this functional requirement and the answer is yes, that will appear in project requirements. If there are additional requirements for a particular project, you can list them in here. This is basically a format of an example of how to fill this table in with the requirements to help us.

A table that maps user needs to requirements is PRL because you can see in the first two columns, we have user needs, and corresponding requirements are in the next two columns. This table's definition is emerging right here because now, for every user need, we have one or more functional requirements listed. If one user need has more requirements, they will show up here. There is room for this relationship to emerge in

A309b Understanding Requirements for Ramp Meter Control Units Based on NTCIP 1207 Standard v02

column one, two, three, and four.

Next, we'll be looking at conformance project requirements, and if there are additional requirements. M, as I said earlier, stands for mandatory. Conformance and project requirements work together, and they categorically will declare saying that this particular requirement is mandatory. Nobody has anything to say about it. Specification says you've got to have this requirement met. Project requirement just reinforces what the conformance column says. If there are additional requirements, they will go in the last column.

This PRL forms the agency's RMC specification. We want to conclude this particular discussion on the format. What we are saying here is that we are adopting a PRL based on the ESS standard, and we are following these column arrangements so that we can populate our requirements in rows.

Let's look at the steps for a little clarity here with how we actually develop RMC system PRL. In step one, we will use actual user needs that we developed in module A309a. We'll bring the listing here and list in column one and two. We have some basis to begin our step one. Those user needs will come in very handy, and now, we will put them in column one and column two.

In step two, we will develop the requirements as per this module, the one that we are discussing now, and we've already said that all RMC requirements are going to be organized under section three. We discussed that earlier. Beginning with 3.1 to wherever it ends, 3.N, we will enter all those requirements in column three and four. In one and two, we put the user needs. In column three and four, we will put the requirements. That step three will now make it very clear that it is required, which means it's mandatory, or it's optional. There are only two ways we can put that message across. If it's mandatory, it means the implementation has no choice. It must be met. M stands for mandatory again, and O stands for optional. That information will come in column five, conformance, followed by "yes" or "no", which is a reinforcing mechanism, which is always a good thing to do, and that should go in column six, project requirements.

If we also have any specific requirements in a particular project, then we should put that in the last column, which is the additional requirements. That will be our step four. Sometimes, it's not necessary, but since the table allows us to reinforce our local condition, we should be taking advantage of that.

Now, how do you organize the user needs and requirements together in a PRL? These two columns here, actually the first box on the left, lists all the user needs that we developed in module A309a, and the box on the right lists those three types of requirements. Organization we are following very strictly in the sense that we adopted

A309b Understanding Requirements for Ramp Meter Control Units Based on NTCIP 1207 Standard v02

this organization, so we might as well make the best use of it. Section three lists all the RMC requirements, and section two will list all the RMC user needs in a particular RMC specification.

This is an example of a project PRL that actually maps the RMC system requirements. This is a fully populated table here, a PRL. Again, you see that in column one, user needs are listed, section two, with the titles in column two. Column three has the requirement ID, section three requirement titles. Each one of them has a title. Some of them we already discussed earlier. This requirement was discussed earlier about providing live data, off-line data, general configuration, configuration of a device, meter lane, how many meter lanes for example, and queue override. There are different requirements listed here, and then next to that is conformance. Not all of these requirements are mandatory. You can see that "provide live data" is mandatory, so it's reinforced by saying, "Yes," it's required in a project. These requirements are always followed by whether they're mandatory or optional. And then, we are following up in the last column with particular reference. For example, in general configuration here, we're saying that additional requirement is asked for in NTCIP 1207 v02. By the way, make sure that please spell our version number here. Don't leave it 1207. You must make sure that we are referring to version 02 and Annex B, conformance group B.3. This is a very specific requirement that you are placing on this particular requirement. You want your configuration done according to this requirement. The person who reads this table, the PRL, is going to make sure and go through this particular section and read what B.3 means. This is a great way to reinforce again, for the second time, what your real requirement should look like when you actually design it into design objects. All those things are done in a way that you expect for your RMC to work.

