

Nicola Tavares: Welcome to the ITS Standards Training.

Ken Leonard: ITS standards can make your life easier. Your procurements will go more smoothly and you will 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 are 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 will 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.

Nicola Tavares: Throughout the presentation this activity slide will appear indicating that there is a multiple choice pop quiz following this slide. The presentation lecture will pause at each quiz section to allow you to 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. Please help us make even more improvements to our training modules by completing the post course feedback form. This module is A315a: Understanding User Needs for Actuated Traffic Signal Controllers Based on NTCIP 1202 Standard. Your instructor Ken Vaughn received his bachelor's degree from Tulane University and his master's degree from Texas AMU. He coordinated traffic signals in Los Angeles County in the early '90s and his graduate research study, "The Performance of Actuated Signals in Coordinated Systems". In 1994 he became involved in ITS standards and was the founding member NTCIP Joint Committee in 1995. The next voice you will hear will be of your instructor.

Ken Vaughn: Okay, thank you for joining us and thank you, Nicola for that fine introduction. We'll get right into it and we will see that the target audience for this slide, hopefully you will be included in one of these categories includes traffic management personnel and engineering staff. These people will understand the needs of a signal system and thus what they need out of this particular effort. TMC/operations staff will of course be required to operate these systems and we'll give you a lot of information on what they need to do on a day-to-day type basis. Traffic signal maintenance staff of course also are a good audience because they have to deal with the signal controllers out in the field all the time; they need to know what capabilities they have. System developers will be required to integrate with these so they are also another target audience, and then public and private sector users including manufacturers that have to comply with these standards, have to be aware of them. So all of these are part of the target audience base, hopefully you fit in one of those categories and hopefully also you've taken some of the recommended prerequisites that we list out here. There are quite a few. This is a fairly advanced course so prerequisites include starting all the back to the beginning using ITS Standards: an overview, a very high level overview of what ITS standards are. A101 was an introduction to acquiring ITS Standards and that's just an overall introduction of how you acquire them as opposed to what the standards are. That then followed on to A102 which is an introduction to user needs identification, what the user needs are and how they relate to standards. A201 then followed that on with some details about how you acquire standards. A202 then helped you write user needs when you don't have that in the standard itself which is the case for signal controllers and we'll discuss that in detail in this project but A202 provided an overview of how you do that generically for all standards that don't have SEP content. A103 then included an introduction to ITS standards requirements development. And A203 was kind of an expansion on that based on what happens when you don't have that content in your SEP standard which once again is like our course here. And then finally that led on to C101, which is introduction to the communication protocols and their uses in ITS applications. And essentially that's all the lower layer standards that you deal with and how you pick and select from all the choices you have at those lower layers. This course focuses more on the higher level but you also need that lower level to put together your full interface specification. Then beyond those basic courses we also recommend that you have a basic understanding of the Advanced Transportation Controller, ATC Standards if the ATCs are to be used. Also knowledge of American-style traffic signal operations such as actuate traffic signal control and a basic understanding of NEMA TS2. The curriculum path for Non-SEP standards including this standard is shown as here same exact courses we just mentioned before and then the course we are on at the end there is A315a, understanding user needs for ASC based on NTCIP 1202 standard. And this module will go into great detail about how you specify user needs for signal controllers and there is one course beyond this that specifies the

requirements and trains you on how you specify requirements for your particular project. And that course will actually be done two separate modules but we'll get to that at that point...

Ken Vaughn: There are learning objects for this course include six different learning objectives. Reviewing the structure of the standard, evaluating ASC-specific operational needs, and then identifying and writing user needs for the ASC. And appreciating the trade-offs between stakeholder desires versus the realistic requirements making sure you don't kind of over-require on your project that gets you into high-cost situations. Briefly explain how to evaluate conformance to the standard itself, and then introduce the relationship of needs to requirements and conformance...

Ken Vaughn: So the first learning objective is reviewing the structure of the standard and we'll be able to identify where the standard exists within the framework, understand the content of the standard, and its relationship to NEMA TS2...

Ken Vaughn: Now this is a chart you probably have seen in the other modules but we'll go through it one more time. It's the NTCIP Family of Standards; this is called the NTCIP Framework of how all of the standards fit together. And as you can see by their notes, we are up at the informational level on this particular module dealing with ASC information which is part of the NTCIP data dictionaries which is the 1200 series of standards. And this standard NTCIP 1202 is also tightly referenced and coupled with the Global Objects Standards NTCIP 1201. That Global Logic standard defines a lot of kind of generic data concepts that are referenced by this standard as well as other standards so that's kind of a separate standard for management purposes but it is very much included, most of those objects are included within the **signal** controller as well. A little bit of history of 1202. Initially when it was first approved all the way back in 1996 it was one of the initial group of NTCIP standards and actually at the time they were called NEMA TS3.5. All of the NTCIP standards were to become NTCIP standards were NEMA TS3 standards at the time and later those NEMA standards were also adopted by ITE **and** AASHTO and they became NTCIP standards. Amendment 1 was drafted in '99 but it never actually was finalized. It got rolled into version 2 and version 2 was approved in 2005 and the big change there was adding block object definitions to make the protocol more bandwidth-efficient and more acceptable to the industry. Now what does NTCIP contain the 1202 standard? Well, as we mentioned it is a data dictionary but specifically that data dictionary is designed to communicate between actuated traffic signal controllers and the monitoring system. That monitoring system may be a central management system or it could be a signal system master so either one of those two your traditional monitoring system, communicating with the traffic signal controller and this standard defines those data elements. Those data elements are all defined around the functionality that was originally defined in NEMA TS2.

