Module 31
A315a: Understanding User Needs for Actuated Traffic Signal (ASC) Controllers Based on NTCIP 1202 Standard v03

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, director of the ITS Joint Program Office for USDOT and I want to welcome you to our newly redesigned ITS standards training program of which this module is a part. We are pleased to be working with our partner, the Institute of Transportation Engineers, to deliver this new approach to training that combines web-based modules with instructor interaction to bring the latest in ITS learning to busy professionals like yourself. This combined approach allows interested professionals to schedule training at your convenience, without the need to travel. After you complete this training, we hope that you will tell colleagues and customers about the latest ITS standards and encourage them to take advantage of the archived version of the 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 which improves livability for us all. You can find information on additional modules and training programs on our website www.pcb.its.dot.gov. Please help us make even more improvements to our training modules through the evaluation process. We look forward to hearing your comments. Thank you again for participating and we hope you find this module helpful.

Patrick Chan: Hi. This is Module A315a, Understanding User Needs of Actuated Traffic Signal Controllers (ASC) Based on NTCIP 1202 v03 Standard. This is an actual update of a PCB module first presented in December 2013 because a new version of NTCIP 1202 v03 was released in May 2019. My name is Patrick Chan, I'll be your instructor for this module. I've been involved with the development of ITS standards since the year 2000. I am the consultant, editor, systems engineer, applying systems engineering process to numerous NTCIP standards including this one, NTCIP 1202 v03, which is the topic of this module. I am a voting member of numerous ITS standards development committees, including the NTCIP Joint Committee, and several SAE technical committees involved with the development of connected vehicle standards and recommended practices. I have over 29 years of ITS experience, including four years as an ITS project manager with a public agency.

Our learning objectives for today are, one, to review the structure of NTCIP 1202 v03. We'll talk about what it is, what it's based on, what it contains, and how it's organized. We'll then review the specific operational needs that NTCIP 1202 v03 supports. What transportation problems does this standard address? Then we'll introduce a matrix or a table which is found in NTCIP 1202 called the Protocol Requirements List. We'll describe what it is, how it helps agencies and procurers purchase and vendors provide systems or products that conform to the NTCIP 1202 standard. Then we'll end this module with how an agency can create a project-level protocol requirements list to procure a system using NTCIP 1202 tailored to an agency's specific needs.

Our first learning objective is to review the structure of NTCIP 1202. Over the next several slides, we'll introduce the NTCIP family of standards, describe the relationship with NEMA TS2, a specification, and NTCIP 1202, and introduce the structure of NTCIP 1202 v03 standard, particularly the systems engineering content. This first slide just simply defines what a traffic

signal controller is. This is taken from the NEMA TS2 specification and we're focusing on the controller unit, and the controller unit is that portion of a controller assembly devoted to the selection, timing signal displays, and signal indications at a signalized intersection.

This diagram shows the different NTCIP standards that form a communications interface. NTCIP is actually a family of standards and consists of several standards that together define the rules and data dictionary on how a traffic management center exchanges information with other centers and also manages traffic field devices such as a traffic signal controller, a dynamic message sign, a ramp meter, the loop detectors, weather sensors, and other traffic field devices. This diagram presents the NTCIP communication stack and depicts how the different NTCIP standards are organized and together comprise the NTCIP suite of standards. NTCIP 1202, the focus of this module, is an information-level standard, or data dictionary, that defines data objects when managing a traffic signal controller, but relies on other NTCIP standards to fully define how data exchange between the traffic management center and the traffic controller. An analogy might be the English language. The English language has a dictionary that defines the words, but there's also English grammar to describe how those words are used and communicated, how those words are used to create sentences, and how those sentences are used to exchange information. For NTCIP, NTCIP 1202 defines the data dictionary for traffic signal controllers while there are other NTCIP standards that define the rules on how these words or data elements are communicated and exchanged.

So what is NTCIP 1202? Well, NTCIP 1202 is an information-level standard or data dictionary that defines the data exchange between the traffic signal controller and its control systems. It also references NTCIP 1201, which is a separate data dictionary which defines data objects generic to different traffic field devices including dynamic message signs, detective systems, and ramp meters, for example. Examples of concepts that NTCIP 1201 supports and defines are managing clocks, managing schedules, and managing databases.

This slide summarizes the history of NTCIP 1202 standard. The development of NTCIP family of standards began due to the need to standardize the communications with traffic signal controllers. Prior to the 1990s, every traffic vendor that provided traffic signal controllers and systems used their own proprietary communications protocol. That meant that if an agency wanted to expand a system, the traffic control system, by adding new traffic controllers, they either had to go back to the same vendor, because it was a proprietary communications protocol, or purchase a separate system. So the traffic signal industry recognized there is a need to develop a standard communications protocol, so an agency can purchase a system and, using the standard communications protocol, they can purchase—at least communicate with controllers from other vendors. So that led to development of NTCIP 1202 v01, previously called TS 3.5, and it defined objects or data elements for monitoring, configuring, and controlling traffic signal controllers.

Version 2 was approved in 2002 and, in addition to adding block object definitions to improve bandwidth efficient, it corrected some issues with special functions and added objects to improve support for signal preemption. Version 2 also defined consistency checks to be performed to ensure that critical objects are checked in context before being accepted by the traffic signal controller. So for example, before accepting a new traffic signal pattern in the controller, it made sure that all the associated rings, splits, and timing parameters were

properties configured before saving and committing that new traffic signal pattern. Version 3 was published in 2019 and added systems engineering content, which we'll discuss shortly, and added support for new user needs, including connected vehicles.

