

A315b, Understanding Requirements for Actuated Traffic Signal Controllers Based on NTCIP 1202 Standards, Part 2 of 2

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 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.

Narrator: Throughout this presentation, this activity slide will appear, indicating there is a multiple choice pop quiz following this slide. You will use your computer 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. This module is A-315b, Understanding Requirements for Actuated Traffic Signal Controllers Based on NTCIP 1202 Standards, Part 2 of 2. Your instructor, Mr. Miller, has more than 36 years of experience in the management, design, and development of critical control systems, including traffic control, bus rapid transit, and connected vehicles. Specific ITS experience includes the design and first deployments of ATC 5201 (Linux) and ATC 5202

A315b, Understanding Requirements for Actuated Traffic Signal Controllers Based on NTCIP 1202 Standards, Part 2 of 2

(Model 2070) ASCs as well as the design and first deployments of NEMA TS-2 ASCs with NTCIP.

Dave Miller: Thank you for the introduction. We're going to start on this slide to go over the target audience. This course is targeted primarily for staff who are responsible for the development of requirements for actuated signal controllers, primarily, and also for staff responsible for writing the terms and conditions for the procurement contracts for actuated signal controllers. Now, others on the line that can benefit from this are manufacturers, traffic management system owner/operators, such as the maintenance and traffic engineering staff that use actuated signal controllers, or ASCs, on a day-to-day basis.

These are the recommended prerequisites that you most likely have already taken before this slide. So we started with the very basic concept of ITS standards, what are they, then the A series we proceeded on through kind of a broad-based standards-based systems, and then on down into like A-203, we talked about how writing requirements, when they have content or not. We're going to go on here beginning with 315a, as you can see here. We got specifically into understanding the needs for actuated traffic signal controllers based on the NTCIP 1202 standard, and then most recently Part 1 of this course was how to specify requirements based on the user needs that we learned A315a. So this is Part 2 of the course, so this is one long course that's been split into two and this is the second half of it. And as you can see here, we went through the curriculum path. This is the knowledge flow of going through all of the courses and, again, down at the bottom in the lower right we are doing A315b, which is Understanding the Requirements for Actuated Traffic Signal Controllers based on the NTCIP Requirement and, again, this is Part 2 of the two-part course because it was such a broad topic.

Going into the learning objectives, the first four of our learning objectives that we're going to do here—we're going to review what we did in A315 Part 1 that you completed as a prerequisite for this learning module. And if you'll recall from Part 1, we learned how to develop requirements using NTCIP 1202 v2 standard, and how to achieve interoperability and interchangeability. We also developed an understanding of traceability and how to develop specifications of a typical use case. So here in Part 2 we're going to do a very quick review of Part 1 just so we can have a transition into Part 2. In Part 2 we're going to learn additional skills that you'll need in real-world situations. If you're actually purchasing actuated signal controllers, if that's what you do, or if you are a person that has to write procurement documents to go out for bid for real-world procurements, we'll go into that in a lot more detail in this Part 2. We're going to identify special issues that you'll encounter

A315b, Understanding Requirements for Actuated Traffic Signal Controllers Based on NTCIP 1202 Standards, Part 2 of 2

when you're doing actuated signal controller deployments. We'll learn how to incorporate requirements that are not supported by standardized objects of the NTCIP 1202 version 2 standard.

So let's pick up on just a quick review of Part 1. So we're just going to go through the concepts of what we learned in Requirements, Dialogs, and Traceability, what we call here RTM, or Requirements Traceability Matrix, to make sure that we're conforming to the standard. So if you look at number one here, requirements can be identified from user needs. You recall that we started with user needs, so that would be the end-user, the owner-operator, whoever's paying for the equipment. We collected the user needs. We also have SEP-based standards and conformance groups in the three perspectives. So this is where our requirements came from. We had user needs, the standards, and the conformance group within the standards. Going on to number two here, we know that dialogs define rules on how to exchange information while objects define the meaning of the information. Both the dialogs and the objects must be defined to achieve interoperability. Number three, recall that we did a traceability table that allows a user to quickly determine why a design feature is included. And the finally, number four, interface specifications should reference the standards whenever possible rather than copying their content. You should do a normative reference to each one.

Recall that we have a lot more information than we can fit into these training modules, so we have student supplements. So there's a Student supplement for Part 1. You can see—I'm not going to read this whole slide, but example text includes a boilerplate user needs, on and on. But what you can get out of the student supplement, you can go into more detail. You can see a partially filled example of a real procurement specification, and each of the items that are shown here need to be included in a complete specification. So if you go into the guide, you'll see an example of a real-world situation.

On Part 2—that was kind of a brief review of what we did in Part 1—but moving on to Part 2, here we're going to learn how to develop requirements using NTCIP 1202 Standard that really defines the structure of the ASC communications to support the NEMA TS-2 protocols. Now, recall that we learned in the earlier modules that you took—we learned that the criteria for well-written requirements to fulfill user needs and standards for interoperability, and then next we learned the definition of interoperability and interchangeability, and that's like how to achieve interoperability and interchangeability by using the NTCIP rules for information exchange. Once the requirements are developed and the NTCIP information exchange rules are used to realize the needs, we also learned the concept of traceability that's used to link each

A315b, Understanding Requirements for Actuated Traffic Signal Controllers Based on NTCIP 1202 Standards, Part 2 of 2

requirement to the user need to ensure that the end product meets the requirements through the use of standardized dialogs. Finally, we learned how to develop consistent specifications, of which NTCIP is one part of the interface specification, which may include custom objects that are not supported by this standard. And then learning objective number five, Managing special issues, specific actually to signal controls, there are a number of ASC interface specification issues that are specific that affect requirements, and some of the key ones will be discussed here in Part 2 of this discussion. There are two additional implementation issues that we really suggest that you go to the student supplement, and that is Communications Loading. It's like traditional deterministic modems versus modern networks, things like clock coordination. Special issues like that affect requirements and can affect how you would write your specification. Those are discussed in the student supplement. We didn't really have enough time to put them in here, but when you read the supplement you'll see some other items that could be pitfalls when you're doing specifications for ASCs that you probably want to know about, and there's some information in there on examples and how to deal with them.

Let's move on to managing requirements. We're starting by managing requirements—what does that mean? Managing requirements can mean different things to different people. In the context here for actuated signal controllers, managing requirements means to maintain the correct association of each requirement to the standards, so you want the requirements to be met by the standards. So correct management means that you're doing the correct associations; standards are applied correctly. For example, requirements must be associated to the correct conformance group within the NTCIP standard, and use the correct implementation of the object definitions within that conformance group. The dialogs must be correct to meet the requirement, and the requirements management should have a manager. In a lot of system engineering projects—I'm sure some of you have been involved in some where the requirements change during the process of writing requirements in the project, and you might end up doing cost overruns or time lapse overruns, milestone overruns. The suggestion here is a good requirements manager will have a person called a requirements gatekeeper who enforces correct management of the requirements on behalf of the rest of the stakeholders. So you learn in earlier modules that the requirement must be well formed, it must be testable, it must be structured correctly, on and on, and sometimes that's lost when you get to this point, when you're actually writing requirements and doing a project, and the problem comes in, well, maybe somebody will send an email, say, "I really wanted to do this," and that kind of grows into, "Maybe we're going to do it," and then the person doing the project puts something in that wasn't planned. The suggestion here is to have one person that acts as a traffic cop to make sure that the requirements come from