NEMA TS2 is a well-recognized standard that defines the functionality of Americanized traffic signal controllers particularly actuated traffic signal controllers. What's not included? There are some features like adaptive signal control that is not defined within the NTCIP 1202. You can extend it and we'll talk about that toward the end of the module but that's not currently defined within the standard because there are so many different ways to do adaptive signal control. Also not in the standards is the SEP content which includes your user needs, functional requirements, message dialogs, and test procedures. So that's what we're going to talk about during this module some is the user needs. The next module we'll talk about the functional requirements and the message dialogues and then the future module beyond that on the testing path we'll talk about test procedures. Now how does NTCIP fit into the overall package? When you go out and buy a traffic signal controller per dollars procurement, there are all sorts of details you have to be worried with and they include things like your hardware specification, your functional specification and how that equipment is going to interface with the rest of the world. And it's really that last piece that NTCIP deals with is that interface specification. And NTCIP 1202 is one component there that references also 1201 but there are also other lower layer standards you have to worry about and your own agency specifications. They all fit into that interface specification which then fits into the larger procurement package, the overall picture. Now within the standard itself there are a couple of sections and a few annexes. We'll talk about this real quick. The first section is a general section and it's what you see in pretty much every standard. There's a scope. It defines terms and it defines definitions and acronyms. Section two is the real meat of the standard. It defines your data dictionary. It's the object definitions, SNMP or NTCIP uses what they call objects for each individual data element and that's where all of these are defined for your ASC functionality. Then you get into the Annexes. Annex A is the information profile. It groups all of the objects we talked about before into a set of conformance groups and then those conformance groups can be used and referenced as an item kind of. And so it groups all of your face **phase** parameters into one group and the **detector** parameters into another group. So we'll talk about that later. Annex B is consistency checks and there are a whole range of objects. When you are data downloading into a signal controller and configuring it, all of these different data parameters you are setting and the controller have all sorts of interrelationships among them. And if your data that you're downloading doesn't agree with each other, then you want to make sure the signal controller rejects those commands and the consistency checks are the standardized procedures that have to be followed by an implementation to make sure that any download is properly reconciled before it's implemented into the controller. So this is really kind of a safety procedure that goes through to make sure all the signal controllers follow. Annex C is called Concept of Operations. Now it's not the concept of operations you might see in other SEP standards. It's really more of a design pattern set so it's a very low level design set of how the protocol works at a very low level as opposed to the high level concept of operations you might see in other SEPs in other standards. And then finally Annex D is called Deprecated Objects. These are old design objects from previous versions of the

standard. Any time you see a version 2 or even a 1.1 there's a good chance that they've deprecated some objects in the original standard and we put these off into a certain separate section. That's where Annex D comes into play. I think there's one object in there that we'll talk about a little bit later. Some of the advantages that NTCIP 1202 offers it does a great job of defining the detailed common design elements and their definitions. So all of the little details that are so easy to forget about in a large specification it defines each one of these very, very precisely and it makes sure that all the details are covered and they're all done the same way by different manufacturers which makes it a very, very solid specification. It also defines a common baseline of conformance. By defining those conformance groups it allows you to group objects together and say I want to support this feature or not and that's also what we're doing a little bit better job of here in defining user needs so that everything maps very, very consistently. Next also defines a set of consistency checks that must be performed. That was that Annex B that we talked about. And then finally it's a step towards an interoperability and interchangeability. All of these features we just talked about is a step towards that. It does not yet provide a complete solution. We recognize that. That's just only one portion of your complete procurement specification but it's a key component and we're moving ever closer to trying to standardize these products so that you can get what you need very easily...

Ken Vaughn: So how does the NTCIP 1202 relate to say NEMA TS2? Well TS2 is probably the most well-known standard in the U.S. for traffic zone controller logic. Now I emphasize the term "logic" here. There are also some definitions of hardware within that standard and there are various standards that define signal controller hardware. But as far as the logic of signal operations, it's really NEMA TS2 that defines it for the entire industry for actuated signal control. Once again, it does not include adaptive control but for traditional actuated signal control NEMA TS2 is the standard that defines how things work. And the NTCIP supplements this by defining the individual data elements that get exchanged across the wire in order to enable that functionality. So for example NTCIP 1202 will actually reference phaseYellowChange, that particular data element in NTCIP 1202 references has an actual statement in its clause referenced NEMA TS2 with specific clause numbers. And to give you an example of how that reference is if you compare those two clause numbers between NEMA TS2 and NTCIP 1202 you find that their definitions are virtually identical. A little bit of a change in terminology to be a little bit more precise in NTCIP 1202 within its context. Here we are actually defining the parameter whereas NEMA TS2 uses a more generic term timing control however that is done. And NTCIP we are actually defining how it is done. It's done through a parameter and we actually change that terminology a little bit. But other than that the definitions are identical. So really NEMA TS2 is defined how things work. We are defining the data elements of how it works within NTCIP. And to give you a little bit more example, the same example NEMA defines this element this concept as a Yellow Change interval, has a minimum range of 3-25.5 seconds with 0.1 second increments. While NTCIP we're actually

concerned about exchanging the data so now we're concerned about exactly how is this represented over the wire? Do you send texts saying 3.0 seconds? Or do you send a real number saying 3.0 seconds? Or do you send an integer? And what the designers of NTCIP says it's a lot more efficient to send this value as an integer of 30 rather than 3.0 because of the way things get encoded and all of this, technical issues we don't need to get into. But they've standardized within the industry that it's always exchanged as an integer and a value of 3 seconds gets encoded as the value of 30 because its units is tenths of a second. Now, you may also notice that the minimum range in TS2 was 3 seconds to 25.5. And NTCIP it allows you to go all the way down to zero and this is because NEMA, they were defining a minimum range. Well NTCIP is concerned about how does it look over the wire? We need to fit this into one bite that allows us to have a range from 0 to 25.5 but it's not going to restrict you. Whereas NEMA says you must support at least 3 and above. NTCIP says you can go even lower. We're silent on whether or not we're recommending that and in fact most agencies probably would not want it to go lower, but the NTCIP standard allows it to on the basis that that's what could be supported on the protocol the way we've designed it. An implementation may choose to restrict that to the NEMA minimum range of 3.0 and above and that would allow it to be much more in line with how controllers operate. It actually gives agencies a level of safety there that they don't have to worry about someone accidentally encoding a one-second yellow clearance that they know is always going to be at least 3 seconds. At least they get that protection but that's up for the agencies and the manufacturer community to decide. NTCIP says this is the range that's possible. So NTCIP does not contain user needs but it is based on established standard NEMA TS2 that defines core functionality. So you can trust that the logic contained within NTCIP is well proven. NEMA TS2 has been out there since the '90s and actually it was a follow on from TS1. It goes way back to even before then. So these are well-proven concepts that NTCIP is adopting even though it didn't formally define user needs...