This slide shows the various components that may be found inside a traffic signal controller assembly. So in the upper left-hand corner you see the controller unit and the interface with all the different various subsystems that can be found inside a traffic signal controller cabinet. This module and the standard really focus just on the controller unit. The development of NTCIP 1202 is based on the functionality of North American traffic signal controllers as defined by the NEMA TS2 standard or specification. NEMA TS2, called the traffic controller assemblies with NTCIP A that are exchanged to enable and configure the functions that are needed for traffic signal controllers.

This slide provides an example of the linkage and relationship between the NEMA TS2 specification and NTCIP 1202. This example presents the yellow change interval for a phase. In the top, the first bullet presents the definition of the yellow change interval in NEMA TS2, and the second bullet explains how NTCIP 1202 created a data object that supports that yellow change interval. Note that the object definition in NTCIP 1202 also formally references the same clause in the NEMA TS2 specification.

This slide explains that the NEMA TS2 defines the minimum requirements for conformant devices since NTCIP defines the interface exchange data formats. In many cases they are identical, but in some cases NTCIP allows a wider range based on practical encoding limits or the market demand for broader ranges for some implementations. So in this example, the first bullet, NEMA TS2 says, "Hey, the yellow change interval should only be from 3.0 to 25.5 seconds," but NTCIP 1202 allows a value of zero seconds to 25.5 seconds. That's because of encoding rules.

In summary, NTCIP 1202 complements all the major hardware and functional standards and specifications of a North American traffic signal controller by standardizing the communications interface—in this case, the data elements. This and the next slide present a Table of Contents for NTCIP 1202 v03. The organization of the Table of Contents follows the NTCIP standard for developing field devices standards using system engineering content. That is, each standard that follows the system engineering content defines the operational needs for a device in a concept of operations, defines the requirements that best expresses those needs in detail, followed by the design content that defines how the requirements are fulfilled in a standardized manner. The design content are defined in a form of dialogs in Section 4 and object definitions, which can be found in Section 5 and sometimes through 6 and 7 also. NTCIP 1202 v02, by the way, previously only contained object definitions, but now with v03 we also define the operational needs that the standard addresses, the requirements, and the standard design content for a traffic signal controller.

This module will focus specifically on Section 2 and Section 3, Concept of Operations, and Functional Requirements. This slide shows the remainder of the Table of Contents for NTCIP 1202 v03. Of additional interest for users of this module might be Annex D, which summarizes the changes to the previously published version, or v02, Annex E, which summarizes some of

the needs and requirements that were considered for v03 but were not included. Annex F and H provide us some additional informative and normative reference materials.

We've reached our first activity. These activities are to review some of the material that we've learned in the previous learning objective. The question for this activity is: Which of the following below is not a true statement about NTCIP 1202 v03? Your answer choices are: A) 1202 is not part of the NTCIP family of standards; B) contains systems engineering content; C) describes the hardware functionality of a traffic controller; and D) contains user needs to manage a traffic controller.

The correct answer for this question was: C) describes the hardware functionality of a traffic controller. NTCIP 1202 does not describe the hardware functionality, but other standards, such as NEMA TS2, do. All NTCIP 1202 does is define the communications interface to support the hardware functionality. "A" is incorrect. NTCIP 1202 is an information standard in the NTCIP family of standards. It does contain systems engineering content in the form of Concept of Operations requirements and design content. And it does contain user needs—that is, the user needs that the standard addresses.

Speaking of user needs, we now reach our next learning objective, which is to identify specific ASC operational needs. That is, we'll be reviewing the operational needs supported by NTCIP 1202 v03, what user needs does the NTCIP 1202 standard support, discuss the tradeoffs with specifying all the user needs supported by NTCIP 1202, and discuss how to address the operational needs that are not supported by NTCIP 1202 standard.

Recall that NTCIP 1202 added systems engineering content. Systems engineering document generally begins with a Concept of Operations section that describes the current situation and problem that the system we're trying to procure is trying to solve. So what is in the box is a summary of the current situation with traffic signal controllers. Transportation system managers use ASCs or controllers to control traffic operations on a roadway. Controllers allow different conflicting movements to travel across the roadway in a safe, orderly manner. In a roadway network, controllers can be coordinated to improve mobility and certain movements such as along a major arterial. So this is the purpose of a traffic signal controller as defined in NTCIP 1202.

This slide shows the referenced physical architecture diagram from NTCIP 1202. This diagram shows where the traffic signal controller unit belongs relative to other infrastructure components or devices in the context of a signalized intersection. The diagram shows the interfaces between the traffic signal controller and also indicates which interfaces are addressed by the NTCIP 1202 standard. In this case, it's the red dotted lines. Note that that diagram also shows a DSRC modem for the roadside unit in the bottom left-hand corner. The standard, though it was only published in 2019, was created several years ago and the DSRC model in actuality may be any wireless communications technology such as C-V2X.

This slide presents some of the major problems that agencies face in the context of a traffic signal controller. Managers need controllers to manage conflicting movements at signalized intersection, need to monitor the controllers remotely, and program them to be coordinated. The standard addresses the problem where only proprietary solutions were available to

communicate with the controllers which still support trying to resolve some of the problems that are indicated in the first three bullets. As mentioned, the Concept of Operations contains user needs that describe what the ASC system needs to do to solve the transportation problem. So every user need that is listed as a unique identifier provides a major desired capability, meaning what we're trying to do, what capability we're trying to provide, has a rationale—what is the reason? Why do we need that capability?—and is solution-free. By solution-free, we mean that the user need statement does not assume a specific implementation or a design.