Let me also tell you that certain requirements are not necessary, and they should not appear if they are not required for your specification. How do we know that? Take a look at 2.5 for example, queue override. Understand that this is optional. It's optional because not every agency, according to standard, is going to need to implement queue override. Certain agencies have internal policy to allow all the vehicles together for a demand to go through. It's called flashing the lamp, for example. Some agencies may not prefer to do it that way, so in that case, you have to choose optional. If it's not required, you should not put it in as a requirement because it's not needed and it's going to be difficult to implement and so on. That is a way to look at PRL. PRL reflects your real requirements. As I said earlier, by the way, the last column, you can modify according to your local needs.

To close here, in terms of project specification, by choosing M for mandatory and by choosing "yes" for project requirement, you are doing a closure to your set of requirements, which is very important. PRL adds this particular value so that it's understood in a very proper way by the developers.

A309b Understanding Requirements for Ramp Meter Control Units Based on NTCIP 1207 Standard v02

The benefits of project PRL for users, there are at least three project benefits that we should be aware of and then why we have been struggling to do one for RMC. Well, the project PRL shows relationship. For every user need or feature, we have a requirement or capabilities connected to it. We gain the benefit of why a particular requirement exists or shows up in the specification. We can use the PRL as a checklist to validate the process, and eventually, we're going to say, "Does the system meet my needs?" We're going to make sure that somebody looks at this as a checklist, and then actually goes through one-by-one and really makes sure that the requirement is actually used or the user need is actually met in this case. PRL states the value of user needs traceability very, very clearly.

Within the agency, a PRL forms a basis for potential interoperability. For example, we don't normally do everything in one phase. Sometimes, we do maybe doing ten this year and next year, we will probably do ten more. Our implementation varies year-to-year, budget-to-budget, your cycle, and also in vision. For example, we may be doing part of the vision and then we go back to another part and do that. How do we keep track of this interoperability issues in the different parts of the ramp metering system or previous management system? We use PRL so that the basis for all these different implementations will be remaining the same.

The users, developers, and vendors collectively, the PRL connects all parties together on one common platform, so one level field, for example, as we might say. That itself eliminates guesswork. Everyone knows what is expected from the user needs in terms of requirements and therefore, the guesswork is eliminated. Eventually, overall aspects of the PRL benefits are that it eliminates or reduces the risk. It may not eliminate the risk completely in a general sense, but it certainly helps us to reduce the risk of failure to conform to the standard. And remember, our main purpose in this module is to make sure that you implement RMC according to what the standard is requiring you to do minimally to conform to the standard. With that said, the PRL provides us a lot of benefits.

Let's move on to the requirements. The requirements associate the requirement to a design object. As I mentioned early on, there are some 80 plus objects out there in RMC. Now, we don't know which object is related to which requirement in a general way, but if you look at the RTM, we will be very happy to say, "Yes, I know which requirement requires what object." This relationship is coming up very clearly in RTM. RTM associates each requirement first to standardized dialogue. Dialogues appear in the second column. The first column is requirement ID 3.1, 4.1, and so on. The second column dialogues and we adopted this F.3.1, F. 3.2, F.3.3 again from ESS standards. Each of these three dialogues are referenced in the second column. That's what RTM does. In the first and second column, we are done with the dialogues. Then in a third column, you have the title of the requirement itself, and then in the fourth column, we are suddenly using the

A309b Understanding Requirements for Ramp Meter Control Units Based on NTCIP 1207 Standard v02

object. Now, we have a section listed in object. You're going to need the name of the object, what it is, because we're going to form a PDU in the dialogue process. Immediately, we're going to need the name, title, OID of the objects. For that, we have to move to 3.2.1, which is the section number. In standard 1207 v02, we have 3.2.1 identified for a particular object. That's a design. We're now putting the dialogue and the requirements together. The last column, the additional requirements or objects, again it lists the clause. It will always be good to list the clause so that it reinforces. That's what completes the RTM structure. These two arrows shown here in the red, you can see that one leads to the dialogue in the second column, and then the other arrow leads to the object ID or section, which is the design that we are looking for. RTM brings requirements and design objects together.