Ken Vaughn: Well that brings us to our first activity and this is a question. Which item is not a part of NTCIP 1202? So which item is not a part of the standard? Consistency checks, information profile, C is user needs, and D is object data definitions. So once again, A is consistency checks, B information profile, C user needs, or D object data definitions. Which one is not in NTCIP 1202?

Ken Vaughn: And you see here that the answer is user needs. The NTCIP 1202 does not contain systems engineering content. It does contain consistency checks. They are contained in Annex B. The information profile is contained in Annex A and provides you information about conformance groups and how everything maps. Then finally object definitions, that's the main meat of this standard contained in section two...

Ken Vaughn: That concludes learning objective number one. We discussed about the standard defines the data elements for information exchange and it is based heavily on NEMA TS2. It complements the hardware standards and specifications that are available out there that includes NEMA controllers, 170 controllers, 179 controllers, ATC controllers, all of these will inter-operate with NTCIP. And it does not include systems engineering process content such as user needs. Now learning objective two, we are going to evaluate the ASC-specific operational needs. You will be able to evaluate your own operational needs and understand how the standard meets your particular needs. You'll recognize how TS2 and 1202 are completely intertwined and understand constraints which will impact cost benefits. Finally you will become aware of the issues that you'll encounter when you preparing a procurement. Now when we talk about developing user needs one of the prerequisites we already talked about was course A202, Identifying and Writing User Needs When ITS Standards do not have SEP Content. That was a very generic sort of overview of how you develop these user needs. We're now going to apply the process we talked about there to this particular standard. The first thing we need to do is recognize that the user needs is only one component of your overall concept of operations. And the first part of the concept of operations is understanding what your big picture is, what is your current situation and problem? Who are the users? Who is affected by the system? What are the scenarios? How is the system going to work? And what are the needs of the users of the system? And then finally are there any regional aspects? How does this apply with the regional architecture and other issues? NTCIP makes assumptions about these questions, but as we mentioned it doesn't really define them within the standard itself, and you'll need to specify the use for your own ConOps. Now for the Current Situation, Problem Statement, the general concept we have when we wrote NTCIP 1202 is that agencies are responsible for managing the operation of the transportation network. And part of the transportation network are intersections. And conflicting movements at intersections must be managed to prevent collisions and sometimes you can't really achieve this by just simple yield or stop sign control. You need something along the lines of a signal controller to sequentially grant right-of-way specific movements for efficient operation. That kind of states the situation. The problems involved are signals are sometimes located near one another can impact each other's operations so you need to coordinate these signals and from a practical management perspective agencies want to manage these signals remotely so that they don't always have to be going up there changing things at the local controller. And prior to NTCIP there was no industry standard for these remote communications. Everything either had to be done manually or more typically through proprietary solutions. So the intent of NTCIP was to standardize how you interface these signal controllers remotely...

Ken Vaughn: Another part of the concept of operations is the architecture and understanding and laying out how all of the pieces go together. So this graphic shows you the typical intersection has, a signalized intersection, have signal controller hardware that

the driver sees. Typically the signal heads and whatever. Those signal heads are connected to a signal controller which is then eventually often connected to the control center at some remote site. Now the operational logic of that signal control is defined finding the TS2, as we mentioned that's the functionality of how things operate. This signal controller hardware itself, well there's a variety of standards and specifications that define how they go together. NEMA TS2 has one set of standards, ATC is a different standard, 170 is a specification by CalTrans, 179 is a specification by New York State so there are any number of those specifications and standards that can define the hardware. The key for the information here is they all can work with NTCIP. That NTCIP standard then defines how that signal controller hardware connects to the control center remotely and then allows the two to communicate which eventually effects the operation of the local signal controller. Now there are various steps to writing a user need and this is all taken from course A202. We define that there are operational needs, from that you can figure out which ITS standards to go to, once you have the ITS standards you extract data from those standards, and then you write your user need. So we'll go through these step by step...

Ken Vaughn: How do you define your operational needs? Well there are six key sources for this. The first is regional architecture. Now regional architecture is a fairly high level view. But it does impact how everything fits together in your region and you should look at that to see if there are any specific functions assigned to the local controller. Now it's a high-level view. Signal controllers are pretty low-level so you may not find a whole lot in there but you should still look at that document for anything you may find. The second is project description. Now if your mayor is coming to you and saying "Well, we need you to deploy the signal system in our city and oh, by the way I really want to use this advanced feature," or something or "be able to monitor on a second-by-second level" or something, that's going to be a very important feature to make sure you include in your concept of operations. So your project description whoever has told you to do this project they are going to have their own particular set of features in mind that you need to make sure are included in this project and get included in the concept of operations. Operational scenario is the third resource and this is where you work with your various regional stakeholders and discover what they want out of the system as well. You want to make sure when you deploy a system like this you are spending a lot of money. Make sure you get a system that's going to meet everyone's needs at least the best you can with cost considerations. So get everyone around the table. Make sure you know what you need before you start spending the real money. The fourth item is looking to the NEMA TS2 standard and that is the standard that defines functionality for the traffic signal controllers and it is the main standard. So understand all of the different features and which features you may need and which features you may not need for your project. Likewise NEMA TS sorry, NTCIP 1201 and 1202 also define very, very specific pieces of data and they can also be used as a resource to identify what needs you may have on your particular project

and we'll go into a lot more detail about how you can pull that information out of 1202 in just a few slides. Then finally there are all sorts of case studies and industry experience for whatever we're doing in ITS it's worth looking at case studies and trying to derive what lessons have been learned by others so I can capture that information in my deployment so that I know what I need to do. So now lots of different resources to look at, six different resources, how they all interrelate and that's what we're trying to show here. Your regional architecture is a very, very high level architecture, but it encompasses the entire realm of your regional system and you need to know that to understand properly how your particular project is going to fit into this larger picture. So it's worth being familiar with that regional architecture, making sure that you are doing everything that you need to do to integrate with that. The second was project description. That's going to be a lot more focused to just the details of your project. It's going to go a little bit deeper probably than the regional architecture does. That's another key resource. Then finally scenarios. That's where you really start getting into the meat of the system is figuring out how the stakeholders are going to use the system. And then you also have the standards, ITS standards once again to cover the range of just about everything that's out there. There is a small set of those standards that you are going to need to be familiar with for this particular project. And those standards will go into great depth into what needs to be done, but there is a high level aspect to those standards that can provide you a lot of great insight and what you need to consider during the concept of operations in defining your user needs. Finally you have the case studies. Once again these cover a huge range of virtually everything you can do in ITS but there's going to be a subset of those within your particular project area that relate to your project that you can find out what other agencies have done, what experiences they have had, what lessons they learned so you can feed those into your project as well. And all of that information then goes into feed your ConOps. And all of that helps you write and refine that concept of operations so that you have a very good description of what your system needs to do. Now as any good engineer I've also put my not to scale here. These are going to be varied by your particular project, how wide it is and everything else...