This is an example of a user need in NTCIP 1202. It's actually the first operational need that's supported, and it is 2.5.2.1.1. That's the section it can be found in. So you can see this user need has a unique identifier, identifies a major desired capability, which is to manage the controller's startup function, and a rationale, which also happens to be solution-free. So we need to be able to retrieve and configure the startup capabilities and functions so we can define startup times upon power-up, set the backup timer, and set the minimum clearance times for the controller. This slide summarizes the architectural needs defined in NTCIP 1202. By architectural needs, we means defines the communication systems between the signal controller and something called a management station, or control source. The management station can be the central system at the traffic management center or it could be a maintenance laptop at the controller unit that you hook up to the controller unit in the field. The first architectural need addressed by the standard allows a transportation operator to get live data from the controller. This live data allows an operator to access information—for example, what signal timing patterns are available in the controller—and to alter information—for example, command a signal timing pattern. The controller responds by executing the command and responding to whether the request was successful or not. Although out of scope of this module, NTCIP uses SNMP as a mechanism to provide live data. SNMP is designed for simplicity in processing and flexibility and requesting content, but unfortunately is bandwidth-inefficient.

So when transferring each piece of data or each data element one at a time, it can become very bandwidth-inefficient because there's overhead with every data element that you transfer one at a time. So because many legacy systems have low bandwidth communications, such as 1200-baud copper wire or wireless communications, dynamic objects and block objects were developed to allow an operator to retrieve a group of data in a more bandwidth-efficient manner. The difference is dynamic objects allow the vendor or agency to define what data is grouped together and the format and structure of the grouping. Unfortunately, it does require additional processing power at the controller and by the central system. On the other hand, there's also block objects, which are similar to dynamic objects in that it's bandwidth efficient by grouping sets of data together, but those groups of data are defined by the standard and require less processing power, but limits content flexibility. So block objects are defined in NTCIP 1202.

Another architectural need is log data. Log data is important for diagnostic purposes and in case communication is not always available. Log data also catches transient events that may occur between poles. So as I mentioned, NTCIP generally uses SNMP and communicates with field devices from a management station, such as a central system, by polling the field device at regular intervals, essentially asking, "What's your status?" once a second, or it could be more than once a second, but as frequently as once a second. The problem is that transient events could occur between polls. For example, a vehicle might enter a detection zone immediately after a pole from the traffic management center, say, at midnight, but the traffic management center doesn't poll again until five seconds later. That vehicle might have already left the

detection zone before the five seconds. So from the perspective of the traffic management center, the vehicle was never detected. It occurred after my poll at midnight and then it had already left before I poll it again five seconds after midnight. However, the log data would indicate its presence. So each logged event, if it's set up, is timestamped and can capture transient events.

The architectural need for database management allows an operator to have a controller do consistency checks on the database before committing changes to the database. So, for example, the same example as before, we have a signal timing pattern that has been downloaded to the traffic signal controller. Before committing that change, the traffic signal controller will do a consistency check to make sure that all the information for that signal timing pattern exists and is consistent—are the rings already defined? Are the phases already defined? Are the minimums already defined?—before making that change to the signal and before saving that signal timing pattern.

Recalling that the ASC only responds to requests or polls from a traffic management center or some management station, the next architectural need is condition-based exception reporting. It's a long word, but essentially it's a trap. It allows the controller to send a notification back to a center system if some user-defined event is detected. So for example, if I only poll the controller once every 15 minutes, during that 15 minutes someone may have opened the cabinet door, or the controller might have went into a flash condition. The traffic management center or the traffic operator would not know about that condition until next time it polls, which may not be for 15 minutes later. So the condition-based exception reporting allows the controller to say, "Hey, this event happened," and sends information back to the traffic management center soon after the event occurs. It's kind of an alarm system.

This slide summarizes the major categories of operational and maintenance user needs defined and supported by NTCIP 1202 v03. The major categories are to manage the basic configuration of the controller, manage signal operations such as signal timing and signal timing patterns, manage any detectors that may be connected to the controller, and to manage the connected vehicle's interface to support the connected vehicle environment. The last one is to support backward compatibility with previous versions of NTCIP 1202, but there is really only one backwards-compatibility feature, a special function output signal that was deprecated in v02. Otherwise, the next several slides will present more detailed user needs for each of the categories. Note that not all user needs presented are required to be supported by the vendor to conform to the standard, meaning some of these user needs are optional while there's a handful that are mandatory.

The first major category of user needs is to allow transportation managers to configure the ASC. This includes if you will the location of the traffic signal controller, such as by latitude and longitude, and retrieving the hardware make, model, and firmware number of the controller. The next set of user needs is to manage the communications port for the controller. This need allows the temperament manager to disable unused communications ports for security purposes, for example. Another user need that NTCIP 1202 supports is to allow a transportation manager or operator or set up alarms and thresholds for unsafe operating conditions, such as if the temperature in an enclosure reaches unsafe operating temperatures or electrical power to the controller is suspect. As you see, NTCIP 1202 v03 also supports the need to store and transmit

high-resolution performance measure data, such as the Indiana DOT at Purdue University performance measures. Note that this 1202 supports this feature but in actuality, it's not commonly used.

NTCIP 1202 also supports the need to configure auxiliary inputs and outputs and to manage the database configuration. For example, so we know, "I need to make a change to the database, and it should be version 113," and I can check it and it's like, "Oh, wait a minute, this is database version 114. Something has changed." So my traffic management center might be out of sync. In addition to configuring the traffic signal controller, 1202 allows the transportation manager to configure the signal timing patterns, plans, and the control function. This is Manage Signal Operations. So a little bit more detail, it allows the operator to manage the controller startup functions, the phase configurations, coordination configurations for arterials, manage the timing patterns, the splits, the rings, and the channel configuration. It also allows an operator to manage the overlap; the preempts; the scheduler; the action scheduler, which is a series of events; the input/output mapping, so the mapping of the different connectors. It also allows the operator to manage the communications within the cabinet and to support any ADA functions that the cabinet may have, the American Disability Act. NTCIP 1202 v03 also allows a transportation operator to monitor signal operations at the controller. So this means determining the controller health, what mode of operations is currently running, what type of flashing, the timing pattern, the current cycle, signal indication. What signal indications are currently on and to monitor the phase status?