Let's look at the steps for developing an RTM for RMC system. The table is the same as the one we just looked at. Step one is requirements, which you enter in column one and three. We have already developed requirements earlier, so one-by-one, whatever are the requirements we listed earlier we should bring them in this matrix and start populating it. That's step one. Step two is generic dialogues. We have only three, 3.1, 3.2, and 3.3, so we should put them in column two. Step three uses the Annex B. The Annex B is the listing of conformance groups, which have certain objects associated with that particular requirement, which maps directly. We bring them here. In this case, 3.2.1 is listed as associated to 3.1. You can see that makes a nice connection. And step four, if you have specific project requirements, you enter that in the last column. In this case, NTCIP 1207 v02 clause, which leaves 3.2.1 or whatever additional things you want to put. That's what completes the four-step process in developing an RTM.

Here's an example of an RTM which is populated to a certain degree. In column one, we have requirement ID. In column two, we have generic dialogues. First, we will use 3.1 as an example, and we list the requirements that we developed earlier, and then we're going to put the proper number from section three of the standard, and put additional requirements as we see fit. Again, I will say 3.3.3 requirement—queue override—just to signify it one more time, is optional, so it may not show up in your particular project if you don't need to do this queue override. With that in mind, we can actually populate this RTM to complete the project-level RTM. This gives us the step-by-step process.

The benefits of the project RTM, for users, it shows the relationship of requirements to the specific design items for the interface, dialogues, and data objects. There is lots of design data, so unless we have an RTM in front of us, we wouldn't know which object is related to which particular requirement. RTM helps us to understand that. The second benefit is a significant one because when we accept the system, we are now able to answer this question, "Did they build the RMC system right?" Did they build actually following particular requirements at the level of SEP lifecycle development? Also, at the end of the process, "Does my interface deliver the desired functionality?" Does the

A309b Understanding Requirements for Ramp Meter Control Units Based on NTCIP 1207 Standard v02

functionality come alive? Do I know that the design has actually implemented all aspects of objects so that every function that I was looking for is performed? It's a very nice way of using RTM to help us answer these two questions.

For agencies, developers, and vendors, in particular, the requirements are traced to dialogues and then to objects. This sequence is very important. This order is very important. This relationship is very important. However you look at these statements, the great benefit that comes out to designers is the peace of mind saying that we have actually connected every requirement to every piece of design data. That is a very neat way of achieving the original objectives of making sure the system is designed properly.

Eventually, when we verify at the later stage, when we test, we inspect, we do all kinds of things to see if the system is actually functioning as desired or accordingly, every requirement is now tested. RTM will help us to extract those requirements and take them to a testing level and create a testing plan and testing procedures, and eventually, write a report. All those things are very important in terms of how a device is eventually tested for its functionality. That's what RTM helps us to do. All testing plans are actually linked directly to RTM also. You will learn that in the next module on testing for RMC.

For our activity for this learning objective. Which of the following ensures precise objects necessary to fulfill a requirement? A) the project PRL table, b) the project RTM table, c) generic SNMP Get interface, or d) major desired capability, MDC? Let's review the answers. The correct answer is b) the RTM identifies objects necessary to fulfill a requirement. This is the only time in the RTM you will see an object showing up in that particular column that we had discussed earlier. Answer a is incorrect because PRL traces user needs to requirements, not objects. Answer c is also incorrect because generic SNMP Get interface does not contain objects. Answer d major desired capability, MDC, is incorrect because it's part of the user needs.

What we have learned in learning objective 3, in terms of understanding traceability, is that we discussed how to develop PRL to trace user needs to RMC requirements. We also reviewed the benefits of PRL. We discussed how to develop project RTM to trace requirements to dialogues and objects or design, and we also reviewed several examples and we reviewed the benefits of the RTM.

Moving on to learning objective 4, which is to incorporate requirements not supported by standardized objects. In that, we sometimes run into a situation where certain requirements come up and the standard doesn't support it. What should we do? We need to understand the context and conditions for extending this particular standard. You also have an example, so we can understand what that means.