Ken Vaughn: Now there are various pitfalls going through this process. If you fail to consider any one of these sources it can lead to real problems. So if you don't consider project description you may not identify some of the capabilities. If you don't consider the architecture, you may double up on some of the functionality for example where does your scheduling logic go? Is it only in the controller? Is it only at the central system? Do you have it both for redundancy purposes? All that needs to be considered and that's really an architectural level concept. If you fail to consider scenarios, you may inadvertently ignore some of the needs of your stakeholders like performance needs or something. You build an entire system and you find out that your system is not really meeting the needs of your users. Well now it's very expensive to go back and fix those things so if you capture them up front in your user needs you've done yourselves a lot of favors. So this is largely true,

all of this is largely true, even the SEP standards. You need to go through this entire process. The SEP standards allow you to facilitate the process of getting the information out of the individual NTCIP standards but all of this upfront work really needs to be done regardless of whether or not you have SEP content in your NTCIP standard...

Ken Vaughn: So now that we know that background, let's go through the individual steps of identifying the writing the user need. The first is identifying the operational needs. We've identified the resources of how identified needs but what are those needs? Well you end up with a huge number of needs from this effort particularly from the stakeholders. And one good way to kind of start categorizing these needs is to always start thinking about configuring the device, controlling the device, and monitoring the device. And if those three pretty much all operations you deal with can be categorized to one of those three groups and if you're constantly coming back to those questions am I addressing all there of those issues on all those different principles? Then you have a fairly high confidence that you are catching everything. So that's a good way to categorize everything and we'll go through some practices here pretty soon to show you how to do that...

Ken Vaughn: So to configure maintenance desires at a high level now if you look at the operational needs we identified from our various six resources. You may find that your maintenance group desires a single ASC model that can be configured for any intersection in your jurisdiction. They want to do this probably because they only want to have one spare controller on their maintenance trucks and if they only have one spare controller that saves them inventory, it saves them resources from going back and forth to the maintenance house, warehouse and all of that. Management on the other hand, well they want safety features in the controller so that no one can reconfigure safety-critical information from a remote location. They don't want to change your phase assignments so that you can change the signal heads and everything which indications they are going to indicate. So they want to make sure someone is really configuring the base logic of the signal controller that someone is there in the field so that's maybe another item that comes into play. A third item maybe from a controlled perspective is operator's desire to control the selection of timing patterns. When an accident occurs on the freeway they may need the ability to quickly switch over to timing pattern on a parallel arterial street so they have that need. And then finally operators' desire to monitor real time signal operations and maintenance desires monitor signal diagnostics to detect equipment failures. So all of these different and this is just obviously one small set of potential user needs or operational needs that we need to refine and to properly categorize and define user needs. We also need to consider benefit cost considerations. Everything you do needs to be within budget constraints. Every single user need that's identified you need to ask is this really something that we need to support given the cost implications involved? If it's not standardized you may be paying a pretty penny to get it deployed. And you may

also run into implementation issues because it's a new feature that no one has done and everything else. So but even if it is a standardized feature things like how many faces do you really support? We talked about the NTCIP kind of only arranges things at a byte level so the NTCIP standard says that signal controller can have up to 255 phases but in practice I can't think of any jurisdiction in the country that has a practical need for 255 phases. Traditionally it's known as an eight-phase signal control so traditionally it's been eight phases. More recently people have upped that some to maybe 16 and some even more than that but the question becomes how many phases do you really need to support? If your specification says 255 you are probably going to restrict the number of manufacturers that put a bid in on your project so these are questions you need to ask to say how many do I need to support because that's going to impact memory issues, my controller and everything else. Also in my customized features, are they really justified? What kind of cost is going to be involved? Now the good news is NEMA TS2 actually provides a baseline of what industry typically supports as a minimum. They define the number of phases, the number of detectors and all of this. So you can look at NEMA TS2 and say anything that it requires there's a pretty good chance that the industry as a whole will largely require this. If you start going above those numbers you perhaps should start dealing with the industry and talking to them to see if this is something that is available in the marketplace or if it's going to be a customized product. Finally interoperability requires significant agency specifications. You can imagine that if you are expecting your signal controller to support 32 phases and someone delivers you a controller that only supports eight phases, it's going to break. You're not going to be able to control phase 32 if your signal controller only supports eight. So your agency specifications need to detail all of these issues so that you get a controller that will actually work all the way end-to-end...

Ken Vaughn: That brings us to our second activity. And the question is which of the following can be used to discover operational needs? A regional architecture, B operational scenarios and stakeholder input, C ITS standards, or D all of the above? So go ahead and make your choice from those four options. Which of them can be used to discover operational needs?

Ken Vaughn: So the answer well the correct answer is D, all of these tools can be used to identify operational needs. The regional architecture, the answer is incorrect but obviously you can use the regional architecture. It defines your high-level architectural needs but the other answers are also correct. Operational scenarios and stakeholder input define mid-level user needs but the other options are also correct. And then finally ITS standards define very low level user needs but the other options are also correct...

Ken Vaughn: So that concludes learning objective number two. We went through identifying operational needs is an important part of the process and they should be

derived from six different sources that were identified. The definition of needs will impact your project costs and the baseline features are defined in NEMA TS2...

Ken Vaughn: That brings us to learning objective number three. Identify and write user needs for ASC. Understand that ASC, we will understand that standardized needs do not exist for signals. Understand how to extract and write these needs for the ASC procurement. And appreciate tradeoffs between filling desires of stakeholders versus having realistic expectations...