NTCIP 1202 also allows the transportation operator to monitor signal ring status, channel status, overlaps, the preempt inputs, the preempts, special functions, time base actions, and the communications within the cabinet. NTCIP 1202 also allows the manager to remotely control the traffic signal controller. This includes configuring/controlling the minimum recalls and control the Walk Rest modifier, commanding modes of operations or activating a signal timing pattern, to omit hold phases or to place calls, activate a preempt mode, or to active a force-off. And to activate special functions, action plans, and remotely force advance the controller.

Another category of user needs is to allow the transportation manager or operator to manage the detectors associated with the traffic signal controller. These user needs include allowing the transportation manager to configure the detectors, to suggest a type of detection, whether it's a vehicle, pedestrian, transit vehicle or bicycle; manage and configure the phase assignments for the detectors, if detectors are used for calls or alarms. It also allows the operator to monitor the status of each detection zone for failures or faults while also allowing an operator to reset a detector. Finally, 1202 allows the transportation operator to retrieve detector data, such as the volumes or occupancies and speeds over a sample period—a user-configurable sample period.

So those are the user needs for the detectors. The next major category of user needs is to manage the connected vehicle interface. So that's a new feature of v03. These user needs essentially allows a traffic signal controller to generate and provide signal phase and timing information, otherwise known as SPaT. This slide depicts the logical connected vehicle system context diagram from NTCIP 1202, which shows the ASC system's interaction with the connected vehicle environment. The connected vehicle environment around the controller focuses on two distinct logical processes; one we call the ASC process and the second we call the connected vehicle roadside process. The ASC process consists of the traditional processes

for controlling a signalized intersection, possibly using inputs that indicate the traffic demand around an intersection. The CV roadside process consists of other processes, or subprocesses, that support the connected vehicle environment. From the context of a signal controller, these processes include running the signalized intersection connected vehicle applications, broadcasting the SPaT and MAP messages to connected devices, and processing the basic safety messages and personal safety messages received from connected devices.

The CV roadside process may also perform other functions such as send and manage security certificates, to configure other connected vehicle-related applications. However, these functions of outside the scope of NTCIP 1202. Currently the ASC process is performed by the traffic signal controller and for the most part the connected vehicle roadside process is generally performed by a roadside unit. However, there is an alternate configuration depicted in NTCIP 1202, not shown here, but it allows the traffic signal controller to perform both functions, both the ASC process and the connected vehicle roadside process. In effect, the controller can act as the roadside unit with the addition of an appropriate modem. This diagram also depicts the interfaces between the different entities and processes that comprise the connected vehicle environment around the signal controller. As a note, again, this diagram was copied from NTCIP 1202 v03 and shows a DSRC radio on the left, but it really is any V2X radio. Again, it could be DSRC, it could be C-V2X.

Going into the connected vehicle user needs in a little bit more detail, the user needs are organized by interface. The first interface is between the management station, such as the central system, and the traffic signal controller. The user need for this interface allows the transportation manager to configure the port used to exchange data with a roadside unit, and the polling period, how often does it poll, and to configure a watchdog for timeouts—so make sure that the interface is operating correctly—and to configure what SPaT data is provided with the RSU.

The second interface for the connected vehicle user needs is because the management station or the central system and the CV roadside process. Remember, the CV roadside process includes connected vehicle applications that uses or transmit the SPaT and MAP messages to connected devices, but also processes the basic safety message and the personal safety messages received from connected devices and perform other connected vehicle functions. The CV roadside process currently typically resides in a roadside unit but, again, could reside in the traffic signal controller. So the user needs addressed by NTCIP 1202 across this interface include managing the data from the MAP messages, the ability to create virtual detectors to collect and user data from connected devices, such as the basic safety message or the personal safety message, and to allow transportation operators to monitor SPaT and MAP data being broadcasted. As a note, this NTCIP 1202 was developed by NTCIP 1218, which is object definitions for roadside units, and NTCIP 1218, which should be published by now, provides a more generic method to monitor all the messages broadcasted by the roadside unit, and that's probably the preferred method to monitor the contents of the messages that are being broadcasted.

The last interface for the connected vehicle user needs is between the traffic signal controller and the CV roadside process, or between the ASC process and the CV roadside process. NTCIP 1202 supports the following user needs across this interface, sharing the SPaT data

from the controller to the CV process, sharing the presence of connected devices through virtual detector zones that then set up; they share what MAP data or MAP plan is currently being broadcast. The version of the MAP plans that's being broadcast is important so that a controller can confirm that the contents and the MAP data being broadcast are compatible with the SPaT data being broadcast. This is a check that's been identified and supported in NTCIP 1202. So you want to make sure that the SPaT data that you're sending works with the MAP data that you're—or compatible with the MAP data that you're also broadcasting.