A315b, Understanding Requirements for Actuated Traffic Signal Controllers Based on NTCIP 1202 Standards, Part 2 of 2

an authorized source and they're well stated, like we learned in our previous modules. We learned that correctly-stated requirements must be testable. A gatekeeper can reject requirements that aren't testable and send them back to the person that sent the requirement and have it restated in terms that can be testable. For actuated signal control, managing each requirement to the correct conformance group ensures that each requirement can be tested to the correct standardized objects by using standard test tools. There's no question on whether it conforms or does not conform. Now, we're going to find in rare cases a requirement may not fit within a conformance group, but it's still a requirement. There's a user need out there that came from an owner-operator of a traffic management system; you write requirements to meet the need; you look in the conformance group, there's no conformance group object there, and may require a standardized object that should include testing and documentation in the contract terms and conditions. We're going to talk about that a little bit more later, but we've always said to look in the conformance groups, pick the object, write the requirement correctly, and manage the requirement correctly so that you're conforming to the standards, but we're going to see that sometimes you have some requirements that aren't in the conformance groups, but they will still turn up in contract terms, so that's a skill that you really need to develop and know how to handle when that comes up.

We're going to talk a little bit now about performance requirements. Signal control types can dramatically affect performance requirements, again, this is a module on requirements, so we really need to know that there's different types of actuated signal controllers out there. You may get a requirement to be implemented on a system that has some brand-new controllers that are very fast and have internet IP and Ethernet mixed in with others that are on older modems with different types of communication. So the performance of the equipment on the street can really affect requirements. We really want to dive into that here a little bit.

For example, the objects and dialogs used to implement a requirement on actuated signal controllers that have been out there for ten years may be different from the objects and dialogs used to implement the same requirement on a brand-new controller out of the box that has higher performance. And again, we're going to talk about this more in later slides, but we're trying to give you a groundwork here to make sure you understand that you're going to run into issues of older and newer ASCs and different performance requirements, or different performance, that will have an effect on your requirement. For example, legacy ASCs that are out there that are ten years old, the communications may be a private line, a 1200-bit-per-second modem. It may be old enough that it has a microprocessor that was invented probably 12 years ago that has less than 1 million instructions per second, or MIPS—we'll be using that term here for performance—and

A315b, Understanding Requirements for Actuated Traffic Signal Controllers Based on NTCIP 1202 Standards, Part 2 of 2

some examples of that, I'm sure a lot of you are familiar with the type 170s. There's a lot of them still in service. They work perfectly fine. There's some older NEMAs out there, and there's still electromechanicals out there running on the street that we still run into. But if performance isn't specified and there's no Ethernet IP communications, how do you write your requirements? We're going to talk about that a little bit.

Moving on—so the last thing we looked at in the previous slide was what we were terming as legacy. So as you can see, we're broadly dividing the actuated signal controllers that are out there on the street in service into three broad categories that can affect your requirements. So in the previous slide, we saw the legacy ASCs that probably don't even have an Ethernet port. There's another classification that's called Modern, and that's the performance of most the ASCs that are in service today. If you look on the front of a controller you're taking out of the box or that you've purchased in the last five or six years, you'll see that there's an Ethernet RJ45 plug on the front; it's usually at least 10 megabits per second. The computer chip inside is greater than 4 million instructions per second, and some examples of those are ATC 5202 standard. That is a standard that's written very, very, very closely around the Caltrans 2070E controller. So you can use the 2070E's procurement specification. This is a standard that you can use if you're in another state and you want the same thing, and you want a standard. And there's NEMA TS-2 control ASCs that have been out there for a long time that have Ethernet ports on the front. That's probably the last five, six, seven or eight to ten years. Performance is typically greater than 4 MIPS and it has a single Ethernet. And then the last category here is ATC, or Advanced Transportation Controller. That is the ATC 5201 Standard. That standard is currently at revision 5.2b. It's available on the ITE website. This is the latest standard that was adopted about six years ago. It's been around—all the major manufacturers are building ATCs. It has four Ethernet ports on the front, so you can have a separate one going to your central and then another one for your roadside equipment. Performance is—standard says 60 MIPS. Typically the ones that are in production are around 400. And some examples are like if you go into the Caltrans, there's called a model 2070LX in the T specification. The manufacturers also have shelf-mount ATCs. And again, it's much higher performance and it has Ethernet, so the idea here—we'll talk about this more in later slides—but the idea here is we're just kind of planting the idea of—in the previous slide you may have legacy controllers that don't even have an Ethernet port and they're very slow. You're likely going to see a lot of the Modern controllers that have an Ethernet port, and you're beginning to see the ATC 5201 Standard controllers that are very fast and have multiple Ethernet ports. And we'll see more in later slides, when you start running into these different types of controllers how you write requirements is going to vary.

A315b, Understanding Requirements for Actuated Traffic Signal Controllers Based on NTCIP 1202 Standards, Part 2 of 2

And again, the legacy ASCs are constrained by communication performance. So they just cannot communicate fast enough if you don't have an Ethernet port on them, and we're going to see that we're going to have to have very compact messages because we can't put out very many bits very fast, so the messages have to be very short, very compact. The Modern ASCs are constrained by processor speed. They're going to have an Ethernet port that's at least 10 megabits, but the processor may not be able to handle all the data that's coming in at that speed. And then of course the latest, ATC 5201 Standard, ASCs include performance and they have an internal network switch needed to handle the long strings of standardized objects without overwhelming the processor, and we're going to see how that affects requirements in later slides.

So, we saw that the ASC is constrained by the network; the Moderns are constrained by the processor performance; and the 5201, the latest ATC standard 5201, which is currently 5.2b, they have the fewest constraints. So they have standardized objects and they have superior computational performance.

ASC types can dramatically affect performance requirements. For example, if we go into a conformance group and we have a requirement to get the status of a controller continuously—like we have a display in our central, we want the display to update the speeds in sort of real-time, so you want to be able to get the status over and over again—the legacy ASCs will be constrained by communications. It may only be able to transmit at very low bit rates. The Moderns are constrained by computational speed. So you can see that if the requirements for a dialog—say if you have the old legacies that have been out there more than 10 years or 15 years, they require—you're dealing with requirements that require very small dialogs of very short, compact messages that are complex, meaning that they have a lot of information in them, and those very compact and complex objects must be documented and tested for each ASC; and again, we'll talk about this more later. The Moderns can use long dialogs of standard objects but may be constrained by computational speed. They may be able to get dialogs in at 10 megabits per second, but cannot be able to handle that much and might not be able to handle that and control the intersection at the same time. So of course the intersection is going to run and the dialogs may not be able to be unpacked and you'll get some negative responses on your communication where you have to do retries. And then finally, the ATC 5201s can use long dialogs of standardized objects without being constrained either by communications or computational speed.

And again, be aware that traffic management systems often include a mixture of Modern, legacy, and ATCs, the most modern. Although each type may include identical

A315b, Understanding Requirements for Actuated Traffic Signal Controllers Based on NTCIP 1202 Standards, Part 2 of 2

parameters, the objects will differ by type when communicating. For example, again, if we set all available objects repeatedly—you want to go and set—all the ASCs might be constrained by very low bits, communication bandwidth. But if you're trying to do the identical set, you're trying to exactly the same things of all objects to a Modern, it won't be constrained by the communications, but it may be constrained by the speed required to unpack each message and process it and put it in the appropriate place. The ATC 5201 will not have any of these constraints. You can send long dialogs of objects and it will consume them, and it has the computational power to unpack the messages and put them where they should be.