Ken Vaughn: Now user needs as we mentioned is not in the 1202. You have to do this yourself. That's the purpose of this course. Agencies must develop them. You start with your major desired capability, your operational need that we just defined, and then you extract information from the standards to write user needs. So we have identified operational need. The standard in this case is NTCIP 1202. And now we will go about the extraction process. NEMA TS2 now there is an interrelationship between 1202 and TS2 and we're going to focus in a little bit closer on it is the main functional standard. It's based largely on NTCIP is based on the TS2 concepts but NTCIP does add a few features not defined in TS2. These are features that have been found particularly in 170 controllers for gap reduction and other things. They had gone a little bit further beyond the NEMA specification but typically most of the data is NEMA TS2, there are a few additions. Then we'll go through the extraction process. Now there is no reason to reinvent the wheel. There are some basic underlying needs that are applicable to virtually all devices and those generic needs can be used, reused within the signal controllers. You can find these say in NTCIP 1203 or 1210; 1203 is the dynamic message sign standard, 1210 is the signal system master standard. Both of these have similar sort of specifications but live data exchange, the ability to exchange data across the wire. Log data exchange, the ability to identify events locally store those events in a log so that your remote management station can query your device and pull information that's occurred in the past. That's log data exchange. And then support legacy communication networks is where the low speed communications so some bandwidth efficient options that decrease the number of bytes you are exchanging across the wire. So very high number of signal controller deployments will have very similar needs. Strongly recommend that you go look at those particular standards and copy those as much as possible for your particular deployment because they are probably extremely applicable to you. But in most cases I mean it's a signal controller, not a message sign. So most of your needs are going to be very different. So in this case you have to go to look, read the conformance groups, recognize the categories of functions involved, and infer the major desire capability that you have. And that's a process that was once again defined in A202 module, the generic level, and now we're going to apply it specifically to the NTCIP 1202 standard...

Ken Vaughn: Well when you read the conformance groups, a conformance group is a logical grouping of related objects. They help in determining required information and supportive function. An example is shown here for volume occupancy report. It's largely a listing of individual objects within that conformance group. Now sometimes you can just look at the name. You understand the object. In other cases you maybe unclear. Well the table is going to give you a reference, a clause reference and a standard and you can go look up the definition of that particular piece of data and to understand exactly what that piece of data is doing. So we will go through this. This shows you an example of the slide that the main conformance group table, listing all the different conformance groups and whether they are mandatory, optional, and the clause number they are defined in. And so you can see there are quite a few different conformance groups. NTCIP 1202 is a significant size standard. We're going to tackle a small piece of this to give you an example of how you go about this process. And we'll give you several examples. So the first example is as we start reading through from the top the phase conformance group we identify, we have phaseWalk, phaseMinimumGreen, phaseYellowChange, etcetera. Well pretty much anyone who has done much signal timing is going to fairly quickly recognize that those are your phase timing parameters. And while that directly maps to your identified operational needs of configuring an intersection and also for safety issues making sure that you don't improperly time things. So when we break these down in the infer category we can say well this is going to- there are configuration issues here and I need to be able to configure these parameters. There are potentially well I don't know if there are control issues or not. We consider monitor issues as well. We know what kind of detail. We say well yes, there are configuration issues. You may want to restrict some parameters for safety issues though. Control isn't really applicable for these items. These items are configuration items. You set them once, they manage themselves. If you want to do second-by-second signal timing from a remote location those are a different set of parameters and we'll talk about those in the future. But these you don't really control per se. You configure them once and let them be. But you may want to come back and monitor them. You may want to say what is my current setting for these configurations? And so configure and monitor are the two categories of user needs that we need to worry about here. Next you have extract capabilities from the standard. Read: phase conformance group, maxPhases has a range of 2 to 255. Well we're going to recognize that two phases isn't going to meet our needs. We have all sorts of different intersections; a lot of them are going to have left turn phases. We need more than two phases. Now exactly how many do we need? I don't know. We need to go back, talk to our stake holders, figure out what they think they need, so we can capture that in our standard, or sorry, in our specification. So that we get the product that we need for our agency...

Ken Vaughn: So phase conformance group, still in the same conformance group. We just keep reading down the different objects and we see phaseStatusGroupReds, phaseStatusGroupGreens, etcetera. Now initially we may be a little bit confused by what

these are so we can start jumping in, reading the definitions and the standard, and when you find out that this is how you monitor signal timing. PhaseStatusGroupReds is a set of eight bits in one byte. It indicates which phases are currently in the red state. So maybe phase one, three, four, five, seven, and eight are all red right now. So that allows you to monitor signal timing. That was one of our stated operational needs and we need a map to identify operational need of monitor real time signal time. And then we consider the three major categories. Do we configure these? Well no, these are simply status objects. Do we control these? No, they're only for monitoring so that's where they fit into the categorization. Next we talk about phase conformance group. We have phaseControlPhaseOmit, phaseGroupHold etcetera. Once again if you read the details on these, this is your second by second signal timing. Now that's a need that we did not identify in our operational needs. So one possibility is we don't need this. Another possibility is that we overlooked this need and we need to go back potentially and talk to our stakeholders and say "Just making sure we are not going to include this in our procurement because these are optional items, they don't have to be included and we haven't identified the need for them so why incur costs to implement them if we don't need to?"...