NTCIP 1202 strives to be flexible with supporting the connected vehicle environment. So one of the options—one of the possible configurations right now that has to be selected for a project when implementing 1202, is to determine the ASC process or the traffic signal controller or the CV roadside process, the RSU, is the manager, meaning who initiates a request to get information or set information. So the ASC process is the manager. It will initiate the process, or it will poll the connected vehicle roadside process and will also the connected vehicle roadside process for its status, and the ASC process will send it information. "Hey, here's the current SPaT data. Here's the current data to be transmitted." On the other hand, if the CV roadside process is the manager—that is, it has to ask the ASC process or the controller for SPaT data— it has to initiate the request. So it will ask the SPaT data every tenth of a second—excuse me— it will ask the ASC process for the SPaT data every tenth of a second if the SPaT data is supposed to be broadcast every tenth of a second. So who's the manager, who's the agent, or the response entity is important. It has to be defined for every implementation. There's been arguments for both possibilities, so it's suggested that before you implement your connected vehicle interface that you work with your controller vendor and your RSU vendor to determine which configuration is supported and which configuration works best for your specific implementation.

The next major category of user needs is really security-related. So they relate to security supported by NTCIP 1202. Previously, prior to the connected vehicle environment, security was probably not a big concern because the connection from the traffic signal controller was mostly limited just to the traffic management center. Now, with a wireless interface out to the public, because of the connected vehicle environment, it opens up a new gateway to the controller and security becomes a really critical issue. So some security needs are addressed in NTCIP 1202 and that's shown here, but NTCIP 1202 really just provides a minimal level of security and other ways to address security have to be considered, and that will be discussed more in detail in one of the future modules that [will] follow this module.

So looking at the operational needs that we just reviewed, we can see that most of the operational needs, except for the connected vehicle needs and security, are functionality that's already described in the NEMA TS2 specification, which again defines how most North American traffic signal controllers manage signal operations. So NTCIP 1202 defines those operational needs specific to the traffic signal controllers mostly from NEMA TS2 and that are supported by this interface standard and how those operational needs are satisfied. In addition, there's another standard, NTCIP 1201, that defines some of the operational needs that are generic to field devices, such as clock management.

While 1202 supports many operational needs, not all features may be needed by an agency. So an agency should take care to only specify those features that are needed or will be needed

otherwise the cost to procure, maintain, and test unnecessary features for a traffic signal controller that you purchased may significantly increase. The NEMA TS2 specification is useful as a guideline on what the minimal features are and how something is typically supported, but as you develop your specification, for budget reasons and for maintenance reasons, it is important to only select the user needs that you think you will need, whether now or in the future. Interoperability is a goal of most procurements, whether that agency is procuring multiple traffic controllers in procurement phases or a central system. With interoperability, it doesn't matter which vendor you procure your controller from. That control and central system will understand information that is exchanged and use it the same way. So this NTCIP 1202 standard really goes a long way in supporting interoperability because it defines the user needs in a standardized way, the requirements in a standardized way, and the design that fulfills those requirements in a standardized way.

An example of interoperability is Wi-Fi. It doesn't matter whose laptop you use, whether you're at the airport or Starbucks. It doesn't matter which router the airport or Starbucks uses. You'll be able to connect to the internet so that you can browse, so that you can get your email, so that's an example of interoperability. Detailed specifications are important for your agency. The detailed specifications were selected by the many options available in the standard to ensure that the system that you procure, the controllers you procure, is interoperable as expected.

The previous slides in this module have presented the user needs that are addressed in NTCIP 1202. This slide presents some of the user needs that were considered but ultimately not included in the 1202 v03 standard. These include support for interval-based controllers. Most North American controllers are phase-based controllers; however, there is a significant number of interval-based controllers, mostly in the Northeast United States and with older controllers. The working group that developed NTCIP 1202 intended to support interval-based controllers during its development but it was dropped due to resources and schedule. Other user needs that were considered were non-persistent timing patterns that are just temporary and volatile, that can be generated. Traffic adaptive was not included because it could not be precisely defined and there were different traffic adaptive algorithms.

Peer-to-peer allows traffic signals to exchange information directly with each other for coordination, such as, for example, adjacent traffic signals along a light rail. With peer-to-peer, one controller can be used as an advanced warning to the downstream traffic signal controller. Additional ADA requirements were considered but they weren't well-defined enough. ADA, again, being to support visually impaired persons, for example. Advanced preempt inputs at high-speed at-grade rail crossings were also considered. That user need was also considered. There's also a NTCIP 1214 standard for conflict monitors, or it was called signal monitoring units, so an MMU or a CMU, but that draft work did not progress past user comments.

So what do you do if NTCIP 1202 doesn't address all your operational needs? NTCIP 1202 does address most of the common operational needs but just can't support all the possible operational needs that are out there so most vendors already have a set of proprietary extensions to support those desired features not supported by 1202. Some vendors openly share these extensions, including documentation about these extensions, with the agencies. Unfortunately, not all do, but interoperability, even when using these can still be obtained even if you develop extensions, but can be inhibited if the solution, the design, the data elements, is not

well documented and provided to the agency. The agency cannot distribute the design to other parties, such as another vendor, or to a central system vendor, a system integrator, for example. And just as a note, the cost to implement the design is too costly for a third-party. But what should an agency do if the user need is not found in 1202 but they still want to implement it because it is an important user need? First thing is consider whether there are other nonstandard capability—if the capability is really needed. If it is, determine alternatives that stakeholders will consider. If the agency determines it is still needed, fully document the user need in the Concept of Operations, and the requirements and the designs that trace to those user needs. Extensions unfortunately pose significant cost. Besides implementing the custom feature, there are costs associated with integrating those extensions, maintaining it, and testing those customized features. They have to be weighed against the potential benefits.

We've reached the activity for the second learning objective. So the question is: Which of the below is a benefit of extensions? Your answer choices are: A) addresses a user need that is not supported by the standard; B) addresses interoperability; C) changes the cost for testing and maintenance; or D) requires additions to the agency specifications. So which of the below is a benefit of extensions?