So let's go into this in a little bit more detail now that we've kind of laid the groundwork here. So if you go into Version 4 of the NTCIP guide and go to Annex E, there's some excellent information there that you can use to calculate NTCIP bandwidth. So shown in this diagram, in the yellow part over here on the right side, the actual data elements are prefixed by an SNMP header to form an SNMP. So that's Simple Management Protocol. So we're taking our data and we're putting an SNMP in front of it, and as you can see, the Ethernet communications of modern actuated signal controllers require the addition of the Internet Protocol IP header added in front of that, and then, as you can see here in the little calculation we have off to the left, that the resulting message typically consists of 26 bytes of header plus 23 bytes of data, plus 1 byte of header, and 1 byte of data per element for STMP. So we're kind of doing a comparison here of—you can see that by the time we do an SNMP message, it has a lot of overhead on it and the information is kind of buried in the middle, and Simple Transportation Management Protocol compresses it. So we might use SNMP, the long messages, if we have a high-performance actuated signal controller out there. If it can't handle that, we'll have to use the shorter messages.

Now, you'll say then, "Well, why don't we always use the shorter messages?" The reason is the longer messages, as shown here, where we have a lot of the overhead, the reason for the overhead is that means the data always looks the same. All the objects look the same. The headers look the same. The data messages look the same. So if I'm purchasing signal controllers from Manufacturer A and Manufacturer B, they're going to look the same. They're going to use standardized objects: the headers look the same, the objects inside the information content look the same, they have identification that always looks the same. So it's very, very simple to implement. On the other hand, if you're using STMP—so we have older traffic signal controllers that really can't handle the bandwidth of the communications for the long standard messages—they can be compressed into STMP to make them shorter so that they can go over this lower speed communication medium with fewer bits, but keep in mind when that happens there's differences between manufacturers. So that's kind of what we want to get across here, is it sounds like and it

A315b, Understanding Requirements for Actuated Traffic Signal Controllers Based on NTCIP 1202 Standards, Part 2 of 2

is really true that those long dialogs of standardized messages have a lot of overhead in them, but the overhead makes them look exactly the same, whereas if you're trying to do shorter messages; or if in your contract your procurement terms are allowing shorter messages, you should also keep in mind that when you receive that, they're going to look different. So you need documentation and testing in addition to just simply receiving the controllers, taking them out of the box, and then trying to figure out why they didn't work with the central, because the shorter messages are different.

We're going to move on now to Control Parameter and Status Objects. So if you go into NTCIP 1202 and you look at the conformance groups, there's a table there, and the table shows that they're categorized as C, T, or S, depending on the operation you're doing. So before you attempt to control a signal controller out on the street from a central or attempting to display data at the central, before we do that we really need to know what the capabilities of the controller out on the street is, such as the number of phases. You wouldn't want to go out and say, "Give me Phase 12, please," and you're asking that of an eight-phase controller, because it doesn't have a Phase 12. So the central will typically first query the ASC parameters to determine the range that says, "How many phases do you have?" "Two to eight," something like that. And you'll go around and query that information from the ASCs so you know what's out there. Once these parameters are known, the central may then begin to send control objects that affect the operations that set the time of day. So control object is—it affects what the controller does—change the time of day, set the time of day, set the phase, set instructions. It's demanding the controller to do something that's going to change the operation. And then also, finally, the status objects—the central may also retrieve the status that indicates the ASC operations. So if I'm displaying something at the central, I may want to get the time of the day from the ASC, or I want to get the Phase 2 and see what Phase 2 is doing. So again, it's control, parameter, status. We start by parameter to see what's out there, then we can control the ASC and then we can read the status of that.

We're going to move on now to database transaction sets and the effects and the advantages. A database transaction set will alter the ASC operation, and in the previous slide we learned that a control object set also affects the ASC operation, the difference here being that a transaction set buffers multiple objects into the memory of the controller, and then once all of the objects within the transaction set are received and stored in the controller memory, all of the objects are executed within the controller at once, all at once. So transaction sets must be used whenever the execution of individual objects, if you did them one at a time, would have a detrimental effect on the ASC operation. So for example, say we wanted to have a requirement to change multiple phases at certain times of the day. That would have a detrimental effect if we were

A315b, Understanding Requirements for Actuated Traffic Signal Controllers Based on NTCIP 1202 Standards, Part 2 of 2

sending out set commands that were changing the phases one at a time by individual objects at slightly different times. So for example, if I wanted to change phases by time of day, the time of the day for the change has come, and I start doing them one at a time, and maybe the communication was garbled and I have to resend it, and you're running with the controller that's got part of one timing planned and then part of another. That would be classified as detrimental. So if we did that by a transaction set, all the phase changes would be buffered in the ASC memory. So all these would be coming from the central, down to the memory. The memory would be buffering them, so it's accumulating them all, checking them to make sure that they were there, that the data is correct, it wasn't affected by noise, and then once they're all there, they're all changed at exactly the same time of day. So that would get rid of the detrimental effect. And another example of a transaction set might be creating an entire database for an intersection. The controller would obviously act erratically if we were sending individual phases, if we're kind of running it and configuring it at the same time, while the DB Create would buffer the entire ASC database, and then once the old database is down, DB Create would then put that database in effect all at once. So if communications is interrupted or whatever, none of the changes will take effect until they're all there.

We're going to move on now to consistency checks. Consistency check is performed on data received by the actuated signal controller, so a consistency check compares the received data to what the controller expects as values for that data. It has to be—the values that are expected have to be the values received. An example might be if the central attempts to define a new phase within a ring that already has that phase number assigned. So we've already assigned Phase 2 in Ring 1 and it's already there and it's already up and running, and the central comes back and says, "I'm going to define the same one again." In that case, the controller would run a consistency check and it would determine an error stating that the consistency check failed, with an identification number of the error. So then you could look that up and you could say, "Oh, I tried to define it but it was already defined." So consistency check might be thought of as a "sanity check" of the received data so that it rejects commands that are out of bounds for the correct and safe operation of the controller.

A little more on consistency checks. NTCIP has a protocol that defines the structure of the data transactions. So the protocol and the standard define the structure of the data between the devices. NTCIP has rules to define what is required for interoperability among devices and among various manufacturers. For multiple devices to be interoperable, each side of the communications interface must have an identical interpretation of the order of the data received. For example, the data content may be identical but not interoperable because the receiving end stores the received data in a

A315b, Understanding Requirements for Actuated Traffic Signal Controllers Based on NTCIP 1202 Standards, Part 2 of 2

different order than the transmitting end, which of course would be not good. So they have to have the identical common interpretation of what the standard means at each end.

We're going to move on to block objects here. Block objects are predesigned sets of standardized objects. So we learned about standardized objects, but block objects, they provide an efficient method for uploading and downloading large amounts of data, so we have lots and lots of data we want to upload and download. For example, an entire ASC database could be uploaded with one block object. Otherwise, each object would be sent individually and each of the communications overhead of the header, the redundancy check, and other communications overhead. So you can do all that in one block object, which sounds like an advantage, which it is, but we'll see that there are some tradeoffs when you're using block objects. Block objects may be needed to meet the requirement to get the status of multiple inputs at exactly the same timestamp, and we talked about that earlier. If you used multiple GETs it would result in the status of each input at different times, but block objects have been historically used to maintain data that's small enough to fit within the bandwidth of legacy ASCs. So if you have a legacy ASC, block objects were used long ago for ASCs that didn't really have Ethernet, communicated very slowly, didn't have the computational speed to do things. So a block object would take lots of objects, push them down into one much smaller object that doesn't have the overhead that we saw earlier for the standardized objects, and you can fit that within the communications channel, and we'll talk a little bit more about how that affects requirements as we go along here.