Ken Vaughn: We keep reading down and we jump into now the coordination conformance group and we see coordOperationalMode, coordPatternStatus. These are perhaps even more difficult to figure out what they mean, but if you read them you find out this is how you manually select a time and pattern. It's not second-by-second logic. It's more I'm going to run this pattern for now until the next few hours. So this maps to one of the defined operational needs of selecting a time and pattern and if you're going to do that, there are configurations aspects of that because I need to define what the timing pattern is. There are control aspects to it because I need to control the selection of other timing pattern. And finally there are monitoring aspects to it which is I need to figure out what was the last one commanded, what is the current configuration? There are all sorts of different monitoring capabilities that you might have there. So there are all three of those categories that will be coming into play. Finally I think this is the last one, detector conformance group. We talked about vehicleDetectorTable, maxVehicleDetectors, etcetera. This all relates to the ability to configure detectors at the intersection which maps to configuring to any intersection. And this applies to both configuring and monitoring. One last one, vehicleDetectorErraticCounts, and DetectorAlarms. This all maps to monitor signal diagnostics and out of three major categories this is once again monitoring but rather than this time just monitoring on an operational level, this is more of a diagnostic level. So with those examples we have gone through and identified all of our operational needs, at least sample requirements from the conformance groups. Now bear in mind we did not find all of our operational needs and we did not define all of our conformance groups. We did not go through all of our conformance groups. We leave that exercise to your particular project. But the intent of the slide is to show you that there

is not a one-to-one mapping. It's a complex mini-to-mini mapping that your user need may depend upon multiple conformance groups and a given conformance group may relate to multiple user needs. They are tools that kind of categorize things in the standard, but really when you look at it from a user's perspective, you're going to end up with a slightly different mapping. So that's how all that relates. We then get into how do you write the actual user need? Well we need to go back and review the rules for what a well written user need is. Once again this is from A202. First off it needs to be uniquely identifiable. Every user need needs to have its own identifier. Second, identify one major desire capability like an operational need, explains the need. It's not a "shall" type of statement. It's what do I need? Not precisely how I provide it. And that means it's solution free. And the other key aspect which is often forgotten in our user need is it needs to capture the rationale. You define your concept of operations at the very beginning of a project. It may be two, three, five years later before that project finally comes to an end and you are doing all _____, It's all too easy during that five-year period that people either forget what they wrote, can't understand or reinterpret what they meant, or you lose-- you have personnel staff turnover. All sorts of issues come up. You get to the end of the project and you say, "Why was this even a requirement? Why did we even identify this as a user need? What's the purpose of this feature?" And capturing that rationale can really go a long ways to letting people know why this need exists. So let's take an example. Configure a signal for intersection layout. Start out with a user need. You can categorize these really any way you want to. We've gone ahead and categorized them based on the configure control and monitor capabilities. So we'll start out with 2.1. We're going to configure and what we'll start out with the definition of "The agency needs a signal controller that can be safely configured to control any intersection within its jurisdiction including those with atypical layouts." That identifies your one major desired capability. So we have the unique identifier, major capability, and now you're going to make sure it's solution-free. Which after review we identify it is and now we need to make sure we have a justification. Well what is the justification? We haven't really specified that yet. So we'll go ahead and add a paragraph. "This will result in lower overall maintenance costs for the maintenance division. However the operators must still be able to control every intersection including that nasty six-legged intersection that we have over there. In addition, due to the safety critical nature of this information coupled with the static nature of information this process should require the presence of a field technician to verify any change." So the original configuration for the layout may require someone to change. Now this isn't necessarily in the standard and in fact this should be kind of a custom feature to make sure that I'm restricting some capabilities from operations. But this is an example where you may have a real-world user need that some agency personnel maybe your lawyer comes in and says, "Make sure it doesn't do this." Well, that's a real need that would need to be captured. And that's the importance of getting this justification down. You really need to identify ideally who is even specifying this requirement, this user need?

Ken Vaughn: So another example would be user need 4.1. "Agency needs a signal controller that will allow management system to monitor the status of each signal location with a one-second resolution. The agency needs a way to remotely verify the signal is operating as expected." Now one could probably debate whether that one second resolution is appropriate there. Understanding this is a user need level. That gives the user the designer's guidance about what level of accuracy they need in the system and if that's something that's important for users to specify that there's no real problem with that thing in there. But it does need to specify at minimum roughly what level of detail they need and that's a problem if you don't say one-second resolution what do you put in there? And very quick but what does very quick mean? So one second gives you very good guidance even if it is a little bit overly specific for a user need. But recognizing once again the design process, the designer may come back and say, "Well this is going to be too complex. How about a two-second need or something." In fact a one-second need is met by the standard so it's not an issue...

Ken Vaughn: That takes us to the next activity. Which of the following components is not part of a well-defined user need? A unique identifier, B major capability, C a "shall" statement, or D rationale? Once again which one of those is not part of a well-defined user needs? Unique identifier, major capability, "shall", or rationale? Go ahead and make your selection...

Ken Vaughn: And we see the correct answer is the "shall" statement. Shall is not in a user need, actually should not appear in a user need because that's really for a requirement. The other items should be a user need. Every item should have a unique identifier so it can be easily referenced and traced through the system. Every user need should have a major capability so you understand what it is. And then finally it should have a rationale so when you get to the end of the project, you understand why it's been requested...

Ken Vaughn: So that ends off learning objective number three. User needs are not defined in NTCIP 1202. Agencies must develop them themselves and they are derived from major capabilities, from stakeholder input and other sources. User need should be fully document justification, and then finally they should consider cost implications before extending the standard...

Ken Vaughn: That brings us to our fourth learning objective. The stakeholder desires versus the realistic requirements. Your stakeholders like to ask for the moon and we just have to make those requirements a little bit more realistic. NTCIP provides an excellent starting point for all standardized functionality but some features are not included. Some cases because they are just too early, adaptive control for example; lots of different

options out there for how you do adaptive control. NTCIP was not going to pick a winner or a loser. There is also less common features so a flashing yellow arrow for example or leading pedestrian intervals are not included in the NTCIP standard. Stakeholders may identify other capabilities. I mean technology is always changing so there may be other things out there you'd want that aren't included in the standard. So you have to consider for each one of these, if they are really important, and some examples, recognize if you implement something that's beyond the standard it's likely to be a custom and or a proprietary solution. And that's going to increase your cost both at well during development and specifying it and testing it and maintaining it long term. Determine if there are alternatives that stakeholders will consider, so it may not be exactly what they want but if it meets the standard rather than custom solution it will save you a lot of money. But understand the goal of this project is to make sure your needs are met. So if you talk to stakeholders and they still say, "We really need this need," then you go through the process. You undertake those costs, but you do it with your eyes wide open and there are going to be significant costs. So that's really all we have for objective four. NTCIP standardized most of the features but you may need to add additional features and non-standard features will increase life-cycle costs...