The answer is: A) extensions are used to support user needs not addressed by the standard. Again, it's likely that you may come across a user need that's not supported by the standard, but we still want you to use the standard to address most of your communication needs. So that's why NTCIP does allow extensions. But extensions can lead to proprietary solutions that inhibits interoperability. It does lead to additional cost for testing and for maintenance, and the agency specification needs to include the definition and description of the extension, so that's additional work that has to be performed by that specific agency.

We've reached our next learning objective, which is to describe the purpose of the Protocol Requirements List. So the PRL, Protocol Requirements List, is a table that is found in NTCIP 1202 v03. For the next couple of slides, we'll introduce how to use the PRL to determine those parts of the standard relevant to your operational needs. We'll talk about how to determine specified requirements that can satisfy those needs, and then we'll end with how to determine conformance to the standard.

Up until now, in this module we've introduced the user needs that are supported by NTCIP 1202. So all these user needs are listed in the table provided in 1202 called the Protocol Requirements List, which then maps those user needs of the standard to its associated requirements. The PRL can be used to specify the standard, meaning define the interface between a management station, such as a central system or a field technician laptop, and the controller. It can be used to define what features the interface is expected to support by allowing the agency to fill out the PRL, to check off, "These are the user needs we need for our implementation." While the standard supports many user needs, not all of them may apply to an agency or to your agency. The PRL was also designed to become part of an agency's specifications once it is completed. The PRL guides an agency to select a project's user needs. It was specifically added to support the systems engineering process. The PRL, as I mentioned, lists all the user needs supported by the standard and indicates which user needs are mandatory to be selected to conform to the standard and allows the agency to select those user needs that are optional and indicated which are needed by the agency. Once an agency selects

a user need, the PRL then indicates which requirements are necessary and which are optional to satisfy that user need. Why is the mapping from the user need to the associated requirements important? Well, from a systems engineering point of view, the user needs are descriptive, while the requirements are measurable and enforceable. By satisfying the user needs in a standardized manner and by fulfilling the requirements in a standardized manner, that helps to support interoperability. That means devices and systems that support the same user needs may be satisfied with the same functional requirements and ultimately by the same design.

The PRL also ensures that the standard is complete and consistent. Every user need must map to at least one requirement that satisfies the user need, and a user need may also map to multiple requirements but all must be fulfilled to satisfy the user need. It is also possible that multiple user needs map to the same requirement. Although not depicted here, each requirement in the standard is associated to at least one user need, which ensures that there is no user need without an associated requirement, and that there are no requirements that do not address at least one user need.

This figure shows a capture from the standard, from NTCIP 1202, of the PRL. It shows the heading in the first row, an example of the PRL content. This and the next several slides cover what the PRL looks like and how to complete it. So the first two columns in the PRL list the user needs, also called the features. The user need consists of the user need ID, which is the section number where the user need can be found, and the user need, which is a short description of the user need itself. All the user needs in NTCIP 1202, again, can be found in the first two columns of the PRL. Going to the user need itself, in this example, 2.5.2.1.9, we see that the user need is to manage the preempt configuration. Now, we have to determine—or the agency needs to determine—if this feature is needed for their specific implementation. If there is no need for preempts, whether it's for railroad crossing or maybe the fire department, then that feature is not needed for your implementation. And note, we say "needed," not "desired."

This figure shows a capture, again, from the NTCIP 1202, but below the user need 2.5.2.1.9 are the requirements associated with this feature, in this case, Manage Preempt Configurations. Each functional requirement consists of the functional requirement ID, which is the section number where the functional requirement description can be found, and the functional requirement itself, which is a short description of the requirement. So, for example, for the user need Manage Preempt Configurations, these are the requirements that are associated with that user need. The requirements are being able to enable or disable the preempt inputs, configure preempt control, non-locking memory, etcetera, etcetera. The empty space on the left indicates that all the rows containing the requirements to the right traces to the user need above them—in this case, Manage Preempt Configuration.

In this snapshot, the Conformance column of the PRL identifies whether the user need or the functional requirement on each row is mandatory or optional to claim conformance to the standard. A user need or requirement is considered mandatory by the standard if that user need or requirement is considered a basic user need or requirement for the traffic signal controller. In other words, there are some user needs that fundamentally must be satisfied and therefore some requirements must be fundamentally fulfilled by all signal controllers.

For this slide, we'll talk about the optional. So some user needs and requirements are mandatory, but others are optional. So in this example, O.1, open bracket 1..(1), means that this user need is optional—that's what the O stands for—is part of group 1, but one and only one of the user needs in group one must be selected. For this example, the specifier must indicate what type of cabinet the controller is located in. Some needs or requirements in the standard explicitly depend on the type of enclosure the controller unit is enclosed in, and additional details about the cabinet types can be found in this student supplement. In this example, five options, as depicted by Group 1, O.1, are shown, and only one must be selected, as indicated by the "1" in the parenthesis. If it's 1 dot, dot, star, that means one, or multiple, or all of the options may be selected, but that's not the case here.

Under Conformance, sometimes we also see a predicate. In the example shown, the predicate is preempt. This means that the user needs, 2.5.2.2.8 and 2.5.2.2.9 are optional if the preempt user need shown in 2.5.2.1.9 above is selected. If the preemption user need is not selected, then this user need is not applicable and therefore cannot be selected. So in this case, 2.5.2.1.9, if it is selected, then 2.5.2.2.8, the functional requirement of this user need, Monitor Preempt Input State, can be considered, and is mandatory. On the other hand, if 2.5.2.1.9 was not selected, then this user need does not apply and is not applicable. It's the same thing for 2.5.2.2.9, this requirement, but it's optional, meaning if the user need 2.5.2.1.9 above is selected then the ability to monitor the preempt state is optional.