So let's talk about a block object. An example might be high-resolution performance measures. So you might have a requirement that the controller must log all vehicle detection actuations; every single one of them has to be recorded, and every one of those actuations has to be correlated to a signal phase and also has to be correlated to the signal phase color. So say you have to record every one, every single car that went over every single detector, and what color the phase was in and what phase correlates to that detector, and say the requirement is you have to do that every second. So if you tried to do that with multiple GETs and you tried to push all that information up to the central, you probably wouldn't be able to do that very well. So what you could do is if you had a block object, you could record all those, put them into a block object, push them off as one record.

We're going to back up a little bit here, and again, I'm going to bring this same slide up that you can see. On the right, again, this is the content, the part that's shown in yellow

A315b, Understanding Requirements for Actuated Traffic Signal Controllers Based on NTCIP 1202 Standards, Part 2 of 2

there, and on the left you can see that an SNMP message size is 26 bytes of header and 23 bytes of data, so there's actually more overhead than there is actual data, typically. SNMP is an acronym for Simple Network Management Protocol—we brought that up a little bit sooner. We'll go into a little bit more detail here. It's widely used by computer networks, so if you have an Ethernet in your office or in your home, it's likely to use SNMP. It's the essential protocol used by NTCIP-compliant devices. So NTCIP uses SNMP, and it uses a client server communication model, where the central computer acts as the client and the field device acts as a server. So in this case, with an actuated signal controller, the controller acts as the server to the central station, as the client. So as applied to NTCIP, SNMP takes our data and appends IP and SNMP headers to the information, and then also sends along the information, where it says Information Content in the yellow there, as the standard object that you see in conformance groups. And although this carries a lot of overhead, this protocol is identical among field devices, meaning that the field devices don't have to decode or encode this information in any unique manner. So SNMP takes more bandwidth, but it's identical among manufacturers. If you do it that way and you're using modern controllers that can handle the long standardized objects and dialogs, that has a very high probability of being installed in the field without any problems—without any additional testing, without any additional definition of the requirement and misunderstanding among Vendor A and Vendor B in the central.

So we talked a little bit earlier about STMP, which is Simple Transportation Management Protocol. STMP is an extension of SNMP, whereby the central and field devices dynamically on the fly establish messages, usually when they're first connected to each other. So it's kind of you connect them together and they do an initial dialog to see what's out there. These are composite messages, by composite meaning they're composed of other information or the same objects and dialogs, but these composite messages are made up of sets of data elements. So instead of sending multiple objects that consist of the IP header, SNMP header, the data and the other communications overhead for each object, STMP composes the data into one object. The result is substantially shorter messages, as you can see in the yellow off to the right that they are the header and the composite data. The downside is each side of the communications interface must know how to interpret the data that's inside the message and how to interpret the messages that are received, because they'll be different among manufacturers. So the advantage is, yes, they're shorter, less communications overhead, they're special. Really need to know that when you're writing requirements. So STMP works the same way as SNMP, uses the same data elements, but STMP simply provides a more efficient bandwidth solution to the same communications problem. And if you were kind of following along here in these slides off to the right where there was yellow, you could see that the—if you

A315b, Understanding Requirements for Actuated Traffic Signal Controllers Based on NTCIP 1202 Standards, Part 2 of 2

were counting the bytes in the standardized dialogs or versus STMP you can see reductions is very dramatic. It reduces the communications traffic and overhead by a factor of 25. So for about every 25 bytes we have to send out over the communications link for standard objects and dialogs versus STMP, we would send 1. So that's a very dramatic reduction. But again, when you're writing requirements, keep in mind that the short messages are not the same between manufacturers.

STMP cons, complexity, maintenance and testing. So, STMP provides a significant reduction in communications bandwidth. STMP has a large number of disadvantages. To reduce overhead, the object identifiers are removed, meaning that each ASC must know how to interpret the messages and the objects. These composite messages don't have standard identifiers with them, so if you get the object, the controller can't look it up and say, "Oh, that's object ID number"—whichever object it is. So they just come as composite objects, and that identification has to be done in a special fashion. So you can see that STMP increases the software complexity as the STMP objects are dynamically formed at startup, and software must be added to form and interpret each object in addition to the software required for NTCIP mandatory and optional objects, which we'll get to later.

Another disadvantage stems from that the data content is not defined by the standard. So if you are looking for the data content of an STMP message inside the NTCIP standard, you will not find it. Again, each end of the communication link must know what the data structure of that composite message is from each manufacturer. So even though, again, they're short, they differ between manufacturers, which can lead to extra requirements and extra work you need to do when you're writing your requirements.

If you'd like to refer to the student supplement, it's interesting that there's a graph in there that kind of shows—and we won't go into detail here because we really don't have the time—but you can see that STMP was done kind of back in the beginning when the controllers were slower and the bandwidth was slower. So those short messages were kind of required back then, or really needed, but now that the Modern and the ATCs, the ones that are being built now—if you go in the student supplement you'll see that there's a graph that shows how many signalized intersections there are out there and how many are being replaced, but Modern ASCs and ATCs have the communications performance to use SNMP with identifiers that have a common standard interpretation among manufacturers. So if you go in there and look at that, there's a graph that shows there's a lot of those old ones out there but they are being replaced. The ones that need these special short objects are naturally being replaced and the reason for that is

A315b, Understanding Requirements for Actuated Traffic Signal Controllers Based on NTCIP 1202 Standards, Part 2 of 2

manufacturers are not building those old controllers anymore. You take a controller out of the box and many of the major manufacturers, you will see that they have one or more Ethernet ports on the front, and they don't really have the need anymore, the requirement, to have these short objects. And then you might say, "Well, you know, that's true. I can replace the controllers but I've still got these old phones lines in the street." But you'll also see that technology, such as Digital Subscriber Line, or DSL, which it's commonly known, you can take your Ethernet port and go through a DSL line, or a DSL modem, and still use your old legacy communication copper pairs in the street, and you can still get data through those old copper lines using modern modems that really eliminate the need for STMP in a lot of cases. And the reason I'm bringing this up is you will be confronted with old controllers and there's kind of a choice there when you're writing your requirements: Do I want to actually keep the controllers on the street to save some money, and then invest some more time and money making these shorter messages that aren't in the standard and have to be documented and have to be tested in a special fashion, or I look at the cost-savings there doing anything special and just replace the controllers? So when you get your competitive bids resulting in a mixture of ASCs that may or may not maybe aware of the composite messages, you're going to have to deal with that. And the reason we're bringing this up is when you write requirements, you really need to take that cost tradeoff into consideration. The short messages will allow you to keep a lot of the older equipment. The cost of replacing it to newer equipment allows you to use standardized dialogs, messages, and identifications that are the same, so it eliminates a lot of the special testing, special documentation, and special things you have to do when you deploy.