Ken Vaughn: We also have learning objective five. Explain how to evaluate conformance, to understand the minimum conformance requirements and how to appreciate backwards compatibility. NTCIP 1202 does not define SEP style requirements. They don't define functional requirements per se but it does define conformance and those conformance statements do relate to objects and those objects have definitions, and those definitions imply functionality and explicitly reference NEMA TS2 in many cases that give precise functionality. So there are requirements. They are just not the traditional functional requirements you see in most NTCIP data level standards. So the mandatory features, the mandatory performance groups must be supported. If you want to claim support for any other object or conformance group, those items must also be supported. And they also reference in many cases objects from 1201. So not only do you have 1202 objects defined by these conformance statements, but also 1201 objects. If user needs do not traced to these items it may suggest an incomplete analysis. Now recognize we are not tracing there yet but once we have to do the requirements definition phase we will provide a complete trace from user need to requirement, to objects. And if at that point you identify some of the mandatory features of mandatory conformance groups that have not traced back to any of your user needs, then you'd be pretty well convinced that you have missed something. And the way you prevent this upfront is making sure you go through look at all of those objects up front and say well is there a user need I should be considering relate to this object? And that's what you can do at this point. We said we'll also discuss backwards compatibility. We are on version two of the standard so there are some backwards compatibility issues. And also this standard references NTCIP 1201 and we're actually on the third version of that standard so there are various backwards

compatibility issues. When dealing with this, you don't specify the need for backwards compatibility just because. The intent is if you have version one deployments, if you have equipment out there that already is NTCIP conformant, then your new equipment needs to interoperate with that old equipment. So your central system for example may need support both versions. On the other hand, if you have your old signal system and you are just expanding your system, you may want your new signal controller to support both versions as well so it will work with today's signal system master or central system. And it will also work five years from now when I replace that system. So this backwards compatibility may exist on either the central system side, signal system master side, or in the local side. If any differences exist in the specifications or design, the new equipment may have to be designed to work with both the older and newer versions and from the user need perspective, this just need to be properly documented...

Ken Vaughn: So there are four major backwards compatibility issues related to signal controllers. The first is data based transaction feature. Now this particular feature defines how you manage these large uploads and downloads of data. And the version one and version two of the Global Objects standard 1201 made a change in this. They identified some ambiguities in the first version. They corrected those in the second version. So any new deployment should definitely support the unambiguous definition which is version two, but if you already have existing deployments, then you may want to make sure you are backwards compatible with them. Same thing with Daylight Savings Time. This was actually a more recent update, version two to version three. This is actually a more critical update as well because in this case version two defined Daylight Savings based on U.S. law. And unfortunately Congress, decided in the mid-2000s decided that we're going to change our Daylight Savings algorithm and we're going to go to Daylight Savings earlier and leave it later. Well that broke all existing deployments at NTCIP and that was really our fault. We designed NTCIP to be inflexible and it broke on us. So we ended up having to update NTCIP 1201. Now that Daylight Savings logic is fully configurable. So if Congress decides to change the standard again, all you have to do is go in and reconfigure your devices rather than doing a software upgrade. Right now you are either forced to do a software upgrade or your management system has to manually go out and change the time in your controller every time Daylight Savings occurs. But at least there's a work-around there. But any new controllers it should certainly support the version three. There is probably very little reason to do backwards compatibility since you probably will never need to do the version two U.S. Daylight Savings Time because that's no longer applicable law. The third item for backwards compatibility is the logic in setting local time and this was an ambiguity that we found early on in the deployments. We fixed it from version one to version two. Once again in the Global Logic standard and that has been fixed since I think 1999 roughly timeframe. To clarify what happens, if you set local time during a Daylight Savings Time transition period and there are some ambiguities in there that we corrected within version two. Once again, select version two. You only need to

worry about version one the old way the ambiguous way if you have that older equipment. Finally there is one item; we talked about Annex D of NTCIP 1202. It does contain one object and a special function output state. That was deprecated. It means it's no longer used. It's the old way of doing things because we found out there were some problems with it specifically it's a poor way of doing things. The output state is whether or not this is on or off. Well we had originally designed that to be both the control feature and the status feature and in version two we replaced that with two distinct objects. So now I can say I am going to turn the feature on through one object and then I can monitor its status by a second object that will tell me whether it's on or off because potentially I could turn it on and there is some logical problem in the controller where it's not actually turning on. That way I can verify that in fact it did receive the command to turn it on but for some reason it's not on. So slight different change there from version one to version two. But for the number of objects contained in the standard it actually shows you how solid this standard is. Very, very few changes have been made. It is a fairly stable standard, even version two. So each change resulted from identifying an ambiguity or a problem in the initial version of the standard you really only need to support the older way of doing it if you have that backwards compatibility. If you have existing equipment you need to deal with...

Ken Vaughn: That brings us to another activity. And this question is which is the best way to complete this user need? User need 5.1 operators need the central system to A work with manufacturer X model one controllers that are currently deployed as these are too costly to update; B support dbErrorTYpe, dbErrorID, dbTransactionID, and dbMakeID; C, support backwards compatibility for older controllers; or D, support NTCIP version one and NTCIP version two so that it can work with older controllers that are too costly to update? So which one of those is the best way to complete the user need? And I'll give you a second to complete those. Go ahead and make your selection now...

Ken Vaughn: The correct answer is work with manufacturer X model one controllers that are currently deployed, as these are too costly to update. This is the correct answer because the statement identifies the desired capability sufficiently to allow the designer create a solution without additional information that is also solution free and indicates rationale. So for example with this statement I know where to go. I can go to the manufacturer X, model 1, controller specification and that should in theory at least tell me exactly how that controller operates. What objects it supports, the ranges it supports and everything. If I don't have the full definition, then I have to go back to a person and say, "What specific controller do I need to interoperate with?" Now answer B was incorrect because user needs should not find a specific solution. This is getting to individual detailed items that you are specifying out of the standard. That's not the intent of user needs should be a much higher level than that, just the capability. The other problem with this is it doesn't provide any sort of justification. It just said support these objects. The third answer is incorrect because the statement does not well once again does not

provide justification. It also does not provide a type of backwards compatibility so it just says support backwards compatibility. Well what type of backwards compatibility? There were four particular types of backwards compatibility to worry about. Which in particular of those four do I need to worry about? And then finally answer D was incorrect because well version one and version two of what? Is it version one and version two of 1201? In which case I don't need to worry about Daylight Savings Time issue? So this statement is also incorrect. It doesn't specify the precise standard. It does however give you the reasoning, justification for it, but it did not really adequately specify what's being resolved so those are different issues. So a summary of learning objective number five. NTCIP 1202 defines a minimum set of items that must be supported to claim conformance. It defines the features within NTCIP 1202 depend on 1201 and there have been multiple versions of these standards and the user should consider whether this will be an issue within their particular system depending on whether or not they have older controllers that they have to deal with...