For Support column, for each row, the user or agency must decide by circling an option under the Support column if that user need or requirement is to be supported. For mandatory user needs and requirements, only "Yes" appears. For optional user needs and requirements, "Yes" and "No" appears, and the user or agency must circle which one; "Yes" if that user need or requirement is to be supported or "No" if it doesn't have to be. If the user need and requirements have a predicate, then "NA" appears to allow that user to indicate that user need or requirement is not applicable because some condition isn't met. Note the contents of the Conformance column takes precedence over the Support column. If there any discrepancies between the contents of the Conformance column and the Support column, the contents of the Conformance column takes precedence. So for example, if the Conformance Column indicates mandatory for a row, Support should be "Yes." And finally, the last column of the PRL table allows specifiers to enter additional project-specific requirements. Some of the roles have predefined content which requires the user to add desired valued, but only when the user need and the resulting requirement are selected. In this example, the user has to specify how many signal timing patterns and the number of signal timing patterns that is required to be supported by the signal controller if the user need Manage Timing Patterns is supported.

This is just a snapshot of a different part of the PRL, which presents a different type of information that has to be completed under Additional Specifications. This is a functional requirement for I/O mapping and it specifies when a new I/O map can be activated. So, for example, we checked Cabinet Door Open. The Cabinet Door Open must be opened before the I/O, the new I/O map, can take effect. The PRL table can be obtained from NTCIP or copied but agencies are not allowed to alter the PRL except to complete the Support column and fill out fields in the Additional Specifications column. This protects vendors from having the agency make a slight change in a very complex specification and only finding out later, "Oh, you changed something." From an agency's point of view, NTCIP 1202 defines the user needs and requirements for the communications interface between a central system and the traffic signal

controller. It identifies what features and functions the ASC system must support. The completed PRL with the selected user needs becomes a checklist for validation: Did the vendor provide the right system? The PRL, again, also aids in interoperability by satisfying the user needs with the same requirements as defined by the PRL. Systems that conform to the standard or the PRL can achieve interoperability—that conform to the same PRL or comply with the same PRL. Sorry.

From a vendor's or system integrator's point of view, the PRL standardizes the procurement specification format for the communications interface and the functionality, allowing the vendor or systems integrator to quickly determine whether its system can satisfy the agency's user needs, and it can provide a PRL for their products also to demonstrate what is supported by their products or systems. The vendor's PRL for their standard products also shows what optional user needs that system or product supports.

This slide compares conformance versus compliance. So conformance means to meet a specified standard. Each standard generally has the conditions on what conformance means so to claim conformance to NTCIP 1202 v03, the vendor shall minimally fulfill the mandatory requirements selected. Vendors providing features beyond the completed PRL are conformant if those features conform to the requirements of NTCIP 1202 v03 and its normative references. Compliance, on the other hand, means meets a specification such as a PRL. The difference between conformance and compliance will be discussed much more in detail in the next module. The conformance definition for 1202 can be found in section 3.3.2.1 of the standard. Highlights of the conformance statement includes as follows: A device may support data that has not been defined by 1202—this is the extension—however, that data will be properly registered with a valid object Identifier. The object identifier is a series of numbers that's uniquely assigned to the object, so we know, "Oh, this is what object this is," and NTCIP has procedures for properly registering an object identifier.

To claim conformance, an ASC shall be provided with a Management Information Base, or MIB, that contains all non-NTCIP-standardized object and block definitions. So the MIB is like a text format but in a very specific format that's machine-readable and contains the definitions of all the object definitions. And finally, to claim conformance, an ASC or controller shall use the NTCIP 1202 standardized objects to manage functionality, meaning if you have an object— NTCIP 1202 has an object, for example, to change the timing pattern that's going to be used, then that controller has to use that object and not some other object as a backdoor way of controlling the controller.

We've reached our next activity. The question for Learning Objective 3 is: Which of the following is a benefit of the PRL table? Your choices are, A) maps needs to requirements; B) provides a list of features supported by the standard; and C) provides a convenient checklist during deployment; or D) all of the above.

The correct answer to this question is D, all of the above. All of the above statements are true. Being able to map needs to requirements is one of the key benefits of the PRL table so we can see which requirements are used to satisfy a user need. The PRL does list all the user needs or operational needs supported by a standard, and it can be used as a checklist for testing. So you can see which user needs and which requirements are supposed to be included in the system

or controller I have purchased and implemented, and you go through the table as a checklist for testing to make sure that you've fulfilled all the requirements and satisfied all the user needs for your project.

The next learning objective is to discuss how to prepare a project-level PRL. So we've introduced how—well, the point of this learning objective is to introduce how the completed PRL can become part of the agency specification, to use to specify the communications interface portion of your controller system.

This table is a simple example of a completed PRL. Your specification for your controller or system of controllers should have a fully completed PRL. The PRL should be based on the NTCIP 1202 with an SNMP database. By competing the PRL, simply by circling "Yes," "No," or "NA," and completing the additional specifications column on the right for the appropriate requirements, you have defined all the user needs and requirements that your controller system or controllers is expected to support. So this is just an example. We've selected "Yes" for all the mandatory user needs. We've selected "Yes" for all of the requirements that are associated with a user need that's been selected. As an example, we did select "No" for Configure Pattern Maximum Mode. We filled out how many timing patterns the controller must support, and we've indicated what types of signal patterns are to be supported by the controller.