And again, if you go into the student supplement, there's a couple more things we didn't really have time to put in here. The student supplement includes a much deeper dive into topics that can affect ASC requirements when you're writing your requirements or you're writing your procurement contract. For example, the supplement will include some data from studies that predict the number of ASCs that will be in service by performance type. And again, you might want to look at that, because it shows—the only ones you're putting on the street now can handle long dialogs and standardized objects, and really take a look at that graph and just try to predict when your equipment is going to go into service. You might want to really take a hard look at, in your planning phase, of whether you want to even deal with the nonstandard objects or not, based on the cost of using newer equipment. The student supplement also includes information to calculate communications loading. I know you're probably saying, "You know, that's true, but I don't really know what the communication loading is." There's a lot of good information in the supplement that shows you how to go in and say, "Well, I got this many ASCs out there. I have to talk to them at so many times per minute," or whatever. You can get down to

A315b, Understanding Requirements for Actuated Traffic Signal Controllers Based on NTCIP 1202 Standards, Part 2 of 2

calculate the communications loading. You can look at your ASC data sheet on whoever's controller you're buying, and you can get a good idea of whether you really need to go to the shorter messages or not.

There's some other topics that we would really suggest you go into the student supplement. It covers requirements for ASC clock coordination. You might wonder why that affects requirements, but ASCs typically get the time of day from service frequency. We're on a 60-cycle-per-second, 60-hertz line. There's a circuit on the controllers that counts the cycle, so every time 60 cycles go by, it updates the clock by one second. That's very typical. That works very well. Now we're starting to see things like, "Well, you know, we're starting to do some 48-volt DC units. There's portable intersections where there's no service frequency," so you have to have—you're using the crystal oscillator, which is very, very accurate short-term but can drift over long terms. If you're trying to do coordination that's not going to work very well, if you're doing like greenway progression. And then a lot of controllers have GPS receivers, Global Positioning System. We're not really trying to see where the controller is, but the GPS signal comes with a very precise time of day, down to the—a very precise time—and another way of doing it is if you're on a network where you have a central has the time of day, the central may just broadcast time out to all the ASCs from central to keep them in coordination. But we're not going to go into any more detail, but you really need to grasp the concepts there when you're writing requirements. If you're on service frequency, fine; if you're not, if you're trying do very precise calculation, you're on other power other than 60-hertz lines, there's some things you need to know that affect your requirements that's worth reading about in the supplement.

Okay, we're going to do an activity now. So this is kind of a test of your knowledge and understanding of dialogs. So let's ask the question: Which of the following is not—and again, not—part of managing requirements correctly? And please be aware that we're doing a negative here. So, the answer choices are: Which of the following are not part of managing requirements correctly? A is associating each requirement to the correct conformance group; B is correct implementation of objects within the conformance group; or C, correct dialog to meet requirements; and finally, D, adding a requirement received via an email. So of these four, which of the following is not part of managing requirements correctly? We're going to move on to the answers. The correct one is "adding a new requirement received by email." Now, again, we went back to the concept of a gatekeeper. In the prior modules leading up to this one, we learned how to write requirements, but when you finally get down to writing requirements, you're doing a real-world project. There's almost certainly going to be, as the project goes along on an ITS project that goes months or years—on large projects the requirements may change and

A315b, Understanding Requirements for Actuated Traffic Signal Controllers Based on NTCIP 1202 Standards, Part 2 of 2

you need to manage them correctly and part of managing them is having a person that makes sure that that requirement came from an authorized source. It's well stated, it includes an incremental estimate of the cost and schedule impact before it's reviewed and approved by the stakeholders. A is not correct, each requirement is fulfilled by a conformance group. And again, we did have the word "not" in here, so these are going to be pretty much the same. Content of each object must be correct—that is part of managing requirements correctly. And finally, C, objects must transact in the correct sequence. So to manage your requirements correctly, you have to have A, B, and C, and the outlier there is D.

We're going to move on to another activity. Again, we're going to have a question here. How do ASC types affect requirements? So we're talking about controller on the street and the type of controller. And again, it was legacy—really old—Modern, that has Ethernet communications, and ATC, the latest, that has high-speed communication as well as very fast computational power. So again, the answer choices here is: Different objects and dialogs to meet identical requirements; legacy ASC requires the use of compact, complex objects; C ATC 5201 ASCs handle long dialogs of standardized objects; D Legacy ASCs are being replaced within existing budgets; and the ASCs that are being replaced can handle standardized objects. And the last choice there, the final choice, is All of the Above. Reviewing the answers. The answer is E, all of the above are true. A legacy ASCs are constrained to short dialogs. We know that to be true. The legacy ones don't have the computational bandwidth or communications speed to handle anything but short dialogs. B is also correct. Legacy ASCs are constrained to low speeds. C, ASC 5201s—yeah, each object can be standardized. It comes with an ID and they have the computational speed to unpack the long objects, long dialogs, all the overhead, and look at the ID and look it up in a table and know what's going on, and they're all the same within vendor-to-vendor. And then again, D, yeah, suppliers are building only Modern ATC 5201 and ATC 5202 controllers, so that's really all you can buy now. Probably—and again, we just bring this up—if you want to retain your old controllers, that's perfectly fine, but just be aware that there's a cost associated with that with the very short messages differ among manufacturers.

We're going to sum up here what we learned in this learning objective number 5. We learned that requirements are managed on behalf of the stakeholders, meaning to associate each requirement with the correct conformance group, implement the object correctly, design correct dialogs, and force the quality of the sources of the requirements through the assignment of a gatekeeper. That's what we learned for how to manage requirements, what does it mean and how to do it. We learned that ASC types may have differing requirements to perform the same functions. Types include ATC, Modern and

A315b, Understanding Requirements for Actuated Traffic Signal Controllers Based on NTCIP 1202 Standards, Part 2 of 2

legacy, with significantly different speeds and communications abilities. ATCs can handle large dialogs of standardized objects and it eliminates the differences among manufacturers. Therefore, the cost of documenting and testing the differences should always be compared to the cost of controller replacement in order to use standardized objects as part of writing your requirements. Control objects alter ASC operations; status objects indicate controller operation; and parameter objects indicate capabilities, such as the number of phases. We learned that database transaction sets consist of multiple control objects that are received and executed at one time to avoid the detrimental effect of if we tried to alter one parameter at a time. Consistency checks and rules provide a sanity check to ensure the data is within the expected range and is in a commonly interpreted format by each device on the channel. Block objects are predetermined sets of standardized objects. We learned that SNMP is a communications protocol widely used by computer networks, and we also learned that STMP is an extension of SNMP designed for the transportation sector, whereby the central and the field device dynamically establish messages, usually when they're connected, the messages can vary among manufacturers. So that's kind of a roundup of what we learned in learning objective 5.

So now we're going to move on to learning objective number 6, and in this learning objective we will learn how to fulfill requirements that are not supported by standardized objects. And again, we talked a little bit briefly, in an earlier slide, that if you go into NTCIP 1202, you go into the conformance group, standardized objects, if your controllers are fast enough to use standardized objects, that's great. You can, if you're trying to preserve legacy controllers and need the short ones, you can use block objects. But actually, if you look in the standard, NTCIP 1202 allows the use of what is called Manufacturer-Specific Objects. So what is that? What is an MSO? They're allowed, and described in the standard, but let's recall that from the first part of this module NTCIP 1202 supports the functions of the NEMA TS-2 Standard. So if you look at the NEMA TS-2 Standard for a controller, the NTCIP objects support those functions. So, as you can well imagine, controllers often do things beyond what's just the basic requirements of meeting the TS-2 standard. So manufacturer-specific objects are allowed, and the best way and the proper way to use them is to support advanced functions beyond TS-2 standards.