Let's understand the context and conditions for extending the standard. The context of

A309b Understanding Requirements for Ramp Meter Control Units Based on NTCIP 1207 Standard v02

missing requirements, why a requirement is missing or what is the necessity of this requirement that cannot be supported by standard? The current version standard provides common requirements at the time of development of this particular standard. What was anticipated is being provided in terms of ramp metering requirements. However, in some cases, some features or requirements may be new to the environment that we have functioning in the country currently. For example, we now have corridor-wide ramp metering. Years ago, we used to be doing one ramp at a time kind of management. A situation may have changed in certain parts of the country, which requires different features that will generate different requirements and certainly the design. How do we handle this particular context?

We have to understand what the context is telling us—what is the need for a particular way of doing things—which is not supported by the standard. There are certain conditions for extending the standard. You can extend the standard and get your project done, but there are certain conditions that we have to follow. First is that any new objects to RMC MIB is possible if they are documented. If only the vendor knows about it, or the developer knows about it, that's not sufficient because everyone has to be aware of that particular new object necessity to meet the requirement. Once these objects are there, they should be available to anyone. There should be no restriction, agency, vendors, developers. These situations, when they show up, they should be handled accordingly.

The second thing is also important and that's to follow the format. An object has a structure that came from language called ASN.1, which is the format we have adopted in the entire NTCIP family of standards. Even private objects cannot create its own format or miss something. If you do have a specific object, which is unique to your own implementation, it must also follow the ASN format. It must comply with the rest of the definitions of the structure itself. For example, OID, that cannot be altered, and also the value has to be consistent in the way we present it in this structure. Private objects impact interoperability and interchangeability. Going in this process of extending a standard, everyone who is concerned with the project should understand two things. There are two unique issues. One is the interoperability and the other one is interchangeability. There will be an impact of private objects on both of these.

Technical conditions, if you do have to extend, what are the technical conditions? Beginning with the association of a design object, the behavior of the device may be affected. You have to understand that any time you have a private or a non-standard design object, the behavior will be altered in a different way. Second, ASN.1 based objects must support READ. The system should be able to READ. SNMP managers should be able to READ. The agents should be able to READ. All of these things are READ operation for retrieval and writing operation, as well as control functions. They should be without restriction. A private object cannot put a restriction and say, "This is the only way it can be done." Those elements will not work.

A309b Understanding Requirements for Ramp Meter Control Units Based on NTCIP 1207 Standard v02

Syntax in the structure of the object must be non-negative. We've been saying that the number of lanes for example, is positive three—one, two, three lanes. To understand that part, it cannot be negative. The communication interface works on a non-negative integer basis and that should be honored. The object must have OID within the current MIB node. The node cannot be changed. RMC MIB is a fixed thing in a standard, so you have to operate within that node and then you have to find the proper OID so the object could be identified in a very neat way.

Finally, SNMP is the only interface that's allowed in this device, and the SNMP interface itself must itself follow NTCIP 1103 rules. There's a whole ruleset that SNMP works under. With these technical conditions in mind, the extension should be carried out if it's needed. There are drawbacks to extensions. Certainly, the interoperability may be compromised. There may be a need not supported by the standard. A management station that is already functioning may be able to do that and a customized management station may not be working in unison with the other management station or control mechanisms. There are priorities also within the ramp metering, so all of these things will be impacted in a significant way.

So to summarize this particular point, we want to say that the management stations that do not support the new objects or extension will not be able to communicate with the other stations, so this capability will be lost. The second point to this particular drawback is that if you're not consistent in your requirements, the requirements imposes interoperability and you're not going to be able to gain interoperability. In other words, every time you have a custom operation, the interoperability is an issue. With the private objects, you require a custom integration, which will impose its own requirement, for example, time, expertise, you need certain type of people to do custom integrations. It will cost you. It may even delay the project and so on. Those factors in managing a particular project will also come up.