Ken Vaughn: That brings us to our final learning objective. A relationship of needs to requirements. We'll talk about the introduction of the traceability table, then also talk about where the next module is going to follow on from this one. User needs should be traced to the requirements. This is a very important concept so that you understand why each and every requirement is there; so that when you get to the project you get to the testing and something fails you can trace back how is this going to impact operationally my system? And if you do this through a proper table structure you can very quickly identify how is this going to impact my system? This can be done through a simple protocol requirements list very similar to what are used in the other NTCIP-based standards. The difference is if you're familiar with those tables, we don't need a conformance column. Those tables include a conformance column to say this feature is mandatory or required, and or whether it's an optional for the standard. Well, we are all going to be listing up items that are relevant to our particular project so of course everything is going to be required. We don't need a conformance column and a selection column, but the same basic table structure is very important because it provides your project a checklist. At the end of the project you go through and check off each feature, did they meet this? Did they fill these requirements? And it helps you producing your test procedures as well. So what does this table look like? Well there are four basic columns to it. The left two columns identify your user need and give you the title of that user need and give you the title of that user need. The right two columns the same thing for your requirement; it gives you the cause ID of your requirement, and then the requirement title. That way you can get a basic idea of what the requirement user need is from that one column and if you need more details you can look up the reference column and go to that precise clause in your specification and get those details...

Ken Vaughn: So based on all the examples we went through in this training course we filled out a table like this. Right now all we have are user needs. Those are all kind of header rows shown in gray. We have one example of a white row which will be in the future used to show our requirements and that will show you that you just put the requirements right underneath the relative user needs and we'll talk about in the next module there will be mini-mini mapping. Your same requirement can appear multiple times under different user needs and all of that. But the next module we'll talk about some sample requirements specifically related to the control selection of time and pattern and we'll leave that to next time. User needs describe what features the device needs to support and why. The functional requirements support and satisfy these needs and to detailed measurable specifications. So whereas we talked about before the user needs to find your justification and what it is, your functional requirement becomes more the "shall" statement and then from the "shall" statement it's very measurable so that your test procedures can be written to those measured requirements. The traceability table visually maps relationships between the user needs and the requirements and allows very quick referencing so you can quickly identify what those items are...

Ken Vaughn: As we mentioned the functional requirements when we consider the next module really. It's a two-part module: Specifying Requirements for Actuated Traffic Signal Controllers Based on the NTCIP 1202 Standards. Some of the controllers are quite complex. There are a lot of issues to consider that's why it's a two part module...

Ken Vaughn: That brings us to our last activity. The benefit of the needs to requirements table is that it A maps needs to requirements, B provides a high-level summary of the features, C provides a convenient checklist during deployment, or D all of the above? So go ahead and make your selection now...

Ken Vaughn: So the correct answer in fact is D all of the above. And we'll go through each one of these. Each one of these are of course incorrect simply because they all apply. Maps needs to requirements. That's actually the main purpose is to show that mapping. Provides high-level summary of features, well it does. It provides a quick reference guide to each feature we look up reference. And finally provides a convenient checklist during deployment. That becomes very valuable during the testing phase so that once again you can trace all the way back to user need and identify what features the operators stakeholders demanded can't be fulfilled because of this one requirement not being fulfilled...

Ken Vaughn: So in summary learning objective number six, user needs should be summarized in a table. That table will be used to map user needs requirements in the

next phase in our next module. And the requirements will provide measurable statements for subsequent testing...

Ken Vaughn: What have we learned today? NTCIP 1202 does not include SEP content. Number two, a description of ASC functionality and a practical baseline conformance statement for NTCIP 1202 can be found in NEMA TS2. Number three; measured capabilities can be extracted from NTCIP 1202. Number four, requiring non-standard features will increase life cycle costs. Number five, conformance to the standard only requires minimal support of the mandatory objects contained in the mandatory conformance groups. In other words what we mean by "minimal support" is you don't necessarily have to support the full object range. You only have to support the minimal requirement. And then finally six, user needs should be mapped to requirements in a table. So those were the different things we learned today in our six different learning objects. The resources today we have NTCIP 1201, that's where all of the Global Object Definitions are defined. NTCIP 1202 is the Object Definitions for ASC. NEMA TS2 Traffic Controller Assemblies with NTCIP Requirements. That defines once again the functionality of signal controller. NTCIP 9001 we didn't discuss specifically but that provides a guide to all of NTCIP operations, everything from very high level concepts all the way down to the byte level. And then IEEE 1362 is the IEEE standard that defines how you produce a Concept of Operations document and obviously if you're going to be producing your own, that's a document you should be aware of. Well with that, any questions and one question that we've encountered before is how the ATC standards play into this? And the ATC standards are new. There is adaptive logic that is coming along. There are some specifications for adaptive **MIBS**. So something manufacturers who have produced adaptive control for ATC controllers have produced their own MIBS similar to NTCIP 1202 but designed for their particular adaptive logic. And you can use those but they are not standardized yet. But they are you contact the manufacturers or the developers of the adaptive logic and they will likely work with you to help you specify what you need there and that's been the main question. The basic traditional actuated signal controller logic still can work on ATCs and those ATCs can deploy the standard without a problem but if you want to use adaptive control then you probably need to work with some of those other groups to use their efforts.

Ken Vaughn: And that brings us to the final slide of our program which is the next module. It's actually as we mentioned a two-part module because there is a lot of information but it will explain how to write ASC requirements, derive sample requirements to satisfy needs, and explain how to show the relationship between requirements and the design from the standard. So with that we conclude our presentation and I pass it back to you, Nicola.

End of A315a_Edited.wmv