So, let's talk about the proper use of the manufacturer-specific objects. First of all, let's do not confuse MSOs with optional objects. Optional objects are defined by the NTCIP standard. Optional objects are not required to be delivered by the manufacturer of the controller, but if the manufacturer of that controller is implementing that function, it must be done the way the optional object describes it. So you may not have to put that object

A315b, Understanding Requirements for Actuated Traffic Signal Controllers Based on NTCIP 1202 Standards, Part 2 of 2

in if you don't need it, but if you do want to implement that function, you have to do it the standardized way that's described in the NTCIP 1202 Standard, and the reason that's done is for interoperability among manufacturers. Not required to put it in if you don't need it, but if you do use that function, it has to be done and implemented exactly like the optional object definition in the standard. On the other hand, MSOs are not defined by the standards. You will not find manufacturer-specific objects in any of the standards. They're defined by the manufacturers, not by the standard. So Manufacturer A can define a manufacturer-specific object for their controller, another manufacturer can define an MSO for their controllers, and since each manufacturer is free to define their own specific objects, each MSO is non-interoperable among manufacturers. So the power of an MSO is you can implement a new function that's beyond the TS-2 standard to meet a requirement that's based on a user need, but they're not interoperable among manufacturers. If you get a mixture of controllers from various manufacturers, that's a very important point to know while you are writing your requirements.

So let's talk about one proper use. We're going to go into proper uses and improper uses. Let's start with the proper uses. So one proper use of MSO allows innovation and new features beyond TS-2. So MSOs could provide a migration path to become standardized optional objects if they're widely accepted by users and widely used, and it's important that specification details are available and documented in an open format so it can be used by the other users, and one example of this—and I think we have this in a little bit more detail—we're not going to go into exactly what this is—but in the supplement, there's a concept called Indiana Performance Measures that was—the developmental work was done by Purdue University and Indiana DOT. But it's basically how to see how well your controller is performing, how your patterns are performing. Again, we're not going to go into detail on what it does, but it was something that was very cool, so you can measure performance on Vendor A, Vendor B, Vendor C using a common yardstick. So it's one standard way to measure multiple manufacturers, so it makes a level playing field for measurements. So that one started out as a private NTCIP node, and it worked so well that it was adopted by at least three manufacturers, maybe more, for interoperability. So that's the proper use of an MSO. You start with a user need and a requirement for a performance measure, then we went further to make it a manufacturer-specific object, and then it moved on to become documented so that it can be interoperable among three ASC manufacturers. So again, this was done—so the three manufacturers are doing it. They adopted the same way, and it was done that way because it was designed and documented correctly so that the manufacturers could use it.

A315b, Understanding Requirements for Actuated Traffic Signal Controllers Based on NTCIP 1202 Standards, Part 2 of 2

So let's talk about the improper use of manufacturer-specific objects. Rule one, MSOs are never to be substituted for standardized NTCIP mandatory and optional objects that perform the same function. MSOs in the past have been used to disguise a manufacturer's proprietary protocol. For example, each legacy command from an older protocol could be wrapped in an NTCIP header as a manufacturer-specific object, and that manufacturer may say, "I looked at your procurement specification and it said I must be NTCIP compliant. NTCIP allows the manufacturer-specific objects. I took all of the proprietary data flow that I have and I put it in MSOs that only I understand and I conform to the standard." Well, maybe technically, but that's not in the spirit of the standard, because the spirit of the standard is for interoperability, so this would not be allowed. So again—and we'll get into how you can enforce this later on—but MSOs are not to be substituted for mandatory and optional objects. If there's a way to do it using mandatory and optional objects, you should require that they be used and not MSOs. MSOs should be barred from being used in that manner. So, clearly improper uses of MSOs include replacement of mandatory objects, replacement of optional objects, and used as a wrapper for proprietary. So we don't want that. Those are improper. So we talked about a good way to do it is if you're doing a performance measurement, TS-2 standard doesn't have that, you do a specific object for that, a new object for that; you document it correctly so that the person, the agency that's buying the equipment, can reuse it on later procurements. That's proper. Just using it as a replacement for the mandatory and optional breaks the interoperability spirit of NTCIP, and should not be done.

So, let's go through a real quick real-world example. There's more of this in the student supplement. But let's just say we're—say you had a need to—you had a sports stadium that you had a corridor where there was 30 thousand cars a day going up and down in a normal day, but there's a sports stadium, we're in a dense center, and then suddenly there's hundreds of thousands of people coming out, getting their cars. So that's a need to keep the traffic moving. The requirement would be to change the traffic signal control timing pattern based on approaching vehicles and not based on a fixed timing plan. So you could do that as a manufacturer-specific object. So you could have a requirement that says ASC is required to detect vehicles departing each intersection heading toward the next one, so it sends that information to the central once per second. A requirement is that that information has to be sent in time to calculate a new signal timing before the vehicle arrives. Another requirement might be you have to do that every three seconds. You look at NTCIP, you look at the conformance group, you look at the TS-2 standard. That's not in the standard, so based on that analysis, we say, yes, that's a candidate for a manufacturer-specific object. So that's just a quick way that you might run across from user need, to requirement, to MSO in the correct manner. So again, if we were doing that, that would be an example of adaptive control. In the previous slide, we saw the need

A315b, Understanding Requirements for Actuated Traffic Signal Controllers Based on NTCIP 1202 Standards, Part 2 of 2

and the requirements. Moving on to the actual design—so we might make a manufacturer-specific objects that calls stages to phases that would lump every combination of non-conflicting phases into a stage with a number, and then we'd create a stage—next MSO—to set the signal controller to the stage that best handles the approaching traffic. So the sports stadium lets out; a whole bunch of cars are coming. We know the stage which is a conglomeration—or is a combination of a set of non-conflicting phases that would best handle the traffic coming from that direction. Sends that object, tells the controller to go to that stage, turns on the signals into green so that it handles that lump of traffic, then goes on to another stage. So that would be a good application of an MSO, just as an example.

We touched on this a little bit and again, the intent of the MSOs for advanced functions that aren't handled by the standard. We know that in the past it hasn't been done for that, but in your contract wording we would suggest strongly that you enforce the intent. The binding contract should include provisions stating that all mandatory objects shall be provided. If it's mandatory, it shall be provided. You have to provide the mandatory ones to be compliant. Contracts should further require that all optional objects required for the function shall be documented for interoperability so that I know which optionals are done in which controller. The contract wording should ban the use of MSOs except for those specified in the contract for functions that are not contained in the NTCIP standards. So we're suggesting your contract wording should say, "No MSOs", or "No MSOs except this one that we've identified that's not in the NTCIP standards." We want this one because it's something that's beyond what the standards can do."

We're going to move on now to exception-based reporting. What is exception-based reporting? Exception-based reporting is instead of polling each controller with GETs to determine the changes in status, exception-based reporting allows the controllers to report their events, such as changes in status, without continually reporting status fields that have not changed. So exception-based reporting ensures that critical events from the controllers are not missed. If there's lots of controllers and lots of dialog, and so much transactions are going on that the communications channel may not be able to handle the volume, you don't want to miss an event. So if the required update rate and required bits per second exceeds the communication channel bandwidth requirements, exception-based reporting may be used to meet the requirements.