Here's an example of extending the standard. This is the standard requirement currently supported by the standard, and this is how the metering levels are created. The metering rate is computed based on threshold for occupancy, flow rate, and speed. Current condition of the current requirement can be met by using this arrangement of measuring occupancy. That's how the tables are set in RMC. Someone says, "Well, instead of occupancy, we want to use density-based approach." This extended requirement may look like this. The RMC system shall support density-based for ramp metering to maintain saturation density. This is the new approach the standard does not support currently.

In the sense of implementation, for example, there are several locations currently in the country that use the density-based approach to arriving at what is the best metering rate. Density, for example, cannot be measured. It has to be calculated and all that stuff. It's a

A309b Understanding Requirements for Ramp Meter Control Units Based on NTCIP 1207 Standard v02

new approach. We don't have a situation and a neat way to handle it. What should we do? This will actually be a custom, or proprietary, or local implementation, which will have its own set of issues, and it will impact the interoperability beyond the project, perhaps in the region, corridor. Those are the issues, again, that you have to think about, but it at least gives you an idea that the standard does not support density. The standard supports occupancy. It is a way to look at how to extend the standard to meet your needs.

Let's have an activity here. Which of the following answers is false? A) an extended requirement is non-conformant to the standard, b) an extended requirement will break the interoperability, c) only SNMP interface is permitted in RMC system, d) the project idea may contain private objects. Let's review the answers. The correct answer is d) the project RTM may contain private objects is correct because the statement is actually false. The project RTM does not reference a private object for an extended requirement. Remember, a project RTM is from the owner to the developer. An owner has returned that this is my requirement in RTM, and there is no reference of private object in that. RTM does not list private objects. Answer a is incorrect because the statement is true. An extended requirement is non-conformant to the standard. If you have a private object, you will be non-conformant to NTCIP 1207 v02 standard. Similarly, answer b is also incorrect because extended requirement will break the interoperability. Answer c is also incorrect because the statement is true. Only SNMP interface is permitted in RMC statement.

Moving on to the summary of learning objective 4, we discussed that the private objects, or extending a standard must be done under the context and change of conditions, and it must meet certain technical requirements if you need to extend the standard. Also, we discussed an example of occupancy versus density in terms of extending the standard.

Now, we move onto our last learning objective 5, developing RMC specification. How do you actually put the RMC specification together in a specification package? And then, we have to probably understand certain issues clearly as a checklist, so we need to discuss those two parts.

Generally speaking, this terminology that you may be aware of or not, I'm not sure, but in many ways, publications have used this terminology to put their contract document together. Plans-specifications and estimates (PS&E), some of you may have heard that. PS&E is actually a contractual package that will be prepared for every project. In that, you have the major element of contractual requirements during the system development, testing, deployment, integration, operations, administrative requirements, and certain things. Years ago, we used to call it boilerplate as some of you may know. The boilerplate meant the basic things, so all those things go on in the specification.

A309b Understanding Requirements for Ramp Meter Control Units Based on NTCIP 1207 Standard v02

That also included hardware specification, functional specifications requirements, performance requirements, electromechanical, environmental departments, humidity, temperature, all of those requirements are listed in a general sense for a particular device or system that you tried to acquire, specifically, software specification, interrelated functional requirements, and performance requirements. What needs to happen and how it needs to happen and all those criteria were listed in number two.

Particularly in RMC specification, the subject of our discussion today in this module, the RMC specification should include communication interface specs, which is basically an SNMP interface that we discussed earlier. We also have listing of architectural requirements, data exchange requirements. We talked about PRL and RTM. These are the minimum requirements that we pose in terms of RMC specification.

Let's look at the RMC specification. What are the minimum sections that we should have in our specification? We should be discussing communication interface specification, SNMP, the architecture requirements, data exchange requirements, project PRL and RTM. Every project should have a PRL and also RTM, so those should be included in these sections.

As a checklist, there are at least four items that we should be very concerned about. First, address the interoperability issues. Second, integrate the project PRL and RTM in the specification. Make them a bona fide part, an inherent part of the specification. In fact, that gives strength to the specification. Then, coordination requirements, there are certain things that we should be aware of in terms of how you coordinate this specification with other things. Because ramp metering is done through the controller, there are certain specific issues like type of controller, also the detector station, so we need to understand those four issues.