Some key points for completing a project PRL: each agency should not blindly accept all the user needs supported by the standard; really only select the user needs that are relevant to your agency, whether it is now or in the future. Keep it simple; less is more. PRL has to be consistent with the hardware specification. For example, please indicate which cabinet type the controller is expected to be placed in, and thus supported. If you only have 24 channels, don't expect to support more than 24 channels. If you don't have preemption or don't expect preemption, don't select that specific user need. So it has to be consistent. If you don't have a temperature monitor in your cabinet, don't select the user need for temperature.

This is just another example of selecting user needs. In this example, there's a group of options and this group is O.10, and to complete it, the agency should indicate at least one of these should be selected, and this group of options are requirements for configuring the coordination point. So in this example, we selected two—we're allowed to select one. The star * means you can select them all, but we only need these first two. Maybe your agency only does coordination by the beginning of green. So it says select two, and if you don't do it by the end of green or end of yellow, just don't select that. Chances are the vendor or system has that capability anyway, but if you don't need it, don't select it.

This is another example. This is for the Manage Controller Startup function. It is mandatory, so the specification must select "Yes," and underneath that, for the mandatory requirements, the specification must select "Yes" also. So you are expected to select all the mandatory user needs—in this case, Manage Controller Startup Function—and you are expected to select "Yes" for all the mandatory requirements, which in this case, configures startup flash time, enable/disable automatic pedestrian clearance time, and configure the backup time. Configure Startup All-Red Flash Mode is optional, so for this example we've selected "No." In summary, the PRL has all the user needs and associated requirements in one place, together with a solid

relationship so by completing the PRL table, you are really specifying NTCIP 1202 and those parts of NTCIP 1202 that the project is expected to implement and support.

Going back to extensions, if extensions have been defined for the implementation, those extensions are being defined in the PRL also, both user needs and requirements. It does create your own PRL table with the same headings, but fill it out with your user needs and requirements. Non-standardized user needs should be added and traced to the standardized requirements and user-defined requirements as appropriate. Sometimes it's a standardized requirement, sometimes you have to create your own requirements to satisfy your specific extended user needs. Non-standardized requirements should also be added and traced to the standardized user needs. You may have a user need for preemption but you have an additional requirement, so just indicate that in your PRL extension. The PRL should also indicate if conformance to a non-standardized user need or requirement is mandatory or optional. Although it may seem counterintuitive that user-defined user need or requirement can be optional, different configurations of traffic signal controllers may be included in a single procurement. So a set of controllers may require support for a user need and requirements while another set, maybe purchased at a different procurement phase, may not need to support those user need and requirements.

Just recall the completed PRL becomes part of your overall ASC specification and contract document. While the PRL addresses needed features and functionality, it does not address what your hardware requirements are, what your environmental requirements are, or what your performance requirements are. Your complete contract documents will also contain terms and conditions, prequalification requirements, etcetera, product specifications. Scope of work may include your hardware specs, your software specs, if any, the communications interface spec, which will consist of the PRL containing user needs and requirements to be supported.

It needs to be understood that the user needs and the functional requirements, which are the topic of the next module, are not standalone items. They have a tight relationship with the desired functionality of the overall device that you're hoping to acquire. For example, if the user need to manage the preemption features is desired, then the hardware requirements must include support for including a preemption device in the cabinet assembly.

We've reached our last activity. Which of the following is a false statement related to an ASC specification? Your answer choices are: A) an ASC specification should include a project PRL; B) conformance requires satisfying all mandatory user needs; C) the vendor must comply with the project PRL in the agency specification; or D) compliance requires only satisfying the user needs in the standard and not the specification.

So the correct answer is: D) compliance requires only satisfying mandatory user needs in the standard and the specification. Actually, the vendor must satisfy all the mandatory user needs and all selected optional user needs in the specification. It's not only satisfying the user needs but, fulfilling all the requirements, also satisfying any optional user needs that you have indicated in the PRL. "A" was true. ASC specifications should include a PRL to describe the communications interface with the controller. All mandatory user needs must be satisfied to claim conformance to the standard and the vendor should use the project PRL to indicate which requirements it will fulfill.

In summary, this is what we've learned from this module. First, we learned what the purpose of NTCIP 1202 v03 is and where it belongs in the family of NTCIP standards. We revealed the operational needs that are addressed by the NTCIP 1202 v03 standard. We've learned how to use the PRL in NTCIP 1202 to determine conformance. And finally, we've learned how to use the PRL to develop a project specification.

This is actually the first of a series of modules for traffic signal controllers. In this module, we've introduced the user needs. There is a next module that follows this series, which is A315b, Part 1 of 2—it's actually in two parts—and it's for understanding requirements for actuated signal controllers based on NTCIP 1202 standard. The learning objectives for that module are to identify the 1202 standard requirements. We've gone through the user needs, and this module discussed the requirements. There is another matrix table that's in 1202 called the Requirements Traceability Matrix, and this module will introduce what's in the Requirements Traceability Matrix and how to use it. It'll talk about how to prepare a project-level Requirements Traceability Matrix. Again, just like we have project-level Priority Requirements List or PRL, this module will introduce how to create your own Requirements Traceability Matrix, and it'll give a little bit more detail on how to prepare an ASC specification. There is a Part 2 to that also that goes into a little bit more detail on some of the implementation issues that have to be considered when deploying and implementing a signal controller system. And then finally, the fourth module is T315: Applying Your Test Plan to the NTCIP 1202 Standard.

Thank you for completing this module. There is a feedback form that can be found with this module, so please use the Feedback link to provide us with your thoughts and comments about how helpful this training was so that we can learn from it and as we develop future modules or update this module we can address some of your concerns and provide a better quality module. Thank you again for joining us, and this concludes the module.