So, let's look at a real-world case. Say you have 10 thousand—and there are real-world cases where there are 10 thousand controllers connected to a central—all of them must be updated in a short amount of time, must receive updates. We don't want to miss

A315b, Understanding Requirements for Actuated Traffic Signal Controllers Based on NTCIP 1202 Standards, Part 2 of 2

anything that's going out on those 10 thousand controllers, but the communications bandwidth into my traffic management center is nowhere near enough to be able to go out and pull every one of those 10 thousand controllers individually for all of their data all the time, and then try to figure out what's changed. So instead of bringing all the data up from everyone and then going through and saying, "What's changed? What's different?" the ASC will be able to push a change up to the central. So that's exception-based reporting. So when we're doing the design, we would determine that the available bandwidth is less than the communications load of get statuses from all the ASCs. So to meet the requirement of updating—getting status from all the controllers without missing any—that's a hard requirement—each ASC can report exceptions when the data has changed. So instead of sending up all the data all the time that hasn't been changed based on polling, it says here something that's different you might want to know about and push it up as an exception to normal operation.

We're going to move on onto verifying conformance to the NTCIP standards. Conformance exists when an ASC meets all of the mandatory requirements as defined by the standard—all of the mandatory ones. You must supply the mandatory ones if you're a manufacturer. NTCIP conformance should be clearly defined in procurement contract wording, and what we're trying to do here—and we'll get into that a little bit later—but we're trying to avoid enforcing conformance. We are trying to design conformance in as we go along. So it should be clearly defined in the procurement contract wording what is—we are going to say, "We are going to accept your controllers and we're going to accept the system and we're going to pay for it based on this wording that's in the contract. This is what you need to do as a manufacturer, as a service provider, as a system engineer, to be conformant." So we're suggesting here the contract, the terms and conditions, should include the actual list of mandatory and optional objects necessary for operation, put them right in the contract, state that, "You shall provide the mandatory ones and you shall list the optional ones that you need to make your equipment operate, put them in the list. You must supply those with your bid." The contract terms, we're suggesting here that manufacturer-specific objects should be forbidden. "MSOs are not allowed except for new functions not defined by NTCIP." And the same terms should include acceptance terms. The terms and conditions should usually say acceptance terms, but let's go take it a step further here and say, "Acceptance includes these test scripts that runs on this off-the-shelf test equipment to verify interface operation before the equipment is installed." So we're suggesting that your contract terms at bid time should say, "In addition to the mandatory and optional objects and the proper and improper use of manufacturer-specific objects, when you deliver the system and each piece of the system, we're going to use this particular make and model of off-the-shelf tester that runs on a personal computer or whatever, and we're

A315b, Understanding Requirements for Actuated Traffic Signal Controllers Based on NTCIP 1202 Standards, Part 2 of 2

going to run these scripts on it.” That way I know how to bid—if I’m a manufacturer I know how to bid it, I know what you really want as requirements. It’s a level playing field for all bidders. I can buy the same equipment, and as I’m doing my design, I can do verification as I go along so there’s no surprises at the end where verification is going to have to be enforced through penalties, etcetera.

And again, test scripts are required for off-the-shelf test equipment to verify the operation before it’s installed, and that way, again, each manufacturer is able to test using the same independent test equipment, using the same acceptance test scripts. So if my actuated signal controller connects to someone else’s central, someone else that’s designing the central has the same equipment and the same test scripts that I have, I can test against the third-party equipment. The central manufacturer can test against the central equipment when they’re connected together they work.

Again, conformance—again, we’re trying to verify, not enforce. The conformance to NTCIP is verified by using the Requirements Traceability Matrix, and we’ve got a little example here, if you look at the green bar across there. In the green bar that you see, each functional requirement as a requirement ID and a dialog ID. So those are identifying the requirement of the dialog. The dialog consists of several objects and each with an object ID and an object name. Conformance to NTCIP is verified by the prescribed test scripts. So we’ve already identified the scripts in advance. An example might be you have—an off-the-shelf NTCIP test equipment can be used to select the applicable objects and to design the dialogs of the objects to fulfill the requirements. And we’re going to get later on into testings. We’re going to cover that more later on in how we actually do testing. But again, for requirements, the requirements should be listed as far as verification and conformance. So ideally, again, the procurement contract will specify the make and model of the test equipment and a list of the test scripts that make up the dialog. So again, each manufacturer uses the same equipment, same interface dialogs. Each manufacturer can verify their design against equipment, simulate the responses that would come back from another manufacturer’s equipment if they were connected together. So this allows every module to be tested thoroughly, up to the interface stage, and then once you do integration and field test, it will go smoothly.

And again, the student supplements includes a deeper dive into the topics that affect ASC requirements. For example, the student supplement lists adaptive systems that are currently in the United States that are shown on the USDOT website. In addition, ASC requirements that can affect adaptive controllers are discussed in the student supplement, are we include vehicle detection requirements and roadway technologies

A315b, Understanding Requirements for Actuated Traffic Signal Controllers Based on NTCIP 1202 Standards, Part 2 of 2

such as adaptive control of arterial corridors versus central business district, grid topologies. You'll see that adaptive controllers—a number of different adaptive controls—and a lot of them, if you're writing your requirements, you will see from the supplement that some of them are better suited for very tight grids with lots and lots of intersections versus arterial corridors, and that's discussed in the supplement. The supplement also includes a hypothetical example of designing and documenting an MSO for adaptive. It's purely theoretical, just to show you how you would go through that, because NTCIP and NEMA TS-2 do not include mandatory or optional objects for adaptive. We're kind of including that in here because adaptive is very powerful and can change timing plans on the fly and adapt in real-time. It's not really like having quick response, where you have a plan, something happens, you have an incident, and you go to a different plan. This is actually keeping up with the traffic that's approaching on a second-per-second basis. We put that in the supplement for further reading.

We're going to have a final activity here. So the question here in this activity is: Why should dialogs be defined in a procurement specification? Why would you want to do that? You're writing your requirements, you're writing your procurement specification, why do you want to go so far as to do the dialogs? That's going to be a lot of work up front, it's going to—you're going to have to really think through a lot of the design, high-level design, before you can do that. So why would you want to do that? So the answer choices are: a) devices are more likely to conform to the standards; b) devices are more likely to interoperate and interchange; c) you'd want to do that it increase the total cost of ownership; or you want to do that because it'll likely make your devices less expensive. So those were our choices—A, B, C, D. Moving on to the answers, the correct answer is B. We would do that because dialogs promote a common expectation of how objects are to be exchanged among devices. So you would do that, even though it's going to take more time up front, more time to write your contract, a longer contract—you're going to have to probably go into the high-level design a little bit to get the right selection of the dialogs—but in the end, when you get to field test, you plug it together and it works; and again, it's trying to avoid enforcement after the equipment is installed and it doesn't work. It's trying to verify it as you go along, and beginning with writing your requirements, you try to lay the groundwork for verification as it's going through the design phase, system integration, system test, field test, turning it over to the owner-operator. So you try to do that in the beginning so that can be carried all the way through. A, not correct. Dialogs are not part of the standard. It's not that devices are more likely to conform to the standard because the dialogs aren't even included in the standard. They're not part of the standard. The standard has the object and the conformance groups and it has a definition of the objects, but it doesn't show what order they come in and what you use them to do. So dialogs are part of the design used to meet the requirements, not part of the standard.