Let's look at them closely. Our first issue is addressing interoperability. If you are looking or seeking interoperability in the specification, you must select same user needs and requirements. Same objects, messages, PDUs, and dialogues. This is the foundation for achieving interoperability. You must make sure that this actually is conveyed through the specification. MIB objects, GetRequest, GetNextRequest, SetRequest, GetResponse, PDU, they form the basis for what the statement says. Then it goes into dialogue 3.1, 3.2, and 3.3. We want to make sure that we have capability in the specification to impose any of the dialogues that we need, and pertaining to this particular RMC unit. All these dialogues are adopted so that we can work through RMC unit.

Number two, integrating PRL in the project specification is very important because the PRL defines data exchange requirements for the communication interface, at least everything that we have discussed in learning objective 2 and 3. Then, we move on to the particular standards. Remember, SNMP is only the communication protocol. Then you

A309b Understanding Requirements for Ramp Meter Control Units Based on NTCIP 1207 Standard v02

have the different layers underneath the NTCIP stack that you need to impose at different levels of protocols, such as the transport level. Those protocols are also supposed to be listed properly so that we get the dialogues process completed using those protocols.

Also, references to interface standards must be specific in the version. We cannot say just the number. We have to provide a version number and publication date. This is one of the small issues, but a big issue. It's small because the agency may say, "Well, the developer will know what version we're talking about." Well, we cannot assume that, so we have to be, as the owner to people who prepare the specification, we must be very concerned about the version number and the publication date. That's very important. Also, included the PRL with the objects value ranges. I mentioned several times, the value is coming from the syntax, 1 to 255, and so on. Value ranges for all objects must also clarify the parameters because they are a part of the PDU, and that makes it a very important point.

Coordination requirements, there are significant number of requirements along the way, however, when we do have the communication interface specification, it must be consistent. SNMP was mentioned, so in RMC, we are using SNMP. Such consistencies are very important, and it should be practically reflecting to how the specification of RMC unit is prepared.

We also need to be concerned about the standardized design solutions, specifically in project RTM. We are saying that only objects that are specified in RTM are allowed to be implemented. We make that kind of form statement in the coordination of requirements. We also include PRL and RTM as a source for the design of the system. When we design the system, we use both PRL and RTM and then we test it again using the traceability tables as well. These are the activities that we need to coordinate that the user level, at the design level, and again at the testing level.

The last point here that we need to make is the types of controllers. Remember, ramp metering has come from many decades now, and we used old traffic controllers. Today, a type of modernized traffic controllers have showed up. ASC or ATC are very powerful new controller. We use one '70s type like 2070 and so on. Every type of hardware or firmware will have details, versions, upgrades along the way, so we should be aware of what is the hardware that we are talking about. We need to focus on what needs to be replaced maybe, or what are the technical issues with the firmware. The old controller was doing only pure limited things. Modern controllers do many things. If you write the requirements and write the specification, we need to be aware of what can be done and how it can be done.

We need to understand that there is an absolute need for mainline detector stations. If you're going to do traffic responsive, you must have a detector station on a mainline. It

A309b Understanding Requirements for Ramp Meter Control Units Based on NTCIP 1207 Standard v02

used to be a loop detector station. It could be replaced with TSS or it could be any of the different types of measuring devices that you might have, but there is a real-time need for traffic responses. Without detector data coming in consistently, you will not be able to know what the current condition is on the freeway, and therefore, you will not be able to change the metering rate on the ramp.

The last point here is the detectors on the ramp. While there are several types of detectors, the demand-passage and queue, demand is absolutely necessary. The ramp metering is based on the demand actuation and all that is important for inputting into the RMC. Passage detector may or may not be needed. It's good to have. Passage detector will tell you if the vehicle has actually left the ramp area, and is now going into the merging area. That may be or may not be necessary, but it's up to the local project. Queue detector, for example, is also one of the detector types that we use to know how long the vehicles are lining up on the ramp. Some of these things need to be understood in the practical way of its usage, and it may or may not have implications on a specification, but they need to be looked at as a checklist item.