A315b, Understanding Requirements for Actuated Traffic Signal Controllers Based on NTCIP 1202 Standards, Part 2 of 2

C is incorrect. Interoperability results in lower integration and testing costs.

Interchangeably allows competitive bids of standard equipment throughout the service life of the system. So it's not going to increase the total cost of ownership, even though for sure or it's very likely that it's going to cost you more to write your contract, it's going to cost you more to do a little bit more of the high-level design and look and see what the interfaces are and see what the scripts and applicable test equipment looks like—yes, that is a cost, but the total cost of ownership over the life of putting it into the system and installing it and running it—and even later on, when you're trying to add equipment to it—maybe five, six, seven, eight nine years you have to expand the system. That's very common and it's good to have all those scripts and that test equipment there so when you go out for another bid it's all laying there and you can just pick it up and run with it again. And again, D is not correct. Specifying longer dialogs of standardized objects might require upgrading or relocating legacy controllers. So it may increase your device cost because you may have this old controller that's been out there 15 years. It may cost you something to replace it, but the advantage of that is it runs standard objects; the objects from the new controller are going to look like the objects from everybody else's controller, forever.

Let's go quickly through what we learned in learning objective number 6. So, in this learning objective we learned that manufacturer-specific objects are intended to be used to implement controller functions that are not defined by the NEMA TS-2 Standards. We also learned that manufacturer-specific objects should never be used to replace standardized objects that implement standard functions defined by the NEMA TS-2 Standard. MSOs are defined by manufacturers, not by the standards. So we learned that using an MSO to replace a standard object will create conflicts in the standards—you're not meeting the standards, you are breaking the standard, and it breaks interoperability among vendors that use standardized objects, because the other equipment on that communication link aren't going to know what those are, and if there's no documentation, they're not ever going to know what they are and how they work. We also learned that adaptive control is an example of the proper use of an MSO to automatically adjust signal timing in advance of arriving vehicles in real-time. That's because adaptive signal timing to traffic conditions is not a function defined in the TS-2 standard. It's offered by several manufacturers. It's becoming very popular, and it would be a good candidate for an MSO. And here we're suggesting that the procurement contract wording is included that prevents the use of MSOs to implement standard functions. So that wording prevents manufacturers from claiming compliance to the NTCIP standards through the use of manufacturer-specific objects without being interoperable.

A315b, Understanding Requirements for Actuated Traffic Signal Controllers Based on NTCIP 1202 Standards, Part 2 of 2

Moving on with the summary of what we learned in 6, we learned that adaptive control algorithms automatically adapt signal timing ahead of approaching vehicles instead of the vehicle having arrived, like actuated. By the time the vehicle actuates the intersection, it's maybe too late to clear the traffic; or, based on historical data, like you would do, like a quick response plan or something like that. It's really not based on any history of data; adaptive control is based on the vehicles that are coming your way and is controlled second per second. Adaptive algorithms provide immediate and automatic response to incidents and abnormalities such as collisions and sporting events that would alter dramatically a normal traffic flow. We also learned that exception-based reporting is most useful in large ASC deployments that might otherwise overwhelm the available bandwidth. And again, our example is instead of continually pulling thousands of ASCs using multiple dialogs to determine the need for service from central, the ASC itself can issue an exception only when service is required. So it decides when something is different and needs service, so it issues an exception. We talked about the fact that verification should always be part of the procurement contract and not an afterthought at the end of the contract and the period of performance. And our example is the interfaces between manufacturers should be identified with the contracts. So we should identify the assemblies we're going to be buying, the controllers, the central. We need to know what the interfaces are between them, whether they're Ethernet IP that's very fast, or an old modem that's very slow, and identify all that in the contract for sure. And then we also learned that we desire the test equipment, objects, and script—the dialogs to also be included in the contract, even though it's going to be some extra work up front. And if we do that, each manufacturer can verify the design against a common set of equipment and scripts while they're doing the design, so verification is built into and through the design phase, and that will avoid surprises when the modules are integrated and the system test is conducted later on.

Let's go through the fill-in-the-blank here on what we've learned. Okay, number one: The same functionality may lead to different requirements for ASC performance types. The ASCs may have the same requirements but desire the same function from each one—but depending on how old they are, what their performance are, you will write the requirements differently, even though you're trying to meet the same user need and the same function. Number two: Depending upon requirements, parameter control and status can be realized as individual objects, block objects, or transaction sets. So they can be realized three different ways. STMP should be avoided to use standardized objects when communicating on modern networks. If you have a modern communication network, you should be using standardized objects. All objects are exactly the same among manufacturers with the same identification. So later on, when you're expanding the

A315b, Understanding Requirements for Actuated Traffic Signal Controllers Based on NTCIP 1202 Standards, Part 2 of 2

network, it'll be standard forever and you don't have to go back through trying to figure out nonstandard objects.

ASC clocks coordination—we touched on this briefly and, again, there's more information on that in the supplement—but clock coordination is constrained by service power, communications, and coordination type. So if you have service power, you're probably going to run your clock off of line frequency. If you don't have line frequency, you'll probably run it off of crystal, but if you're trying to do crystal—you're trying to do time-based coordination, you're trying to do a greenway, you have crystals on multiple controllers, and you have a 45-mile-an-hour greenway and the crystal is gripped a little bit, your greenway is not going to work and you're going to have to rely on communications or GPS to keep them in sync. So again, this is just—more detail in the supplement—but you should be aware of it when you're writing requirements. Always go through your requirements and include clock coordination as part of your requirement considerations.

Moving on: Manufacturer-specific objects are properly used for functions not defined by optional and mandatory objects. If there's optional and mandatory objects, you use those. Except for their proper use, manufacturer-specific objects should be banned by the contract terms and conditions. This will keep a manufacturer from saying, "I conformed to the NTCIP 1202 because it allows manufacturer-specific objects. All of mine are specific." So you ban the ones that are improper. Adaptive control is an example of a manufacturer-specific object used to meet requirements not possible by using standard objects. So there is a place for MSOs, and that would be to do something beyond what the standard NEMA TS-2 defines.

Exception-based reporting can be used to receive status changes without repeatedly polling all of the ASCs on the communications network. So if you're going to have a lot more data than your comm link would allow and it would be overwhelmed if you tried to go around and listen to every single ASC all the time and your math when you do your communications calculation shows that you're going to miss some polls, you can use exception-based reporting so that it's not sending the parts that haven't been changed; it only sends the exceptions.

Number nine: Contract terms should include Requirements Traceability Matrix and test scripts for third-party test equipment, and we covered that. Again, that relates to—put that in your contract up front so the manufacturers know exactly what they're bidding. They

A315b, Understanding Requirements for Actuated Traffic Signal Controllers Based on NTCIP 1202 Standards, Part 2 of 2

bid it the same way, and they have the test scripts and the equipment during the design phase to verify conformance all the way through the design process, so if you put it in the street you don't have to try to enforce it if it doesn't work.

Finally, this is a list of our resources. You can go through those. One of those is the AASHTO Deployment Analysis. That's where it goes through and explains a lot of the ASCs that are out there and the different types. You can go through and look at that one. That's very good, and there's some more information here on the standards and where to find them.

And that ends our module. Thank you for participating. We appreciate it.

End of A315b, Understanding Requirements for Actuated Traffic Signal Controllers Based on NTCIP 1202 Standards, Part 2 of 2