For our last activity, let's look at the situation here. Which of the following statements is false? A) the project RTM specifies the objects and dialogues, b) RMC unit can be readily replaced with a traffic controller, c) RMC unit can also control an advanced warning sign, or d) SNMP must be specified to conform to NTCIP 1207 standard v02? Let's look at the answers. The correct answer is b) RMC unit can be readily replaced with a traffic controller is the correct answer because the statement is false. RMC is a special controller equipped for ramp operations. It has a specific firmware/software and it cannot be replaced with an intersection controller, for example. Answer a is incorrect because the statement is true. The project RTM specifies objects and dialogues. C is also incorrect because the statement is true. RMC unit has outputs that turns signs on during ramp operation, and turns off when not in operation. We've seen those signs on or off, so that is the reason why this answer is incorrect. Answer d is also incorrect because SNMP must be specified to conform to 1207 standard v02. SNMP is the only protocol we use for the interface at the top level. This exercise has again reminded us of what is important in terms of specifying at the project level specification.

What we have learned in our learning objective 5 is to develop an RMC unit system specification. While we discussed how RMC system specification fits the overall project specification package, we also discussed the checklist of four items that form the key parts of our concerns and issues that we need to be concerned about.

What we have learned in this module? We have learned that the RMC unit standard does not provide requirements, and the user must identify and write them for project specification. Second, we have learned that the requirement is a translation of user needs and has a structure and contains characteristics. We learned that requirements

A309b Understanding Requirements for Ramp Meter Control Units Based on NTCIP 1207 Standard v02

are linked to interoperability, and vendor independence. At the level of a project, a particular project, we also specifically learned that each requirement is traced to at least one user need in the project PRL. We also learned that the requirements should be traced to objects and dialogs in the project RTM. We also learned to retrieve data from reading operation from the RMC unit device SNMP GET interface is used. To control an RMC unit, writing operation, we learned that SNMP SET interface is used. Finally, to support the same features, the management station and an RMC unit device must have the same MIB, management information base, and must use the same dialogs.

Here are some of the resources that we can use for this module and related discussion that we also had in the previous modules. The student supplement has a lot of different details of requirements of the PRL and RTM, also criteria discussions. We have additional sources that you can use, conformance listing, so please utilize the student supplement. We have all documents that we have cited here in this module are available at www.ntcip.org. 1201 v03 Global Object Definitions standards, 1207 v02 RMC standards, also NTCIP Guide 9001 document is very good. It has a lot of good discussion about objects and its movement within the development process. Then we have several modules that we already have available, which I will remind you again. Module A103 talks about how to review and write well-formed requirements. A203 writing requirements and details, and also A309 module, which preceded this one, on RMC user needs. So you have pretty good lineup here for development information of RMC.

The next module that will be available in this series is T309: Applying Your Test Plan to the Ramp Meter Control (RMC) Units Based on the NTCIP 1207 v02 Standard. This will take the requirements from this module and show you how to actually develop the test plan so that you can actually verify whether the RMC system is being built right and you get the desired functionality you are looking for. That's what you will learn in this module.

Some of the questions that have been raised in the past. For example, can I use ATC? I have right now 170 controller and the answer is yes. The second question could be also answered in a similar way. We are now doing corridor wide operation and we are collecting data from the TSS, which is the primary source of collecting conditions for congestion management in the corridor. Can we use those data also for ramp metering? And the answer is yes, if you make the data available to the RMC unit. The way the standard is working right now in the current version is that it drives the data directly into the RMC unit. So the current condition data should be available at the RMC in the field. It's not sufficient to have data available at the center. It should be available locally. Some of these questions are pertaining to RMC. We've also discussed about the limitation of occupancy ways approach to developing traffic responsive strategy, so that's also a question that people may have in these regards. These are some of the technical issues

A309b Understanding Requirements for Ramp Meter Control Units Based on NTCIP 1207 Standard v02

that may be relevant to today's module.

Thank you for taking this module and this completes the discussion today.

[End of Audio]